

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 3**



“Modelación y construcción de los componentes
Gestor de actividades y Calendario para el
Subsistema Planificación por Objetivos del Sistema Integral de Gestión
de Entidades Cedrux.”

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

Autores:

Ariadna Rendón Artola
Manuel Alejandro Castellanos Pérez

Tutor:

Ing. Yuliet Galán Ramírez

Co-Tutor:

Ing. Nestor Bernal Vidal

Ciudad de la Habana, Junio de 2011.
“Año 53 de la Revolución”

El Estado organiza, dirige y controla la actividad económica nacional conforme a un plan que garantice el desarrollo programado del país, a fin de fortalecer el sistema socialista, satisfacer cada vez mejor las necesidades materiales y culturales de la sociedad y los ciudadanos, promover el desenvolvimiento de la persona humana y de su dignidad, el avance y la seguridad del país.

Artículo 16. Constitución de la República de Cuba



DECLARACIÓN DE AUTORÍA:

Declaramos que somos los únicos autores del trabajo “Modelación y construcción de los componentes Gestor de actividades y Calendario para el Subsistema Planificación por Objetivos del Sistema Integral de Gestión de Entidades CedruX” y autorizamos a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Ariadna Rendón Artola
Autor

Manuel Alejandro Castellanos Pérez
Autor

Ing. Yuliet Galán Ramírez
Tutor

Ing. Néstor Bernal Vidal
Tutor

AGRADECIMIENTOS

A la Revolución Cubana por brindarnos la oportunidad de graduarnos como Ingenieros en Ciencias Informáticas en una universidad de excelencia.

A nuestros padres, familiares y amigos.

A nuestros tutores Nestor y Yuliet por su apoyo incondicional en la elaboración de este trabajo.

A todas las personas que de una forma u otra nos han apoyado a lo largo de la carrera.

DEDICATORIA

A mi Madre que con su esfuerzo y dedicación ha sabido encaminarme
por el sendero de la humildad, tomando como
premisas la buena voluntad y la gratitud.
A ella que con un cariño desinteresado
ha apoyado cada etapa de mi vida.
A mi Padre que me ha enseñado
el verdadero significado del trabajo.
A mi familia, por su incondicional apoyo
en los instantes crudos y felices de mi vida.
A Eddy por ser mi látigo, mi seductor y maestro.
A todos mis amigos, que con sus ir y venir han engrandecido mi alma.

.....*Ariadna Rendón Artola*.....

Dedico esta tesis ante todo a mis padres, quienes me han enseñado a encarar las adversidades sin perder nunca la dignidad ni desfallecer en el intento. Me han dado todo lo que soy como persona, mis valores, mis principios, mi perseverancia y mi empeño, y todo ello con una gran dosis de amor y sin pedir nunca nada a cambio.

A mi mamá por su dedicación. A mis hermanos y mis abuelas Nena y Nana por su gran corazón.

A todos mis familiares, con quienes he compartido momentos inolvidables y a mi nene
quien me dio los mejores momentos de mi vida en la universidad
y me enseñó lo que es amar.

.....*Manuel Alejandro Castellanos*.....

RESUMEN

El Estado cubano organiza, dirige y controla la actividad económica nacional conforme a un plan que garantice el desarrollo programado del país. Dicho plan se elabora bajo los conceptos del modelo de Planificación por Objetivos establecido nacionalmente cuyos cimientos son los procedimientos: Dirección por Objetivos y la Planeación Estratégica. Uno de los subprocesos que encierra este modelo para la consecución de los resultados según los objetivos trazados en el plan, es el correspondiente al Registro, Control y Seguimiento a las actividades. Este posibilita un ordenamiento de las responsabilidades a cumplir en una determinada entidad por cada uno de sus integrantes. Sin embargo, en las entidades nacionales se gestiona a partir de herramientas informáticas que no posibilitan desarrollarlo en correspondencia con el presente curso de las funciones en una entidad. Con tal motivo se propone una solución que comprende la Modelación y Construcción de los componentes Gestor de actividades y Calendario para el Subsistema Planificación por Objetivos del Sistema Integral de Gestión de Entidades CedruX; garantizando de este forma que se registre, controle y se facilite el seguimiento a las actividades en el proceso de Planificación por Objetivos vigente en las entidades nacionales.

Palabras clave:

Planificación, Planificación por Objetivos, Registro, Control, Seguimiento, Elementos de planificación, Plan, Objetivos, Actividades.

ÍNDICE

Introducción.....	1
Capítulo I: Fundamentación Teórica	5
1.1 Introducción.....	5
1.2 Marco conceptual.....	5
1.3 Estado del arte del proceso: Planificación por Objetivos.....	7
1.4 Breve valoración del estado del arte	10
1.5 Lenguajes de Modelado y desarrollo.....	10
1.5.1 Lenguajes de modelado.....	11
1.5.2. Lenguajes del lado del cliente.....	11
1.5.3. Lenguaje del lado del Servidor	12
1.6 Marcos de trabajo (Frameworks)	12
1.6.1 Sauxe	13
1.6.2 ExtJS 2.2.....	13
1.6.3 Zend Framework	13
1.6.4 Doctrine	14
1.7 Tecnologías y Herramientas de Desarrollo	14
1.7.1 AJAX	14
1.7.2 Herramienta CASE: Visual Paradigm.....	14
1.7.3 Sistema de Control de Versiones: Subversion.....	15
1.7.4 Entorno Integrado de Desarrollo (IDE): NetBeans	15
1.7.5 Servidor Web: Apache	15
1.7.6 Sistema Gestor de Bases de Datos: PostgreSQL.....	15
1.8.7 Navegador web y complementos	16
1.8 Modelo de Desarrollo	16
1.9 Conclusiones del capítulo:	17
Capítulo II: Características del Sistema	18
2.1. Introducción.....	18
2.2. Objeto de estudio	18
2.2.1. Descripción de los procesos del negocio.....	18
2.2.2. Objeto de automatización.....	19

2.2.3.	Propuesta de solución.....	21
2.3.	Modelación del negocio.	22
2.3.1.	Diagramas de proceso de negocio (BPMN).	22
2.3.2.	Especificación de procesos de negocio.	25
2.3.3.	Validación del proceso de negocio.....	28
2.3.4.	Modelo Conceptual.	29
2.4.	Requerimientos.	29
2.4.1.	Técnica de Captura de Requisitos.	29
2.4.2.	Listado de Requisitos Funcionales Identificados.	30
2.4.3.	Especificación de Requisitos Funcionales.	31
2.4.4.	Validación de los requisitos funcionales.	34
2.4.5.	Requerimientos no funcionales.	34
2.5.	Conclusiones del capítulo.	36
Capítulo III: Modelación del Sistema		37
3.1	Introducción.....	37
3.2	Arquitectura de la solución.	37
3.2.1	Estilos arquitectónicos aplicados	38
3.2.2	Estructura de empaquetamiento	39
3.3	Patrones de diseño empleados en la solución propuesta.....	40
3.3.1	Patrones GRASP	40
3.3.2	Patrones GoF	42
3.4	Diseño de la solución.	42
3.4.1	Diseño de la solución en términos de componentes.	43
3.4.2	Diseño de clases.	44
3.4.2.1	Diagramas de clases del diseño	44
3.4.2.2	Modelos de datos	46
3.4.3	Estrategia de integración.	48
3.4.4	Prototipos de interfaz de usuario.	51
3.5	Conclusiones del capítulo.	52
Capítulo IV: Implementación y validación de la solución propuesta		53
4.1	Introducción.....	53
4.2	Implementación.	53
4.2.1	Estándares de código.....	53

4.2.2	Diagrama de despliegue.	54
4.3	Validación de la solución propuesta.....	55
4.3.1	Validación del diseño propuesto.	55
4.3.2	Pruebas de aplicación.	59
4.4	Conclusiones del capítulo.	64
	Conclusiones generales	65
	Bibliografía Referenciada	67
	Bibliografía Consultada	69

ÍNDICE DE FIGURAS

Figura 1: Relación de los conceptos asociados al dominio.....	7
Figura 2: Procesos del modelo de desarrollo.	17
Figura 3: Diagrama del proceso de negocio Registro de actividades.	23
Figura 4: Diagrama del proceso de negocio Control de actividades.	24
Figura 5: Diagrama del proceso de negocio Seguimiento a las actividades.	24
Figura 6: Modelo conceptual.	29
Figura 7: Organización de las vistas de la arquitectura del sistema CedruX.	38
Figura 8: Diagrama de componentes del Subsistema Planificación por Objetivos.	43
Figura 9: Diagrama de clases del componente Gestor de Actividades.	44
Figura 10: Modelo de datos del Subsistema Planificación por Objetivos.	47
Figura 11: Estrategia de integración.	48
Figura 12: Prototipo de interfaz de usuario del requisito Adicionar Actividad.	51
Figura 13: Prototipo de interfaz de usuario del requisito Buscar Actividad.	52
Figura 14: Prototipo de interfaz de usuario del requisito Establecer Relaciones.	52
Figura 15: Diagrama de despliegue de escenario para PC cliente con disco.	55
Figura 16: Diagrama de despliegue de escenario para PC cliente sin disco.	55
Figura 17: Representación de la cantidad de clase y el número procedimientos que contienen.	56
Figura 18: Representación en % de la cantidad de clase y el número procedimientos que contienen.	57
Figura 19: Representación del valor en % del atributo Responsabilidad.	57
Figura 20: Representación del valor en % del atributo Complejidad de Implementación.	57
Figura 21: Representación del valor en % del atributo Reutilización.	57
Figura 22: Representación de las asociaciones de uso por cantidad a de clases.	58
Figura 23: Representación en % de las asociaciones de uso por cantidad a de clases.	58
Figura 24: Representación del valor en % del atributo Acoplamiento.	58
Figura 25: Representación del valor en % del atributo Complejidad de Mantenimiento.	59
Figura 26: Representación del valor en % del atributo Cantidad de Pruebas.	59
Figura 27: Representación del valor en % del atributo Reutilización.	59
Figura 28: No conformidades por iteración detectadas durante las pruebas funcionales.	64

ÍNDICE DE TABLAS

Tabla 1: Proceso Registro de actividades.	25
Tabla 2: Especificación del requisito Adicionar Actividad.	31
Tabla 3: Descripción del diseño de clases asociadas al componente Gestor de Actividades.	45
Tabla 4: Servicios internos del Subsistema PDO gestión de las actividades y el calendario	49
Tabla 5: Caso de prueba “Adicionar actividad”.	62
Tabla 6: Descripción de las variables.	63

INTRODUCCIÓN

La experiencia de la aplicación del perfeccionamiento empresarial en las Fuerzas Armadas Revolucionarias y su extensión a la economía civil permitió a la dirección del país, a raíz del V Congreso del Partido Comunista de Cuba, trazar una estrategia económica con el fin garantizar el mejoramiento y adecuación de las bases teóricas, metodológicas y organizativas de la planificación a las realidades del momento. De esta forma se repasaba el papel primario de la planificación, en un momento complejo del desarrollo del país, como consecuencia de la incorporación, a principios del año 1996, de procedimientos novedosos como los relacionados con la Planeación Estratégica y la Dirección por Objetivos.

Devenido de estas concepciones de planificación, en el país se confeccionó un modelo adaptado a las condiciones nacionales; el cual se definió por los especialistas como “Planificación por Objetivos”. Este se encarga, a través de fórmulas más autónomas y ajustadas a la realidad cubana, de establecer resultados finales hacia los cuales se dirigen las actividades organizacionales e individuales, en aras de cumplir los objetivos económicos y sociales que demanda el desarrollo integral de la sociedad. La puesta en práctica de este medio de administración posibilita avistar una transparencia en la gestión profesional, una planificación infalible, así como la justificación irrefutable de petición de recursos; posibilitando priorizar, clarificar y unificar los propósitos en las entidades. (2) Uno de los subprocesos que encierra la Planificación por Objetivos, para la consecución de los resultados, es el correspondiente al Registro, Control y Seguimiento a las actividades. Este subproceso posibilita el ordenamiento de las responsabilidades a cumplir en una determinada entidad por cada uno de sus integrantes, garantizando que todo el personal se encuentre identificado con las tareas a desempeñar, las metas a alcanzar y las prioridades establecidas. Sin embargo, el referido proceso en algunas entidades se gestiona a partir del sistema P-TRAB¹. Debido a su arquitectura (desarrollado sobre MSDOS²) y a la concepción del modelo de Planificación por Objetivos mediante el que fue concebido, no posibilita desarrollar el proceso de Registro, Control y Seguimiento de las actividades, en correspondencia con el presente curso de las funciones en una entidad.

La solución que implementa el P-TRAB descuida en la ejecución actual del proceso de Registro, Control y Seguimiento de las actividades los siguientes requerimientos primarios:

- ✓ Asociación dinámica de las actividades con los planes establecidos por la entidad, los factores que influyen en el mismo y los objetivos trazados.

¹ **P-TRAB**: sistema informático cubano que automatiza desde finales de la década del 90 el proceso de Planificación por Objetivos.

² **S.O. MSDOS**: Microsoft Disk Operating System, sistema operativo de disco de Microsoft.

- ✓ Registro, control y seguimiento de las actividades dependiendo de una jerarquía de roles.
- ✓ Reporte de las actividades atendiendo a criterios determinados.
- ✓ Escalabilidad del sistema.
- ✓ Acceso inmediato y desde cualquier lugar.

Diversas entidades encuentran auxilio en herramientas como Outlook, Project y Excel del paquete de programas Microsoft Office. Estas alternativas que no están destinadas a gestionar una planificación que tenga en cuenta el manejo de Planes, Objetivos, Factores que influyen el plan y Actividades de manera secuencial, centralizada y sistemática como define el modelo cubano de planificación. Por lo tanto, tampoco satisfacen la ejecución del proceso con los requerimientos estimados.

Actualmente carece el país de un sistema informático de gestión que contenga la totalidad de los requerimientos de funcionalidad, interoperabilidad y seguridad, de manera que pueda ser utilizado como herramienta para potenciar el cumplimiento de las funciones en las entidades a todos los niveles.⁽¹⁸⁾ Con tal propósito el Ministerio de Finanzas y Precios en conjunto con la Universidad de las Ciencias Informáticas (UCI), las Fuerzas Armadas Revolucionarias (en lo adelante FAR) y la colaboración de especialistas de diversas entidades desarrolladoras de software³, se encuentran absortos en la producción de un Sistema Integral de Gestión de Entidades denominado CedruX. Con este sistema que se pretende proporcionar a las entidades nacionales una herramienta que les permita evaluar, implementar y administrar con mayor facilidad su negocio.

Uno de los subsistemas que componen el producto CedruX es el de Planificación por Objetivos (en lo adelante se referenciará por las siglas PDO). Este debe ser capaz de interrelacionar objetivos de trabajo, actividades y recursos en tiempo real; garantizando el seguimiento del desarrollo y cumplimiento de los objetivos y tareas principales en las entidades, de una manera segura y factible. Sin embargo dicho subsistema carece de una solución que en contribución al logro de una funcionalidad íntegra, permita informatizar el proceso de Registro, Control y Seguimiento a las Actividades en las entidades nacionales.

Derivado de la situación problemática expuesta, se deduce que el **problema a resolver** radica en: ¿Cómo registrar, controlar y facilitar el seguimiento a las actividades en el proceso de Planificación por Objetivos vigente en las entidades nacionales, dentro del Subsistema PDO del Sistema Integral de Gestión de Entidades CedruX?

Con vista a la solución del problema se define como **objeto de estudio**: el Proceso de Planificación por Objetivos, trazándose como **objetivo general**: Desarrollar los componentes Gestor de actividades y Calendario para el Subsistema PDO del Sistema Integral de Gestión de Entidades CedruX, de modo que se registre, controle y se facilite el seguimiento a las actividades

⁴ **Software**: se refiere al equipamiento lógico o soporte lógico de una computadora digital.

en el proceso de Planificación por Objetivos vigente en las entidades nacionales. Quedando este desglosado en los siguientes **objetivos específicos**:

1. Fundamentar la investigación del proceso de Registro, Control y Seguimiento a las actividades como subproceso de la Planificación por Objetivos, mediante la elaboración del Marco Teórico.
2. Modelar los componentes Gestor de actividades y Calendario.
3. Implementar los componentes Gestor de actividades y Calendario.
4. Efectuar pruebas de validación a los componentes implementados.

Por consiguiente la presente investigación se encuentra enmarcada en el **campo de estudio**: Proceso de Registro, Control y Seguimiento de las actividades en las entidades nacionales.

Teniendo en cuenta los elementos expuestos anteriormente, se define como **idea a defender**: El desarrollo de una solución que automatice el proceso de Registro, Control y Seguimiento a las actividades en el Subsistema PDO del Sistema Integral de Gestión de Entidades CedruX, constituirá una mejora en la realización del proceso de Planificación por Objetivos en las entidades del país.

El desarrollo de los objetivos específicos se encuentra encauzado por la planificación de las siguientes **tareas de la investigación**:

1. Recopilación de la bibliografía necesaria para comprender los principales conceptos relacionados con el proceso de Registro, Control y Seguimiento a las actividades dentro de la Planificación por Objetivos.
2. Caracterización de determinados sistemas, tanto nacional como internacionalmente, que implementan alguna variante de la Planificación por Objetivos.
3. Descripción del Modelo de desarrollo propuesto y de la Arquitectura Base definida para el Sistema.
4. Justificación de las herramientas que se utilizarán en la implementación de solución que se presenta.
5. Modelación del proceso de negocio Registro, Control y Seguimiento a las actividades en la Planificación por Objetivos.
6. Identificación y especificación de los requisitos del proceso de negocio Registro, Control y Seguimiento a las actividades.
7. Diseño de los componentes Gestor de actividades y Calendario para el proceso Registro, Control y Seguimiento a las actividades.
8. Implementación de los componentes Gestor de actividades y Calendario para el proceso Registro, Control y Seguimiento a las actividades.
9. Realización de pruebas de validación y calidad final de los componentes desarrollados.
10. Documentación integral del trabajo realizado.

Lo que permitirá como **posible resultado** de la presente investigación la modelación y construcción de los componentes Gestor de actividades y Calendario para el Subsistema PDO del Sistema Integral de Gestión de Entidades CedruX.

La **estrategia de investigación** empleada es la referida como Experimental o Explicativa, pues permite establecer los vínculos causales, leyes y mecanismos internos del objeto de investigación, partiendo de conocimientos suficientes a cerca del problema que posibilitan plantear una hipótesis (en este caso se traduce a idea a defender) a nivel explicativo. (1)

Los **métodos de investigación científica** en los que se encuentra cimentada la investigación son: dentro del conjunto de Métodos Teóricos, el de Modelación, donde se lleva a cabo una modelación del proceso de Registro, Control y Seguimiento a las actividades, permitiendo predecir la respuesta de dicho proceso a variaciones de algunos de sus parámetros, sin tener que ejecutar el proceso en la realidad; el Analítico – sintético para el estudio y análisis de la bibliografía, la formulación del problema y los objetivos y el Histórico- Lógico para el manejo de los datos relacionados con el establecimiento del proceso de Planificación por Objetivo en el país.

Para la obtención de los datos se empleó el Método Empírico de Entrevista, con el objetivo de establecer una concepción inicial de la problemática. Estas entrevistas fueron realizadas a profesionales de la Factoría de Planificación por Objetivo del proyecto ERP-Cuba.

Estructura del documento:

Capítulo 1: En este capítulo se expone el estado del arte referente al proceso de Planificación por Objetivos y específicamente del subproceso de Registro, Control y Seguimiento a las actividades. Se realiza la fundamentación teórica del tema. Se describen y valoran algunas de las soluciones existentes. Se justifican las herramientas y tecnologías a utilizar para la modelación e implementación de la solución que se propone y por último se señalan las características principales del Modelo de desarrollo adoptado.

Capítulo 2: Características del sistema:

Se describen las características esenciales que debe tener la solución propuesta, sustentadas por la modelación de los procesos negocio y la definición de los requisitos. Se figuran, además, algunos de los artefactos propuestos en la fase de Modelación definida en la metodología de software aplicada.

Capítulo 3: Modelación del sistema:

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

Capítulo 4: Implementación y Validación de la solución propuesta:

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 I: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción.

En el presente capítulo se ofrece información acerca del proceso de Planificación por Objetivos y específicamente del subproceso de Registro, Control y Seguimiento a las actividades en las entidades nacionales. Además se ofrece una breve reseña de las concepciones: Dirección por Objetivo y a la Planeación Estratégica como cimientos del modelo de planificación adoptado en Cuba. Se exponen también, los principales conceptos afines al problema en cuestión. Se hace alusión a las principales características de algunas soluciones tanto nacionales como internacionales, que implementan alguna variante de la Planificación por Objetivos. Por último se ofrece una visión de la metodología, tecnologías y las herramientas que se utilizarán durante el desarrollo de la investigación.

1.2 Marco conceptual.

En la actualidad predominan muchos enfoques acerca del concepto de planificación. Sin embargo, de acuerdo con la definición del Máster en Administración y Dirección de Empresas Andrés M. Núñez Parada, se plantea que la planificación consiste en determinar las metas u objetivos a cumplir, en seleccionar las misiones, los objetivos y las acciones para alcanzarlos; requiere seleccionar entre diversos cursos de acción futuros.(46) En concordancia con dicha opinión, los autores consideran que la Planificación como una de las funciones básica del Proceso de Gestión, permite establecer un determinado nivel de estabilidad en las entidades, teniendo en cuenta condiciones de adaptabilidad que avizoran los acontecimientos que pudieran suceder.

Un rasgo esencial en el proceso de planificación lo constituye el método Dirección por Objetivos. Este posibilita la determinación y consecución de las metas en una organización. Peter Drucker, considerado por muchos especialistas como padre de dicha concepción, en su obra clásica “The Practice of Management” publicada en 1954, la formuló como “proceso por el cual los miembros de una organización identifican objetivos comunes, definen las áreas de responsabilidad de cada uno en términos de resultados medibles esperados (objetivos), y usan esas medidas como vías para la operación de las acciones”. La Dirección por Objetivos se conoce también como la Administración por Objetivos, de igual forma se utiliza el concepto de Dirección o Administración por Resultados. La presente investigación acoge la primera nomenclatura.

La Dirección por Objetivos es combinada en muchos los países en vía de desarrollo e incluso en algunos países desarrollados, con elementos de la Planeación Estratégica. Esta última surge formalmente a mediados de los años sesenta auspiciada por el libro de Ansoff (1965), Estrategias Corporativas. Según el investigador mexicano Alfredo Acle Tomasini, la Planeación Estratégica “es

un conjunto de acciones que deben ser desarrolladas para lograr los objetivos estratégicos (...)" Sin embargo Henry Mintzberg y James Waters en el artículo "De las estrategias deliberadas y emergentes", del Diario de Gestión Estratégica, ofrecen una definición más acertada en la que señalan que la Planeación Estratégica es un proceso formulado de planeación a largo plazo que se utiliza para definir y alcanzar metas organizacionales.

De la combinación de las concepciones Dirección por Objetivos y Planeación Estratégica, se conformó un modelo cubano, denominado por los especialistas "Planificación por Objetivos". Este modelo a partir de fórmulas más autónomas, establece resultados finales hacia los cuales se dirigen las actividades organizacionales e individuales. La concepción: Planificación por Objetivos, se refiere a una forma particular de funcionamiento en torno a las tareas directivas. Es un sistema de planificación 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. (2) Por consiguiente, en el presente trabajo se entiende por Planificación por Objetivos, los fines hacia los cuales se orientan los planes, actividades, recursos y tareas de cualquier unidad organizativa para cumplir con el propósito de su creación.

El proceso de Registro, Control y Seguimiento a las actividades como subproceso de la Planificación por Objetivos constituye el eslabón primario en la materialización de las acciones encaminadas al cumplimiento de las metas trazadas en una entidad. Los objetivos definidos centralmente serán cumplidos, en la medida que los miembros de la organización tengan una contribución consiente, imperante y activa, en la concreción de las acciones emprendidas para alcanzarlos. Es el proceso que garantiza la puntualización de las actividades a efectuar por cada individuo, teniendo en cuenta elementos que direccionan el proceder del personal, en consecuencia con los objetivos previamente establecidos. Además, precisa el control del estado de cumplimiento de las responsabilidades asignadas, a partir de parámetros que permiten establecer trazas del desarrollo de una determinada actividad.

Este proceso, como materialización de la planificación a corto plazo (denominada por los especialistas como Planificación Operativa), posibilita una mayor coincidencia entre lo que aspira la dirección y lo que debe proponerse cada miembro de la organización. Fija en mayor medida la responsabilidad de los directivos y trabajadores al generar y comprometerse directamente todos con las acciones a desarrollar. Se potencializa la responsabilidad y objetividad de la evaluación de los directivos, pues se fundamenta en lo que cada individuo debe hacer para lograr el desarrollo con resultados concretos de la actividad que dirige, a partir de sus propios compromisos. (3)

La siguiente figura se resalta la relación entre los procesos asociados al dominio del problema:

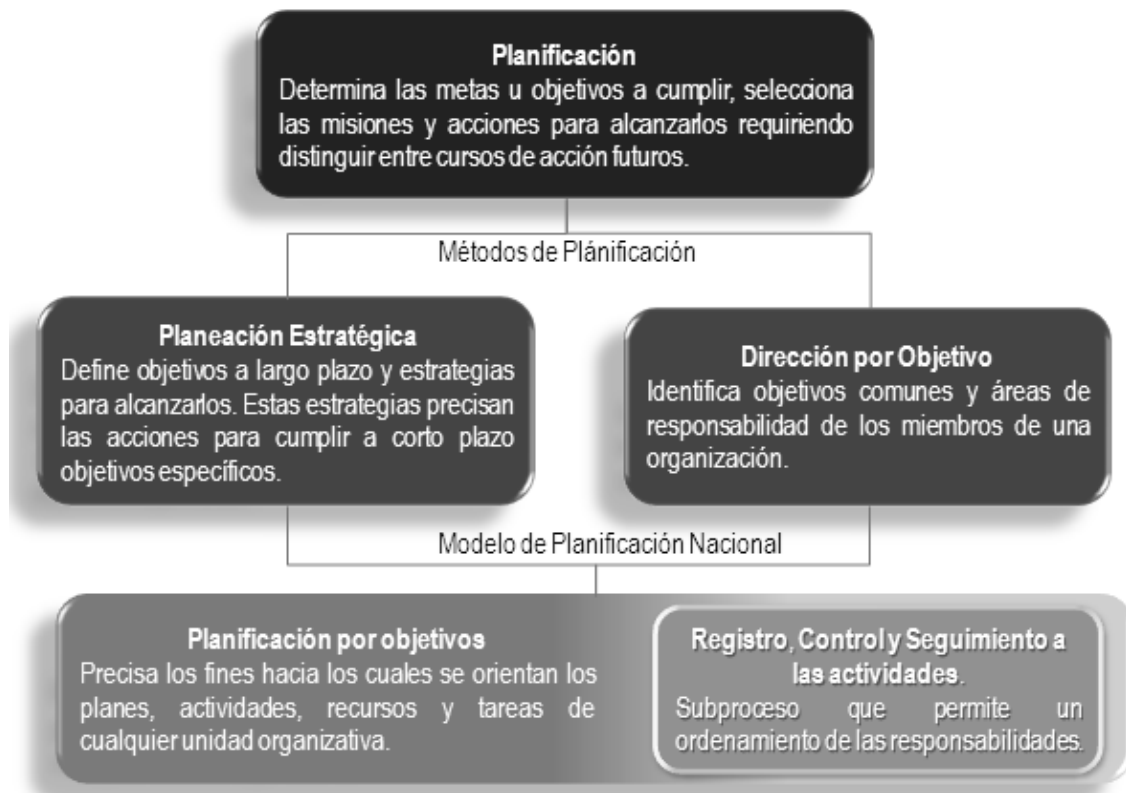


Figura 1: Relación de los conceptos asociados al dominio.

1.3 Estado del arte del proceso: Planificación por Objetivos.

En Cuba el sistema que se utiliza para gestionar el proceso de Planificación por Objetivos en las entidades es el P-TRAB. Este sistema está diseñado para trabajar solamente sobre el sistema operativo⁴ 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.

Entidades vinculadas al Ministerio de Educación, de las FAR y al Consejo de Estado, encuentran auxilio en herramientas como el Outlook, el Project y el Excel del paquete de programas Microsoft Office. Estas aplicaciones, presentan ínfimas funcionalidades para desarrollar una planificación definida por objetivos, pues no fueron elaboradas con ese propósito. A nivel internacional no se conoce ningún sistema que implemente a cabalidad el proceso de Dirección por Objetivo (dígase nacionalmente “Planificación por Objetivos”). Muchas compañías como la Thinking People⁵, adecúan aplicaciones informáticas que solo trasladan la estrategia de la empresa a objetivos concretos tratando de linear, medir y conocer el rendimiento de sus trabajadores. Sin embargo no profundizan en la vasta gestión que implica desarrollar una planificación regida por objetivos.

⁴ **Sistema operativo:** es un programa informático que actúa de interfaz entre los dispositivos de hardware y el usuario.

⁵ **Thinking People:** es una consultoría de Zaragoza formada por Licenciados en Psicología, Empresariales, Derecho, Ingeniería e Informática, especializados en Recursos Humanos.

A continuación se expondrán algunas características de sistemas de gestión empresarial utilizados tanto en el ámbito nacional como internacional, con el objetivo de establecer un análisis de las funcionalidades que estos poseen respecto al proceso de planificación, y dentro de este, requerimientos referentes al ordenamiento de las responsabilidades a cumplir en una organización. Los sistemas internacionales han sido seleccionados atendiendo a que son valorados como los principales sistemas de gestión a nivel mundial, tanto los de carácter propietario como los de libre distribución. (47) Se ha escogido para analizar el sistema nacional Versat Sarasola debido a que este es considerado el más completo de los existentes en Cuba.(48)

OpenBravo ERP

Se caracteriza por ser una aplicación completamente web. Una de sus principales ventajas es que sigue un licenciamiento de software libre. En este sistema de gestión económica, las funcionalidades referentes al proceso de planificación se encuentran distribuidas por algunos de sus módulos. Por ejemplo:

- ✓ **Gestión de aprovisionamientos:** Se planifican las necesidades de aprovisionamiento, por explosión de las necesidades de producción, teniendo en cuenta acciones mínimas, plazos de entrega y pedidos en curso.
- ✓ **Gestión proyectos y servicios:** Su funcionamiento se basa en el modelado de la estructura productiva de cada organización, así como de los datos relevantes para la producción: planes de producción y productos involucrados en las mismas, entre otras funcionalidades.
- ✓ **Gestión Financiera y Contabilidad:** Su actividad se resume entre otras acciones en el establecimiento de los planes por defecto, la definición de los planes y ejercicios contables, así como de la gestión interanual y los presupuestos. (4)

Observación: OpenBravo ERP constituye un potente sistema integral de gestión de pequeñas y medianas empresas; cuya solución comercial en software libre tiene la ventaja de integrarse eficientemente con una herramienta de Inteligencia de Negocios. Sin embargo, describe la ejecución de un proceso de planificación que no se ajusta a la gestión de un plan central desde el nivel de Estado como se establece en el modelo cubano de planificación. Además es concebido para un ambiente netamente empresarial, es decir no tiene en cuenta la gestión en entidades no productoras (presupuestadas según modelo cubano).

OpenERP

OpenERP es un completo sistema de gestión de empresas de licencia libre (AGPL⁶), que cubre las necesidades de las áreas principales de una organización. Incorpora funcionalidades de gestión de documentos y conectores con otras aplicaciones. Permite trabajar remotamente mediante una

⁶ **AGPL** (Afferro General Public License) es una licencia de copia permitida derivada de la Licencia Pública General de GNU diseñada para asegurar la cooperación con la comunidad en el caso de aplicaciones que corran en servidores de red.

interfaz web o aplicación de escritorio multiplataforma (Windows, Linux y Mac) e incluye un entorno modular de programación rápida de aplicaciones. Presenta entre sus módulos el correspondiente a la Gestión y Planificación de proyectos. Este permite entre otras funcionalidades: definir tareas y requerimientos de estas, asignar con eficiencia los recursos a los requerimientos, planificar proyectos, controlar tareas y establecer calendarios de tareas. (5)

Observación: OpenERP implementa, a partir de las funcionalidades que presenta el módulo Gestión y Planificación de proyectos, un ordenamiento de las responsabilidades a cumplir por los integrantes de una organización. No obstante, estas no desprenden de metas y prioridades orientadas centralmente como se establece en el modelo de Planificación por Objetivos. Además al igual que OpenBravo es concebido para un ambiente netamente empresarial. Dadas las condiciones de adaptabilidad que presenta este sistema puede ser posible incorporarle funcionalidades que permitan ajustar el módulo Gestión y Planificación de proyectos a las necesidades particulares del modelo cubano de planificación. Sin embargo la primicia de construir un ERP que netamente responda al modelo de gestión empresarial de las entidades cubanas exige una implementación autóctona del mismo.

SAP ERP

Se caracteriza por ser un sistema totalmente preparado para trabajar mediante la web, debido al empleo de la plataforma tecnológica denominada SAP NetWeaver⁷. SAP ERP no cuenta con un módulo centralizado destinado a la planificación de todas las áreas de las empresas, en su lugar incluye este proceso en los módulos o áreas funcionales que lleven a cabo una planificación de determinado recurso, variando en algunas soluciones en dependencia de las particularidades de la cada solución ofrecida.

SAP presenta los siguientes módulos que manejan el proceso de planificación de las empresas:

- ✓ **Planificación de la Producción:** Engloba las distintas tareas y metodologías utilizadas en el proceso de producción.
- ✓ **Gestión de materiales:** Tiene como función principal brindar soporte a todas las fases de gestión de materiales: planificación de necesidades y control, compras, entrada de mercaderías, gestión de acciones y verificación de facturas.
- ✓ **Control:** Este módulo además de documentar sucesos reales, permite determinar las desviaciones comparando los datos reales con los datos planeados. (6)

Observación: SAP ERP es un sistema completo de gestión empresarial que proporciona a sus usuarios la tecnología necesaria para contar con una amplia visión de todas las actividades de su negocio. Sin embargo, presenta como principal inconveniente que es privativo. De hecho, Microsoft

⁷ **SAP NetWeaver** es una plataforma conformada por varias herramientas enfocadas a optimizar y sincronizar los recursos informáticos con los requisitos particulares a las estrategias y aplicaciones.

Windows es la única plataforma que soporta todas las soluciones SAP. Además utiliza como gestor de base de datos Oracle.

Versat Sarasola

Se caracteriza por ser una aplicación de escritorio. Concebido sobre una plataforma de trabajo Cliente-Servidor lo que permite además su instalación en red. Se encuentra implementado en el lenguaje de programación Delphi y trabaja sobre el sistema operativo Windows. Además presenta soporte para bases de datos SQL Server 2000. Entre los 10 módulos con que cuenta este Sistema cubano de Gestión Contable-Financiero se encuentra el de Planificación Económico-Productiva. Mediante este subsistema se logra una planificación adaptable a cualquier actividad económica, bajo los principios de que todos los resultados se logran por los niveles más importantes de la entidad: Centros de Costo, Áreas de Responsabilidad, Productos, Unidades Contables y Consolidados Empresariales. (7)

Observación: Este sistema presenta una gran significación para Cuba por ser el primer sistema de contabilidad cubano certificado, según las nuevas normativas establecidas por los Ministerios de Finanzas y Precios y de la Informática y las Comunicaciones, para este tipo de programa. (7) Sin embargo, solo se sustenta sobre mecanismos o principios del Presupuesto Maestro, descartando el proceso de Planificación por Objetivos. Además presenta el inconveniente de estar desarrollado sobre tecnologías privativas.

1.4 Breve valoración del estado del arte

Existe un solo producto informático como solución a la automatización del proceso de Planificación por Objetivos en Cuba, que no cumple además con los requerimientos estimados para desarrollarlo (P-TRAB). Por tal motivo algunas entidades se auxilian en aplicaciones del paquete Microsoft Office que no están diseñadas para estos propósitos. En el contexto internacional, se percibe una carencia de aplicaciones que automaticen cabalmente la gestión dicho proceso según las condiciones nacionales. Estos elementos evidencian la imperiosa necesidad de elaborar una solución que responda acertadamente a la ejecución actual del proceso de Planificación por Objetivos en las entidades cubanas. Solo de esta manera es posible pautar las directrices de sus operaciones encauzadas al logro del progreso organizacional y económico.

1.5 Lenguajes de Modelado y desarrollo.

Los lenguajes de modelado y desarrollo que se emplean para el desarrollo de la solución que se propone, han sido establecidos por la dirección del proyecto ERP-Cuba, en consecuencia con las tecnologías, herramientas y marco de trabajo definidos para la confección del producto CedruX.

1.5.1 Lenguajes de modelado

Se denomina lenguaje de modelado de objetos al conjunto estandarizado de símbolos y de modos de disponerlos para modelar un diseño de software. Para la modelación de la solución que se propone se emplearan como lenguaje de modelado los siguientes:

➤ **UML**

El Lenguaje de Modelado Unificado (Unified Modeling Language) como notación orientada a objetos, se empleará con el fin de especificar y documentar un sistema de software, de un modo estándar incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema. UML implementa un lenguaje de modelado común para todos los desarrollos por lo que se crea una documentación también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo. (12)

➤ **BPMN**

BPMN (Business Process Modeling Notation) es empleado en el desarrollo de la solución que se propone para presentar gráficamente las diferentes etapas del proceso. Este estándar de modelado de procesos de negocio ha sido diseñado específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes. BPMN a diferencia del lenguaje de modelado unificado (UML) toma un perfil orientado a procesos en el modelado de sistemas, por lo tanto la combinación ambas notaciones, al ser compatibles entre sí, ayudan a modelar con mayor precisión la situación actual y deseada en los procesos de negocio del cliente. (13)

1.5.2. Lenguajes del lado del cliente

➤ **HTML**

Es el lenguaje que se utilizará para definir páginas clientes de la aplicación. Se compone por un conjunto de etiquetas utilizadas para definir y ubicar los distintos elementos que componen una página web. Como un lenguaje de marcación de elementos para la creación de documentos hipertexto, HTML puede describir hasta un cierto punto la apariencia de un documento. Puede incluir uno o varios scripts, como por ejemplo JavaScript o PHP, los cuales pueden afectar el comportamiento del HTML. Su principal desventaja es que todos los navegadores no interpretan el código HTML de la misma manera. (15)

➤ **JavaScript**

Se recurrirá al lenguaje JavaScript para acceder a los objetos de la capa de presentación de la solución que se propone con la presente investigación. Este lenguaje es basado en objetos y principalmente se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. Entre las acciones típicas que se pueden realizar en JavaScript existen dos vertientes. Por un lado los efectos especiales sobre páginas web para crear

elementos dinámicos. Por el otro, permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se puede crear páginas interactivas⁸. Sin embargo en caso de ser un script muy cargado de datos, tomara mucho más tiempo visualizarse en el navegador.

➤ **Hojas de Estilo en Cascada(CSS)**

Las hojas de estilo en cascada serán utilizadas para representar todo lo referente a los estilos (dígase tamaños, colores, iconos, imágenes, tipografías, espacios y bordes). Constituyen el estándar para la inserción de estilos a documentos estructurados, como por ejemplo, páginas HTML o XML. El objetivo de la definición de este estándar del W3C es permitir la separación entre las normas de presentación y el propio contenido a mostrar. (16)

➤ **XML (Extensive Markup Language)**

XML como un metalenguaje de definición de documentos estructurados mediante marcas o etiquetas se usará en la creación de las reglas básicas que permiten el intercambio de información estructurada entre aplicaciones; se emplea también para tareas de validación y configuraciones en el sistema. Se trata de un estándar del W3C⁹ que posibilita compartir la información de una manera segura, fiable y fácil. Además permite compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. Es también un lenguaje del lado del servidor. (14)

1.5.3. Lenguaje del lado del Servidor

➤ **PHP v5.2**

PHP es el lenguaje que se empleará para programar del lado del servidor. Es un lenguaje de programación interpretado, completamente orientado al desarrollo de aplicaciones web dinámicas. Es gratuito, fácil de usar y aprender, portable, de código abierto y multiplataforma. Presenta interfaces para una gran cantidad de sistemas de base de datos, así como bibliotecas incorporadas para muchas tareas web habituales. Sin embargo este lenguaje debido a su flexibilidad presenta como principal inconveniente, que mal utilizado, puede convertir al sitio en un punto de fácil de acceso a piratas informáticos. Por otra parte con respecto a otros lenguajes de programación como Java o C++ es mucho menos robusto y la programación orientada a objetos es aún muy deficiente para aplicaciones grandes. (17)

1.6 Marcos de trabajo (Frameworks)

Un marco de trabajo, en el desarrollo de sistemas computarizados, es una estructura de soporte definida mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros

⁸ **Páginas interactivas:** Páginas web, que permiten mediante sus características comunicarse con el usuario.

¹² **W3C:** El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web.

software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Muchos los definen como 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. (18)

1.6.1 Sauxe

Sauxe es el marco de trabajo que se empleará para el desarrollo de la solución que se propone. El cual fue desarrollado en la UCI como fruto del paradigma de independencia tecnológica por el que aboga el país. Este marco de trabajo, fusionado bajo tecnología totalmente libre (entre ellas PHP, Postgresql, Apache) posee el desarrollo de tecnologías propias basadas en otros marcos de trabajo como ZendFramework para el manejo de la lógica de negocio, Doctrine para el acceso a datos y ExtJS para la capa de presentación. Cuenta con una arquitectura en capas que a su vez presenta en su capa superior un MVC¹⁰. Contiene un conjunto de componentes reutilizables que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo. (19)

1.6.2 ExtJS 2.2

ExtJS es el marco de trabajo que se empleará para el desarrollo de la capa de presentación. Está basado completamente a la programación orientada a objeto. Cada objeto contiene lo típico: propiedades, métodos y eventos. Basa toda su funcionalidad en JavaScript a través de librerías. Así, en tiempo de ejecución carga y crea todos los objetos HTML a través del uso intenso de DOM¹¹. Los datos son obtenidos con AJAX¹² a través de XML. Una de las grandes ventajas de utilizar ExtJS es que permite crear aplicaciones complejas utilizando componentes predefinidos. La carga de procesamiento se distribuye, permitiendo que el servidor al tener menor carga, pueda manejar más clientes al mismo tiempo. (20)

1.6.3 Zend Framework

Se utilizará el marco de trabajo Zend Framework para el manejo de la lógica de negocio. Este marco de trabajo de código abierto brinda facilidades de uso y poderosas funcionalidades. Está diseñado para la versión 5 de PHP y posee buenas capacidades de ampliación. Proporciona un sistema de caché de forma que se puedan almacenar diferentes datos, así como los componentes que forman la infraestructura del patrón Modelo-Vista-Controlador. Consta de mecanismos de filtrado y validación

¹¹ **(MVC).Modelo Vista Controlador** es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

¹² **DOM**: El DOM o Document Object Model (Modelo de Objetos de Documento) es una forma de representar documentos estructurados (tales como una página web HTML o un documento XML) que es independiente de cualquier lenguaje orientado a objetos.

¹³ **AJAX**: Acrónimo de Asynchronous JavaScript And XML (JavaScript y XML asíncronos), es una técnica de desarrollo web para crear aplicaciones interactivas.

de entradas de datos. Permite convertir estructuras de datos PHP a JSON¹³ y viceversa, para su utilización en aplicaciones AJAX y provee capacidades de búsqueda sobre documentos y contenidos. (21)

1.6.4 Doctrine

Para la capa de acceso a datos se empleará Doctrine. Este es un potente y completo sistema ORM¹⁴ (en inglés Object Relational Mapper) para PHP 5.2 o superior que incorpora una DBL (capa de abstracción a base de datos). Uno de sus rasgos importantes es la habilidad de escribir opcionalmente las preguntas de la base de datos orientada a objeto. Esto les proporciona una alternativa poderosa a diseñadores de SQL¹⁵, manteniendo un máximo de flexibilidad sin requerir la duplicación del código innecesario. Además, exporta una base de datos existente a sus clases correspondientes y convierte clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos. (22)

1.7 Tecnologías y Herramientas de Desarrollo

1.7.1 AJAX

Ajax por sus siglas en inglés Asynchronous JavaScript And XML (JavaScript asíncrono y XML) es la técnica de desarrollo web que se usará para poder hacer consultas asíncronas al servidor sin necesidad de recargar la página. Esta surge de la combinación de tres tecnologías ya existentes: HTML (o XHTML¹⁶) y Hojas de Estilo en Cascada (CSS) para presentar la información, Document Object Model (DOM) y JavaScript, para interactuar dinámicamente con los datos, además de XML y XSLT¹⁷, para intercambiar y manipular datos de manera desincronizada con un servidor web. (23)

1.7.2 Herramienta CASE: Visual Paradigm

Se empleará Visual Paradigm 3.4 como herramienta CASE¹⁸. Utiliza UML 6.0 como lenguaje de modelado, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad¹⁹ con otras aplicaciones. 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

¹⁵ **JSON**: Es un formato ligero para el intercambio de datos. (32)

¹⁶ **ORM**: Componente de software que permite trabajar con los datos persistidos como si fueran parte de una base de datos orientada a objetos. (33)

¹⁷ **SQL**: Lenguaje de consulta estructurado utilizado para acceder a bases de datos relacionales que permite realizar varias operaciones sobre estas.

¹⁶ **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.

¹⁷ **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)

¹⁸ **CASE**: es un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un sistema informático. (24)

²¹ **Interoperabilidad**: Es la condición mediante la cual sistemas heterogéneos pueden intercambiar procesos o datos.

proyectos UML. Ayudando a construir aplicaciones de calidad más rápido, mejor y en abajo costo. (25)

1.7.3 Sistema de Control de Versiones: Subversion

La versión 1.6.5 de Subversion (SVN) es la herramienta de entorno colaborativo que se utilizará para el control de versiones. Este se encuentra preparado para funcionar en red y se distribuye bajo licencia libre. Mantiene versiones no sólo de archivos, sino también de directorios y versiones de los metadatos asociados a esos directorios. Además de los cambios en el contenido de los documentos, se mantiene la historia de todas las operaciones de cada elemento, incluyendo la copia, cambio de directorio o de nombre. Ofrece mejor uso del ancho de banda, ya que en las transacciones se transmiten sólo las diferencias y no los archivos completos. (26)

1.7.4 Entorno Integrado de Desarrollo (IDE): NetBeans

La presente solución se desarrollará sobre el IDE de programación multiplataforma NetBeans 6.8. El cual tiene soporte para la versión 5 de PHP, ExtJS, el diseño de Hojas de Estilo (CSS) y HTML. Se integra con varias herramientas como el Subversion y servidores web, presenta una gran estabilidad. Es un producto de código abierto, con todos los beneficios del programa disponible en forma gratuita. Hace uso de plugins²⁰ para ampliar sus funcionalidades, lo que le da una gran facilidad de uso.

1.7.5 Servidor Web: Apache

Se utilizará como servidor web Apache 2.0 pues es una tecnología gratuita de código abierto compatible con muchos Sistemas Operativos. Tiene todo el soporte que se necesita para tener páginas dinámicas. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Tiene una alta configurabilidad en la creación y gestión de registros de actividad. Apache permite la creación de ficheros de registro a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor. (27)

1.7.6 Sistema Gestor de Bases de Datos: PostgreSQL

Como Sistema Gestor de Base de Datos (SGBD) se empleará la versión 8.3.3 de PostgreSQL. Este es un sistema de gestión de bases de datos relacional ²¹orientada a objetos. Es una herramienta de código abierto, de bajo coste y multiplataforma. Se destaca en ejecutar consultas complejas, subconsultas y uniones de gran tamaño. Permite la definición de tipos de datos personalizados e incluye un modelo de seguridad completo. Soporta transacciones, claves ajenas con

²² **Plugins:** es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

²³ **Base de datos relacional:** Modelo utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente basados en el empleo de relaciones entre las tablas.(29)

comprobaciones de integridad referencial y almacenamiento de objetos de gran tamaño. Cuenta con varias herramientas gráficas de diseño y administración de bases de datos como el pgAdmin. (28)

1.8.7 Navegador web y complementos

➤ Mozilla Firefox

Se utilizará como navegador web Mozilla Firefox 3.6.10. Este navegador multiplataforma es libre y es compatible con los estándares web (señálese HTML, XML, CSS y JavaScript) que se emplearán para el desarrollo a la solución que se propone. Incluye entre sus funcionalidades un mecanismo para añadir funcionalidades mediante extensiones. Precisamente Firebug es el complemento que se integrará a Mozilla Firefox para ayudar a desarrollar, evaluar y depurar la aplicación, controlando el CSS y HTML en tiempo real, midiendo el tiempo de carga para optimizar la página o corrigiendo los posibles inconvenientes con JavaScript.

1.8 Modelo de Desarrollo

La línea de producción Planificación del proyecto ERP-Cuba, de la cual los autores son miembros, es la encargada de la construcción del Subsistema de Planificación por Objetivos. La dirección de la misma decidió acoger el modelo de desarrollo de aplicaciones informáticas que utiliza la Unidad de Compatibilización, Integración y Desarrollo De Software para la Defensa (en lo adelante UCID) radicada en la UCI. Esto se debe a los resultados satisfactorios alcanzados con la aplicación de dicho modelo, durante ocho años. Por consiguiente se empleará el referido modelo para el desarrollo de la solución que se propone con el presente trabajo de diploma.

La descripción de dicho modelo se encuentra plasmada en el documento Proceso de Desarrollo y Gestión de Proyectos de Software (1ra Versión). En dicho documento describe la secuencia de actividades de alto nivel para la construcción y desarrollo de soluciones.. Este modelo de desarrollo se logra con la combinación entre los modelos basado en Componentes, el Iterativo y el Incremental. El enfoque iterativo e incremental garantiza que luego de cada iteración se obtenga como resultado una parte del sistema parcialmente completo, probado, integrado y estable, es decir un incremento. Mientras que el modelo basado en componentes posibilita alcanzar un mayor nivel de reutilización de soluciones. Proporciona una independencia en la ejecución de las pruebas a cada uno de los componentes minimizando las afectaciones a otras partes del sistema. Posibilita el mejoramiento de la calidad de una aplicación pues un componente puede ser construido y perfeccionado continuamente. (34)

El modelo de desarrollo se encuentra definido por los siguientes procesos:



Figura 2: Procesos del modelo de desarrollo.

El ciclo de vida de un proyecto que utiliza este modelo se encuentra estructurado por las fases secuenciales: Inicio, Modelación, Construcción, Explotación Experimental y Despliegue. Estas definen qué trabajo técnico se efectuará en cada una, cuando se generarán los entregables, como se revisarán, verificarán y validarán en estas, quién está involucrado en cada fase y cómo controlar y aprobar cada una. (34)

1.9 Conclusiones del capítulo:

La Planificación por Objetivos y como subproceso de este, el de Registro, Control y Seguimiento de las actividades, constituyen los eslabones primarios en la proyección de las acciones encaminadas al cumplimiento de las metas trazadas en las entidades cubanas. Este capítulo fue esencial para conceptualizar los mismos y valorar el estado del arte a partir del análisis de sistemas tanto nacionales como internacionales que implementan alguna variante de estos procesos; evidenciándose la carencia de un sistema informático capaz de ejecutar según el curso actual de la planificación nacional, tales procesos. Se realizó una presentación de las tecnologías y herramientas propuestas por la dirección del proyecto para el desarrollo de la solución, así como un resumen del modelo de desarrollo adoptado.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción

En el presente capítulo se describen las características esenciales que debe tener la solución que se propone, las cuales se encuentran sustentadas por la modelación de los procesos negocio. Se figuran, además, algunos de los artefactos propuestos en la fase de Modelación definida en la metodología de desarrollo aplicada.

2.2. Objeto de estudio

2.2.1. Descripción de los procesos del negocio.

Las entidades nacionales, dada las singulares condiciones económicas actuales están convocadas a la realización de una planificación con mayor rigurosidad. Específicamente, el registro, control y seguimiento de las actividades dentro de la Planificación por Objetivos, constituye un proceso crucial para distribuir la fuerza de trabajo, direccionando la organización hacia el cumplimiento de fines precisos. A pesar de la significación de este proceso, la mayoría de las entidades lo efectúan manualmente o sobre tecnologías que no garantizan la totalidad de los requerimientos que la misma demanda. Entre las deficiencias reconocidas en el desarrollo del proceso de registro, control y seguimiento a actividades dentro de la Planificación por Objetivos en las entidades nacionales, se detectaron las siguientes:

- ✓ Utilización de la herramienta P-TRAB, que implementa una versión desactualizada de modelo de Planificación por Objetivo, por lo tanto, no permite que se desarrolle el proceso de Registro, Control y Seguimiento a las actividades según las exigencias actuales de las entidades nacionales, además de no cumplir con requisitos de escalabilidad del sistema, compatibilidad con varias plataformas y de acceso inmediato y desde cualquier lugar.
- ✓ Auxilio en herramientas privativas como Outlook, Project y Excel del paquete de programas Microsoft Office, que no están destinadas a gestionar una planificación definida por objetivos.
- ✓ Las alternativas de solución adoptadas hasta el momento no abarcan la gestión dinámica de las actividades ni las relaciones múltiples que pueden tener estas con los planes definidos por la entidad, los objetivos, los factores que influyen el plan y las áreas de resultados clave.

- ✓ Implementación de un mecanismo insuficiente para el control del cumplimiento de las actividades en el tiempo estimado. Las aplicaciones utilizadas no permiten establecer un control estricto en la consecución de las actividades.
- ✓ Ineficacia del sistema de asignación de las actividades a los responsables y a los participantes de las mismas, pues no se tiene en cuenta la jerarquía de roles, ni un soporte eficiente para informar a todo el personal simultáneamente de la asignación de las mismas.
- ✓ Registro inestable de la documentación referente al proceso de Registro, Control y Seguimiento a las Actividades y su vinculación con planes establecidos por la entidad, los factores que influyen en el mismo y los objetivos trazados. Esto ocurre debido a que los sistemas informáticos utilizados conciben la información de manera aislada.
- ✓ Por consiguiente lo antes expuesto genera dificultades en la elaboración de los reportes del estado de avance de la entidad en dependencia de las actividades realizadas.

Esta situación es estimulada por la inexistencia de un mecanismo informático que eficientemente estreche la vinculación de las actividades a desarrollar por el personal, con los planes establecidos por la entidad, los factores que influyen en el mismo y los objetivos trazados. La presencia de un sistema informático (P-TRAB) desarrollado sobre plataforma envejecida, con un grado nulo de escalabilidad y de soporte a mecanismos de integración con otras aplicaciones, tampoco garantiza la efectividad del proceso; a pesar de constituir esta herramienta, la única solución implementada en Cuba, para la ejecución del proceso de Planificación por Objetivos.

2.2.2. Objeto de automatización

El Registro, Control y Seguimiento a las actividades es el subproceso de la Planificación por Objetivos que será objeto de automatización. Este se encuentra estrechamente relacionado con los subprocesos de Concepción y Ejecución del plan. Precisamente, contribuye a la concreción de las acciones que se plasman en el plan de la entidad, una vez que se haya emitido la documentación donde se orienta la planificación por el nivel superior. Dicha documentación está compuesta por el Documento Rector²², los Factores que Influyen el Plan y la Planificación del Control. Durante el proceso de Concepción del plan se prevén con anticipación los objetivos y actividades que se realizarán para alcanzarlos. Mientras que en el proceso de Ejecución del plan se actualizan y ejecutan las actividades en dependencia de los objetivos a los que tributan.

En la fase inicial de confección de un plan se proponen y concilian las actividades que tributan a los objetivos previamente definidos en la entidad. En caso de recibir del nivel superior una emisión de un

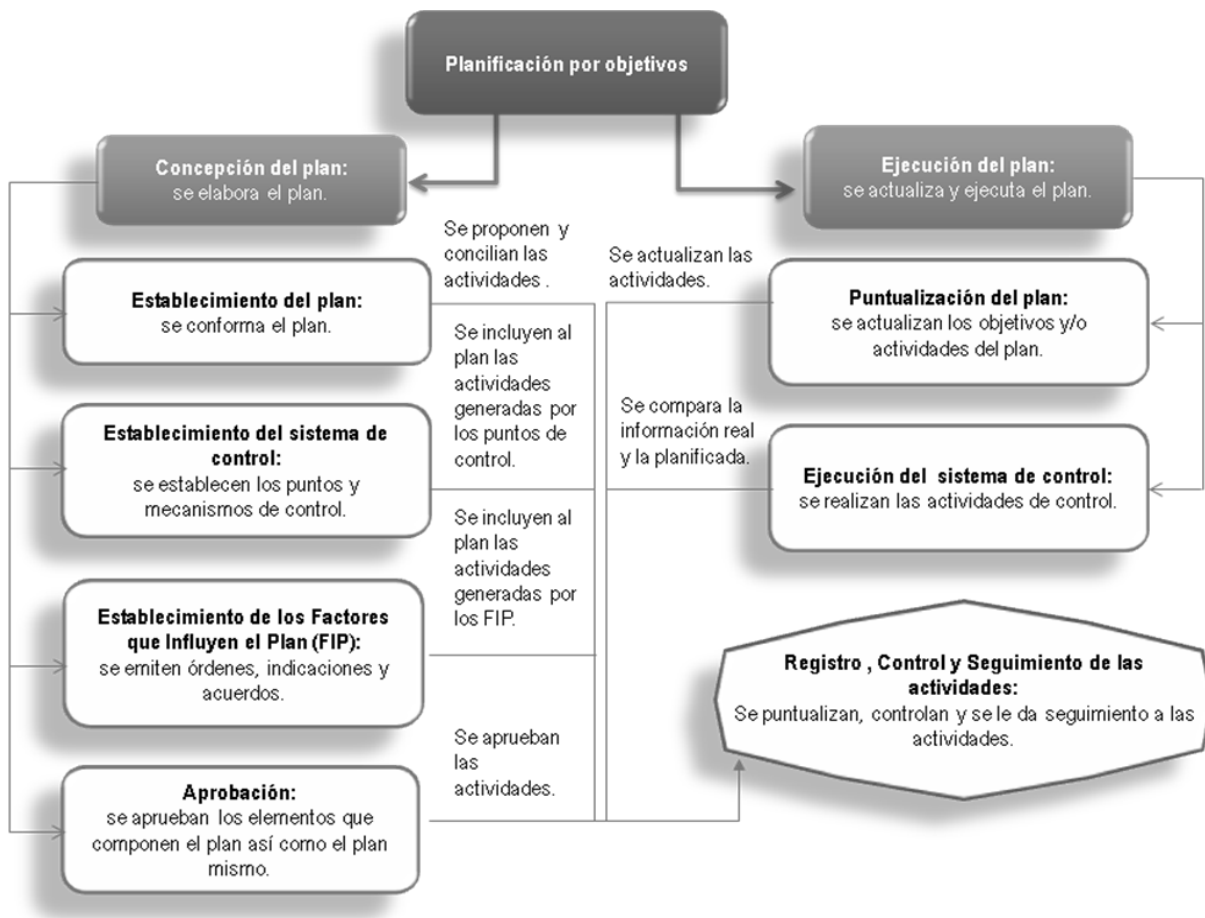
²²**Documento rector:** Disposición de rango superior que contiene las medidas organizativas, áreas de resultados claves, objetivos, actividades principales y otros, que sirven de base para que los niveles inferiores elaboren el plan.

factor que influye en el plan, se formulan para darle cumplimiento nuevas actividades que deben incluirse en el plan. Luego de conformado el documento final que contiene el plan de la entidad con sus respectivas actividades y objetivos, es sometido a aprobación por funcionarios del órgano de planificación. Estos apoyados en el Documento Rector, emiten el dictamen de validación. El plan elaborado puede estar sujeto a modificaciones, esto implicaría además realizar actualizaciones en sus objetivos y actividades.

Las actividades reflejadas en el plan son el núcleo del sistema de control que certifica la correcta ejecución del mismo a partir del cumplimiento de los objetivos trazados según la planificación. Una vez puntualizados y establecidos los parámetros de control, los funcionarios del órgano de planificación proponen los momentos en que se realizará el control al plan. La definición de los puntos de control genera nuevas actividades que se deben incluir en el plan. Como parte de la ejecución del sistema de control se revisa la información real y se comprueba su coincidencia con la planificada. En consecuencia, se decide mantener el curso de acción o emprender acciones correctivas para el caso de que existan desviaciones críticas²³.

El siguiente recuadro resalta la intrínseca relación del proceso de Registro, Control y Seguimiento a las actividades con los procesos del Concepción y Ejecución del plan como subprocesos a su vez de la Planificación por Objetivos.

²³**Desviaciones críticas:** se refiere a los incumplimientos que se generan de la comparación entre lo que se planifica y la información real.



Los elementos antes expuestos constatan la proporción del proceso de Registro, Control y Seguimiento a las actividades, el cual se encuentra compuesto por los tres subprocesos que conforman su denominación. El subproceso Registro de las actividades permite definir y aprobar las actividades que tributan al plan que define la entidad, una vez que se haya orientado la planificación por el nivel superior. Durante el Seguimiento a las actividades se confirma el progreso en la ejecución de las mismas y se actualiza el estado de cumplimiento. Mientras que en el subproceso Control se realiza un examen sobre la ejecución de las actividades del plan para emprender acciones correctivas si existen desviaciones críticas entre la información real que se revisa y la planificada.

2.2.3. Propuesta de solución.

Subsistema PDO del Sistema Integral de Gestión de Entidades CedruX constituye una solución para la automatización del proceso de Planificación por Objetivos vigente en las entidades nacionales. Sin embargo no presenta aún una funcionalidad que permita el manejo de las actividades como parte del proceso de Registro, Control y Seguimiento a las mismas. Por tal motivo con el presente trabajo de diploma se pretende incorporar una herramienta compuesta por un Gestor de actividades y un Calendario de modo se que viabilice el registro, control y seguimiento a las actividades, cumpliendo con los principios de integridad, confidencialidad y disponibilidad de la información.

La solución que se propone garantizará los requerimientos relacionados con la escalabilidad del sistema, compatibilidad con varias plataformas, confiabilidad, usabilidad, funcionalidad, seguridad, así como el acceso inmediato y desde cualquier lugar. Al mismo tiempo permitirá establecer un sistema de asignación (registro) de las actividades dependiendo de la jerarquía de roles, así como un control del estado de cumplimiento de las responsabilidades asignadas, a partir de parámetros que permiten establecer trazas al desarrollo de una determinada actividad. También posibilitará, asociar dinámicamente las actividades con los planes establecidos por la entidad, los factores que influyen en el mismo y los objetivos trazados.

2.3. Modelación del negocio.

2.3.1. Diagramas de proceso de negocio (BPMN).

A continuación se modela el proceso de negocio correspondiente al Registro, Control y Seguimiento a las actividades empleando la notación para la representación de procesos de negocio (BPMN). Esta notación posibilita una representación visual del proceso, especificando la relación entre los roles, artefactos y eventos que intervienen en su desarrollo.

Figura 3: Diagrama del proceso de negocio Registro de actividades.

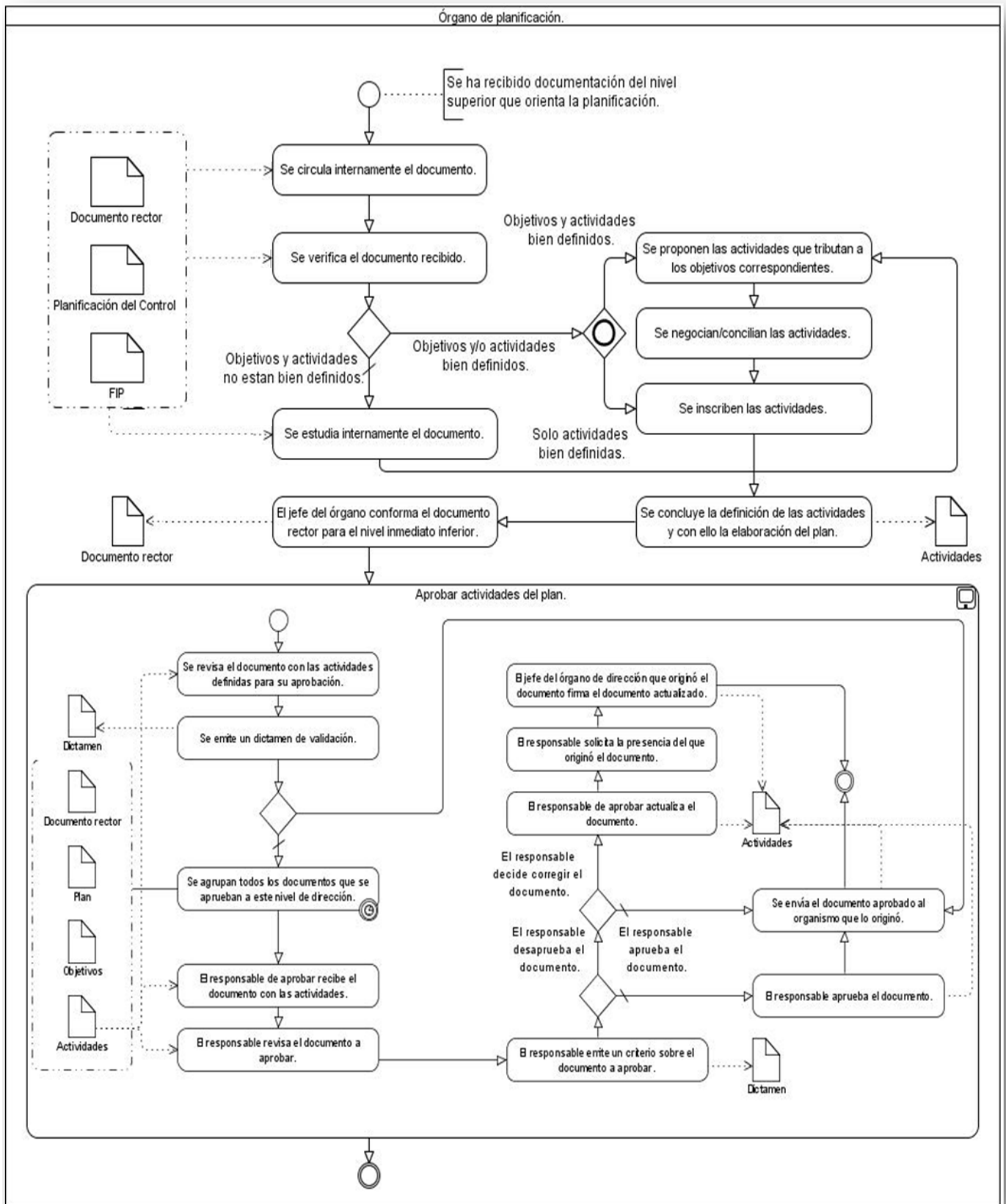


Figura 4: Diagrama del proceso de negocio Control de actividades.

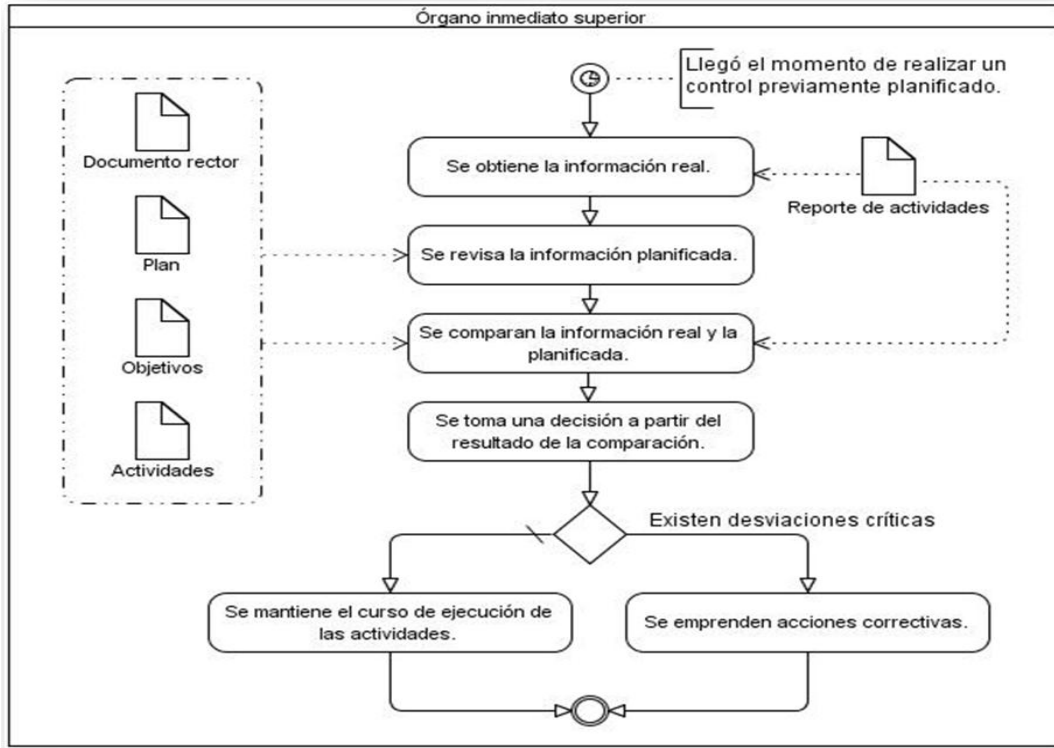
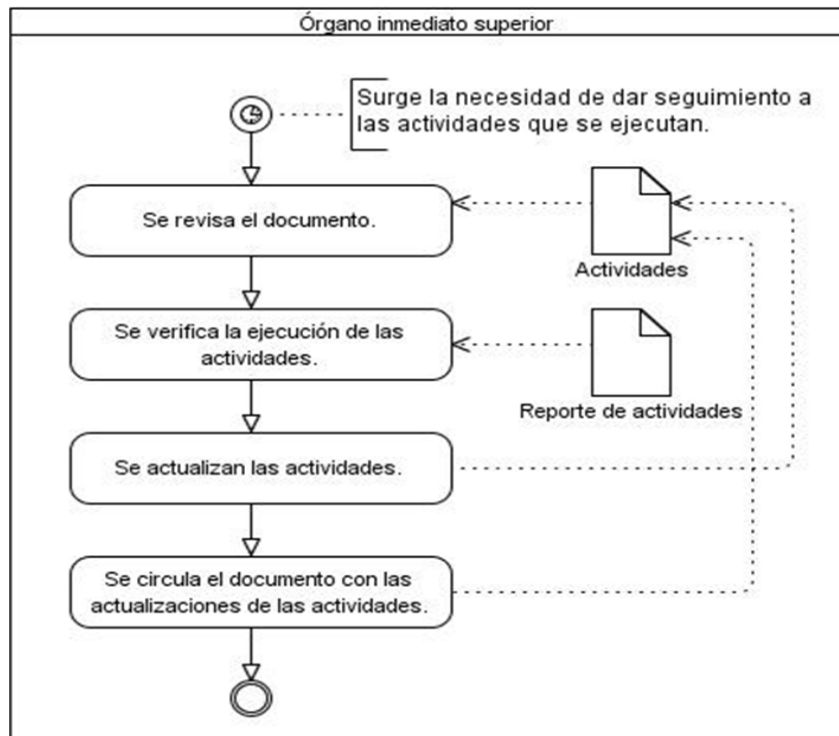


Figura 5: Diagrama del proceso de negocio Seguimiento a las actividades.



2.3.2. Especificación de procesos de negocio.

Tabla 1: Proceso Registro de actividades.

Objetivo	Definir y aprobar las actividades que tributan al plan que define la entidad.
Evento(s) que lo genera(n)	Se ha recibido documentación del nivel superior que orienta la planificación.
Pre condiciones	Se ha emitido una documentación que orienta la planificación en el nivel superior.
Marco legal	N/A.
Clientes internos	Establecer plan. Aprobar plan.
Clientes externos	N/A.
Entradas	Documento Rector (Documento Word). Planificación del Control (Documento Word). FIP (Documento Word).
Flujo de eventos	
Flujo básico	
1.	Se circula internamente el documento. El documento en formato impreso o digital se circula a todos los involucrados del mismo nivel de dirección que participarán en el establecimiento del plan.
2.	Se verifica el documento recibido. Este documento se identifica para conocer si es un Documento Rector, un FIP o un documento de Planificación de control. En caso de ser un documento de Planificación de control o un FIP los objetivos y/o actividades estarán bien definidos. En el caso de que se trate de un documento rector se procederá a ejecutar la actividad 3 del flujo básico: <u>Se estudia internamente el documento.</u> Para el primer caso, si el documento recibido tiene bien definidos los objetivos y las actividades se ejecuta el <u>flujo paralelo Definir actividades del plan</u> , pero si por el contrario el documento recibido tiene bien definidos solo las actividades entonces se ejecuta el <u>paso 3 del flujo paralelo Definir actividades del plan.</u>
3.	Se estudia internamente el documento. El documento es estudiado minuciosamente por parte de todos los involucrados del nivel de dirección que se trate para tener todos los elementos necesarios que ayuden a establecer el plan a este nivel.
4.	Se ejecutan los flujos paralelos <u>Definir actividades del plan</u> y <u>Elaborar documento rector de la entidad.</u>
5.	Se ejecuta el flujo paralelo <u>Aprobar actividades del plan.</u>
6.	Concluye el proceso.
Pos-condiciones	
1.	N/A.
Salidas	
1.	N/A.
Flujos paralelos	
Flujo paralelo 2 Definir actividades del plan.	
1.	Se proponen las actividades que tributan a los objetivos previamente formulados. Cada implicado hace su propuesta de actividades que tributen a los objetivos

-
- correspondientes.
-
2. Se negocian/concilian las actividades. Las actividades que han sido propuestas por los implicados se llevan a un proceso de concertación entre ellos para definir las que finalmente serán incluidas en el plan.

 3. Se inscriben las actividades. Las actividades que han sido conciliadas se inscriben en el plan. Estas tributarán a sus correspondientes objetivos como se definió durante la actividad de conciliación de las actividades.

 4. Se concluye la definición de las actividades y con ello la elaboración del plan. Se conforma el documento final que recoge el plan para posteriormente ser sometido a aprobación por el responsable de aprobar. El jefe del órgano de dirección a este nivel conforma el documento rector para ser entregado a los subordinados del nivel inferior.

 5. Concluye el proceso.

Pos-condiciones

1. Se ha establecido el plan.

Salidas

1. Plan (Documento Word).

Flujos paralelo 5 Elaborar documento rector de la entidad.

1. El jefe del órgano conforma el documento rector para el nivel inmediato inferior. A medida que se van definiendo las áreas de resultados clave, los objetivos y las actividades el jefe del órgano elabora un documento rector propio de la entidad y que será entregado al nivel inmediato inferior para que este a su vez defina su documento rector y su plan.

2. Continúa en el paso 4 del flujo alterno Definir plan.

Pos-condiciones

1. Se ha elaborado documento rector de la entidad.

Salidas

1. Documento Rector de la Entidad (Documento Word).

Flujos paralelo 5 Aprobar actividades del plan.

1. Se revisa el documento con las actividades definidas para su aprobación. Los funcionarios del órgano de planificación leen el documento emitido. Se estudian además el documento rector porque sobre la base de este es que se han formulado las actividades que se desean aprobar.

2. Se emite un dictamen de validación. Se emite un dictamen que valide el cumplimiento de las normas establecidas por el Documento Rector en el documento recibido. El dictamen aborda además la calidad y objetividad del documento recibido. En caso de rechazar el documento se inscriben las recomendaciones pertinentes.

3. Si se valida el documento con las actividades del plan, se agrupan todos los documentos que se aprueban en este nivel de dirección. Se agrupan todos los documentos que como resultado de la revisión están listos para someterlos a aprobación hasta tanto se reciba la totalidad de los documentos que se aprueban en este nivel de dirección. Si no se aprueban ver el flujo alternativo 3.a.

4. El responsable de aprobar recibe el documento con las actividades. El documento con las actividades es recibido por el responsable del órgano de planificación junto con los documentos que se aprueban en este nivel de dirección.

5. El responsable revisa el documento a aprobar. El responsable se estudia el documento recibido con el dictamen correspondiente.
6. El responsable emite un criterio sobre el documento a aprobar. Se emite un criterio del documento por parte del responsable de aprobarlo, puede decidir aprobarlo o rechazarlo. Si rechaza el contenido del documento, el responsable elabora un dictamen con las recomendaciones para corregir el documento. Si se desaprueban ver el flujo alternativo 6.a.
7. El responsable aprueba el documento. El responsable está de acuerdo con el contenido del documento que se estudió y lo confirma mediante su firma.
8. Se envía el documento aprobado al organismo que lo originó. El órgano de planificación envía los documentos los órganos de dirección que los generaron.
9. Concluye el proceso.

Pos-condiciones

1. Se ha aprobado el documento.

Salidas

1. Documento Rector de la Entidad (Documento Word).
2. Actividades. (Documento Word).
3. Plan. (Documento Word).

Flujos alternos

Flujo alternativo 3. a Se rechazan actividades del documento

1. Se envía el documento al órgano de dirección que lo originó. El documento rechazado se envía al órgano de dirección que lo originó con las recomendaciones contenidas en el dictamen, para que pueda ser arreglado y ser sometido nuevamente al proceso de aprobación.
2. Concluye el proceso.

Pos-condiciones

1. Se ha rechazado el documento y devuelto al órgano de dirección que lo originó.

Salidas

4. Documento rector. (Documento Word).
5. Actividades. (Documento Word).
6. Plan. (Documento Word).

Flujo alternativo 6. a El responsable desaprueba el documento.

1. Si el responsable de aprobar decide corregir el documento actualiza el documento. En caso de que se deba enviar a quien lo originó se ejecuta el flujo alterno 3.a. Se rechazan actividades del documento.
2. El responsable de aprobar actualiza el documento. El responsable de aprobar realiza las modificaciones que considera necesario introducir en el documento. De esta forma emite la aprobación del documento.
3. El responsable solicita la presencia del que originó el documento. El responsable envía una notificación al que originó el documento para que éste conozca las actualizaciones en el documento original e indica que se presente para firmar el documento actualizado.
4. El jefe del órgano de dirección que originó el documento firma el documento actualizado. El jefe del órgano de dirección que presentó a la aprobación el documento se presenta para firmar el documento con las actualizaciones correspondientes.

5. Concluye el proceso.

Pos-condiciones

1. Se ha actualizado y aprobado el documento.

2. Se ha entregado el documento aprobado al órgano de dirección que lo originó con las firmas correspondientes.

Salidas

3. Documento rector. (Documento Word)

4. Actividades. (Documento Word)

5. Plan. (Documento Word)

Asuntos pendientes

1. N/A.

2.3.3. Validación del proceso de negocio.

El proceso de negocio referente al Registro, Control y Seguimiento a las actividades, como parte de los procesos globales Concepción del plan y Ejecución del plan dentro de la Planificación por Objetivo, se encuentra validado por la dirección del proyecto ERP-Cuba. Los documentos del Centro de Soluciones de Gestión concernientes al proceso de Planificación por Objetivos constituyen una constancia de la validación del proceso de negocio descrito.

2.3.4. Modelo Conceptual.

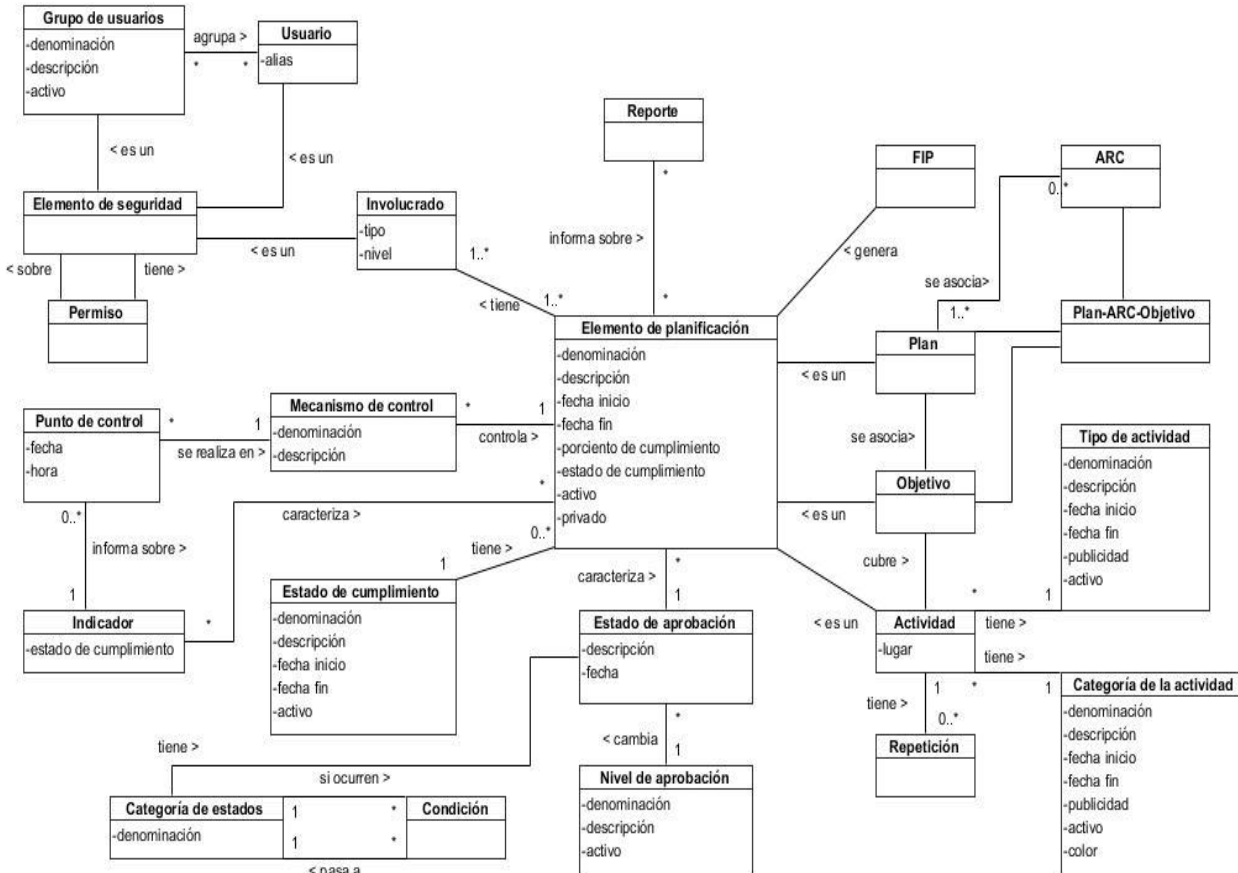


Figura 6: Modelo conceptual.

2.4. Requerimientos.

Un Requerimiento es:

- a) Una condición o capacidad requerida por un usuario para resolver un problema o alcanzar un objetivo.
- b) Una condición o capacidad que debe ser poseída por un sistema o componente del sistema para satisfacer un contrato, estándar, especificación, u otro documento formalmente impuesto.
- c) Una representación documentada de una condición o capacidad como las descritas en los incisos a) o b) (IEEE 610.12-1990)

2.4.1. Técnica de Captura de Requisitos.

La captura de requisitos es la actividad mediante la que el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema (35).

Para extraer los requisitos del sistema se utilizaron las siguientes técnicas:

Entrevistas: Se utiliza a la hora de realizar entrevistas con el cliente, dígase funcionales o especialistas del Subsistema PDO. Se realizaron preguntas con el objetivo de obtener toda la información posible sobre la visión que el entrevistado tiene de los requisitos y comprender los propósitos de la solución buscada.

Tormenta de ideas: Se realizan reuniones con todos los involucrados en la solución del problema, donde cada uno expresa sus ideas. Su objetivo fundamental es dar una visión general de las necesidades del sistema.

Talleres: También consiste en la realización de reuniones con los involucrados en el desarrollo del sistema, pero estas son sesiones más intensivas y estructuradas donde se realiza una preparación previa y es dirigida por un experto (en este caso el funcional o jefe de la factoría de PDO). Su objetivo fundamental es detallar cada requisito y sirve como base para la posterior especificación de los requisitos.

2.4.2. Listado de Requisitos Funcionales Identificados.

Requerimientos funcionales: Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer. (36)

A continuación se muestran los requisitos capturados para la primera etapa del desarrollo de la solución que se propone.

RF 1 Gestionar Actividad

- 1.1. Adicionar Actividad
- 1.2. Modificar Actividad
- 1.3. Eliminar Actividad
- 1.4. Gestionar Observaciones
- 1.5. Buscar Actividad
- 1.6. Duplicar Actividad
- 1.7. Establecer Relaciones
- 1.8. Listar Actividades

RF 2 Gestionar Categoría de la Actividad

- 2.1. Adicionar Categoría de la Actividad
- 2.2. Modificar Categoría de la Actividad
- 2.3. Eliminar Categoría de la Actividad
- 2.4. Desactivar Categoría de la Actividad
- 2.5. Activar Categoría de la Actividad
- 2.6. Listar Categorías de la Actividad

RF 3 Gestionar Tipo de Actividad

- 3.1. Adicionar Tipo de Actividad
- 3.2. Modificar Tipo de Actividad
- 3.3. Eliminar Tipo de Actividad
- 3.4. Desactivar Tipo de Actividad

RF 4 Repetición de Actividad

- 4.1. Adicionar Repetición Periódica
- 4.2. Adicionar Repetición Aleatoria
- 4.3. Modificar Repetición
- 4.4. Eliminar Repetición

3.5. Activar Tipo de Actividad

4.5. Listar Repetición

3.6. Listar Tipos de Actividad

RF 5 Gestionar Actividades en Calendario

RF 6 Generar Reporte de Actividades

5.1. Adicionar Actividad en el Calendario

(La implementación de este requisito

5.2. Modificar Actividad en el Calendario

responde al componente Reportes del

5.3. Eliminar Actividad en el Calendario

Subsistema PDO).

5.4. Buscar Actividad en el Calendario

5.5. Listar Actividades en Calendario.

2.4.3. Especificación de Requisitos Funcionales.

Para la especificación de requisitos se utiliza la plantilla definida por la dirección del proyecto ERP-Cuba, la cual presenta diferentes secciones para describir los flujos de eventos que conllevan a la ejecución dicha funcionalidad en el sistema.

A continuación se exponen la especificación del requisito Adicionar Actividad del paquete Gestionar Actividad.

Tabla 2: Especificación del requisito Adicionar Actividad.

Precondiciones	El usuario ha registrado al menos un plan, un FIP o un objetivo.
Flujo de eventos	
Flujo básico	
7.	Se selecciona el elemento de la planificación al que tributará la actividad.
8.	Se introducen los datos de la actividad: Denominación Categoría de la actividad Tipo de actividad Descripción Lugar Fecha de inicio Fecha de fin Hora de inicio Hora de fin Principal Porcentaje de cumplimiento
9.	El sistema permite seleccionar los involucrados en la actividad.
10.	El sistema permite introducir los participantes en la actividad.
11.	El sistema permite introducir los dirigentes de la actividad.
12.	El sistema permite gestionar las repeticiones que puede tener una actividad.

- 13. El sistema permite establecer relaciones de la actividad con otros elementos de la planificación registrados en el sistema.
- 14. El sistema valida (ver Validaciones) los datos introducidos.
- 15. Si los datos son correctos el sistema los registra.
- 16. El sistema confirma el registro de los datos.
- 17. Concluye el requisito.

Pos-condiciones

- 1. Se registró en el sistema una nueva actividad.
- 2. Se registraron nuevas relaciones entre las actividades y otros elementos de la planificación en el sistema.

Flujos alternativos

Flujo alternativo 8.a Información errónea

- 1 El sistema señala los datos erróneos y permite corregirlos.
- 2 El usuario corrige los datos.
- 3 Volver al paso 7 del flujo básico.

Pos-condiciones

- 1 N/A.

Flujo alternativo 8.b Información incompleta

- 1 El sistema señala los datos vacíos y permite corregirlos.
- 2 El usuario corrige los datos.
- 3 Volver al paso 7 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 CIG-ERP-N-DPO-i2201.

Relaciones	Requisitos Incluidos	
		El sistema permite establecer relaciones de la actividad con otros elementos de la planificación registrados en el sistema: Adicionar relaciones entre elementos de la planificación, en el agrupación

		<p>Gestionar relaciones entre elementos de la planificación. El sistema permite establecer relaciones de la actividad con otros elementos de la planificación registrados en el sistema: Eliminar relaciones entre elementos de la planificación, en la agrupación Gestionar relaciones entre elementos de la planificación. El sistema permite gestionar las repeticiones que puede tener una actividad: Adicionar repetición periódica, en la agrupación Gestionar repetición de actividades. El sistema permite gestionar las repeticiones que puede tener una actividad: Adicionar repetición aleatoria, en la agrupación Gestionar repetición de actividades. El sistema permite gestionar las repeticiones que puede tener una actividad: Modificar repeticiones, en la agrupación Gestionar repetición de actividades. El sistema permite gestionar las repeticiones que puede tener una actividad: Eliminar repeticiones, en la agrupación Gestionar repetición de actividades. El sistema permite gestionar las repeticiones que puede tener una actividad: Duplicar repeticiones, en la agrupación Gestionar repetición de actividades.</p>
	Extensiones	N/A.
Conceptos	Actividad	<p>Visibles en la interfaz: Lugar Hora de inicio Hora de fin Participantes Dirigentes Utilizados internamente: N/A.</p>
	Elemento de la planificación	<p>Visibles en la interfaz: Denominación Descripción Fecha de inicio Fecha de fin Porcentaje de cumplimiento Estado de cumplimiento Utilizados internamente: Activo</p>
	Involucrado	<p>Visibles en la interfaz: N/A. Utilizados internamente: Tipo Nivel</p>
	Categoría de la actividad	<p>Visibles en la interfaz: Denominación Utilizados internamente: Activo Descripción Fecha de inicio Fecha de fin Porcentaje de cumplimiento</p>

		Estado de cumplimiento Privado Color
	Tipo de actividad	de Visibles en la interfaz: Denominación Utilizados internamente: Activo Descripción Fecha de inicio Fecha de fin Privado
Requisitos especiales	N/A.	
Asuntos pendientes	N/A.	

2.4.4. Validación de los requisitos funcionales.

Los requisitos del proceso de negocio referente al Registro, Control y Seguimiento a las actividades, se encuentran validados por la dirección del proyecto ERP-Cuba. Los documentos del Centro de Soluciones de Gestión concernientes al proceso de Planificación por Objetivos constituyen una constancia de validación del proceso de identificación y especificación de los requisitos funcionales descritos.

2.4.5. Requerimientos no funcionales.

Los requerimientos no funcionales son restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. (36) Es por esta razón que como la solución propuesta constituye un subconjunto del sistema CedruX se asumen de manera transitiva los requisitos no funcionales definidos por el equipo de Arquitectura para dicho sistema. A continuación se especifican los requisitos no funcionales asociados a los componentes implementados:

1. Usabilidad (USB)

- 1.1. El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras.

2. Rendimiento (REN)

2.1. Se intercambian datos con el sistema, ya sea mediante acciones de inserción, búsqueda, eliminación o modificación de datos, en un servidor de 1 GB de memoria RAM y se recibe la notificación de la acción realizada en un período de 0.1 a 0.7 segundos.

3. Seguridad (SEG)

3.1. La información que se maneje en el sistema estará protegida de acceso no autorizado, a partir de los diferentes roles de los usuarios que empleen el sistema.

3.2. Si se desea eliminar o modificar un elemento de la base de datos que está siendo utilizado por otro elemento que depende de él; el sistema no permite que este elemento sea eliminado. (Borrar en cascada²⁴) (Seguridad²⁵/ Integridad²⁶).

4. Interfaz (INU)

4.1. El sistema debe contar con una interfaz fácil, amigable, sencilla, permitiendo que los usuarios finales del mismo sean capaces de interactuar con este aún teniendo conocimientos básicos.

5. Aplicación de Estándares (EST)

5.1. La aplicación debe ser implementada siguiendo los estándares de documentación, implementación de interfaz de usuario y de lógica de negocio que han sido diseñados para el proceso de desarrollo de software del proyecto ERP-Cuba.

- ✓ ERP-ARQ Estándar para el diseño de interfaces v1.1.
- ✓ ERP-ARQ Estándar para el manual de usuario CedruX.
- ✓ CSG-CAL Estándar para la documentación.
- ✓ Estándar de codificación

²⁴ **Borrar en cascada:** En base de datos cuando se tiene una tabla *padre* y varias tablas hijos relacionadas y lo que se necesita es que al momento de borrar el padre se borren automáticamente los hijos.

²⁵ **Seguridad:** Atributo de calidad que indica el grado en que un acceso no autorizado (accidental o deliberado) se prevenga .

²⁶ **Integridad:** Atributo de calidad que indica la ausencia de alteraciones inapropiadas de la información.

2.5. Conclusiones del capítulo.

Los argumentos expuestos en el desarrollo del presente capítulo demuestran lo siguiente:

- ✓ Las condiciones actuales en las que se lleva a cabo el proceso de Registro, control y seguimiento a las actividades dentro de la planificación por objetivo, no favorecen la realización efectiva del mismo en las entidades nacionales.
- ✓ El diseño de la solución propuesta debe tener en cuenta elementos como la escalabilidad del sistema, confiabilidad, usabilidad, funcionalidad y accesibilidad, de modo que cubra las deficiencias presentadas en la ejecución actual del proceso.
- ✓ El empleo de las técnicas de captura de requisitos facilitó la elaboración del listado de requisitos funcionales, logrando identificar y especificar tanto los aspectos del proceso de negocio como los recursos de aplicación y de soporte, que definirán la estructura del sistema que se pretende desarrollar como propuesta de solución.

CAPÍTULO III: MODELACIÓN DEL SISTEMA

3.1 Introducción.

La descripción y fundamentación de la línea base de la arquitectura sobre la cual se concebirá la solución propuesta constituye el punto de partida del presente capítulo. Por consiguiente se precisan los estilos y corrientes arquitectónicas adoptadas, así como la estructura de empaquetamiento sobre la cual se inserta la solución que se formula. Además se transcriben los requerimientos funcionales señalados en el capítulo anterior a procedimientos meticulosamente diseñados para su posterior implementación. Como parte de la modelación del diseño de la solución se fundamentan los patrones de diseños empleados en su elaboración y se plasman los diagramas que según la notación UML describen las relaciones entre las clases del diseño, el modelo de datos y la comunicación entre los componentes del Subsistema PDO.

3.2 Arquitectura de la solución.

Rara vez las literaturas asociadas a la ingeniería informática convergen en la definición de “Arquitectura de Software”. Muchos son los autores que la conceptualizan tomando como punto de partida la definición que oficializa la asociación técnica-profesional IEEE en el estándar 1471-2000. En este documento se plantea que: “La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.” Entorno al referido concepto se expone la estructura arquitectónica que sostendrá la solución que se propone con el presente trabajo de diploma. Esta se encuentra definida según las orientaciones establecidas por el departamento de Tecnología del proyecto ERP-Cuba, que es el encargado del diseño y evolución de la arquitectura del sistema CedruX.

La arquitectura definida para sistema CedruX posee elementos del modelo estructuralista y de la corriente procesual del desarrollo arquitectónico. Se organiza fundamentalmente en tres vistas, las cuáles a su vez pueden desagregarse en otras vistas. La siguiente jerarquía muestra la organización utilizada para la representación de la arquitectura de software.

- ✓ Tecnología: Define la plataforma tecnológica sobre la cual se desarrolla el sistema, se especifican los marcos de trabajo y herramientas bases.
- ✓ Seguridad: Chequea e implementa todos los aspectos relacionados con el acceso a la aplicación, la modificación, lectura o eliminación de la información.
- ✓ Presentación: Conformar el aspecto visual *del sistema*, dígame colores, botones, vínculos y todos los elementos significativos de vista de la presentación.
- ✓ Sistema: Define los tipos de componentes que conforman el sistema, así como sus

características y composición estructural interna (vista vertical de la arquitectura).

- ✓ Integración: Se ocupa de los procesos de integración interna (entre componentes de un mismo proyecto) y externa (entre proyectos distintos), establece las definiciones, estándares, protocolos de comunicación y reglas de intercambio de información.
- ✓ Datos: Elabora todas las definiciones a nivel de datos, la integración de los distintos modelos, de los patrones, estándares y definiciones a este nivel.
- ✓ Desarrollo: Define de la plataforma tecnológica a utilizar en la confección del producto, así como la disponibilidad de los distintos servicios telemáticos.
- ✓ Despliegue: Identifica los requerimientos de dispositivos necesarios para la implantación del sistema, y los distintos escenarios de despliegue posibles. (37)

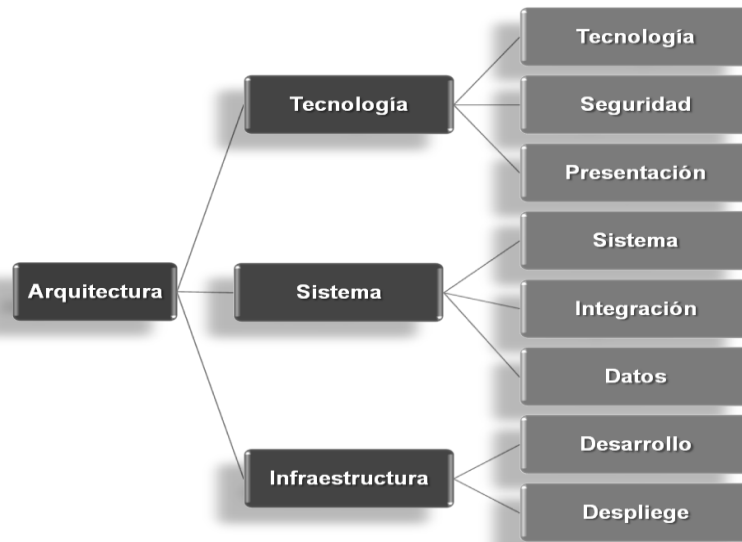


Figura 7: Organización de las vistas de la arquitectura del sistema CedruX.

3.2.1 Estilos arquitectónicos aplicados

Las singulares características del producto CedruX como sistema de gestión empresarial conllevó a definir una arquitectura híbrida pues resulta de la combinación de algunos estilos arquitectónicos. Se adopta el estilo arquitectónico orientado a componentes para el desarrollo horizontal del sistema, debido a complejidad que presenta el dominio de los negocios a incidir con el desarrollo de la aplicación.(37) Este enfoque posibilita alcanzar un mayor nivel de reutilización de las soluciones, así como una independencia en la ejecución de las pruebas a los componentes minimizando las afectaciones a otras partes del sistema y el mejoramiento de la calidad de la aplicación pues un componente como una unidad de composición de aplicaciones puede ser construido y perfeccionado continuamente.

Se optó por enriquecer esta perspectiva de arquitectura basada en componente con la incorporación de elementos como Registro de Servicios, Servicios, Descripción de servicios, propios de las

Arquitecturas Orientadas a Servicios (SOA), lo que permite responder rápidamente a las condiciones cambiantes del negocio empresarial, promoviendo y permitiendo la reutilización, la interconexión de tecnologías existentes en vez de consumir tiempo y costos en la reinversión. (38) La arquitectura del sistema CedruX es también basada en capas aunque en las capas superiores, es decir, en la capa de Presentación y la de Control o Negocio, se implementa un Modelo Vista Controlador (MVC) que controla el flujo de datos entre la interfaz de usuario y la lógica de negocio. Está compuesta básicamente por cinco niveles o capas:

1. **Capa de Presentación:** En esta capa se emplea las facilidades que brinda el Marco de Trabajo ExtJS para la construcción de interfaces amigables a la vista de los usuarios.
2. **Capa de Control o Negocio:** En esta capa se emplea el patrón de arquitectura Modelo Vista Controlador (MVC). En esta capa se realiza el tratamiento de la seguridad a nivel de aplicación.
3. **Capa de Acceso a Dato:** En esta capa estará presente el ORM (Object Relational Mapping) Doctrine, como Marco de Trabajo de persistencia para la comunicación con el servidor de datos mediante PDO²⁷, también consta de un Persistidor de Configuración que es el encargado de comunicarse vía XML con los Ficheros de Configuración del sistema denominado FastResponse.
4. **Capa de Dato:** En esta capa estará ubicado como servidor de base de datos PostgreSQL y un conjunto de Ficheros de Configuración de la arquitectura tecnológica.
5. **Capa de Servicio:** En esta última capa se encuentran todos los subsistemas que prestan y consumen servicios entre sí. (19)

La combinación de los estilos arquitectónicos descritos posibilita: la ejecución de desarrollos paralelos: en cada capa, la construcción de una aplicación más robusta debido al encapsulamiento, un mantenimiento y soporte más sencillo: pues resulta más fácil cambiar un componente que modificar íntegramente una aplicación. Proporciona una mayor flexibilidad: se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad y una alta escalabilidad: pues puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware debido a ser un sistema distribuido. (38)

3.2.2 Estructura de empaquetamiento

Todas las funcionalidades modeladas en las fases de negocio y requerimientos quedan contenidas en al menos un componente, y las distintas interacciones entre estos componentes originan

³¹**PDO:** (en inglés: PHP Data Objects) es una extensión de PHP que define una interfaz de acceso a datos que permite la conexión a diferentes bases de datos utilizando tecnología orientada a objetos. (31)

funcionalmente la existencia de módulos o subsistemas.

Niveles de empaquetamiento que componen la arquitectura del sistema CedruX:

1. **Nivel Sistema.** El nivel de sistema es el TODO. Está compuesto por todos los subsistemas, y es la vista global del producto.
2. **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.
3. **Nivel Módulo.** Este nivel surge para empaquetar un 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.
4. **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.

3.3 Patrones de diseño empleados en la solución propuesta.

Los patrones de diseño especifican la estructura y el comportamiento de una sociedad de clases. Estos son directrices y principios estructurados que describen un problema común y presentan soluciones simples. Están basados en la experiencia y se ha demostrado que funcionan. (39)

Los Patrones de Diseño empleados se concretaron durante la elaboración del diseño de las clases y las relaciones entre ellas, con el propósito de garantizar un bajo acoplamiento y alta cohesión entre las clases, así como un correcto encapsulamiento de las clases y la adecuada distribución de responsabilidades entre ellas.

3.3.1 Patrones GRASP

GRASP: (General Responsibility Assignment Software Patterns) se encargan de asignar responsabilidades a los objetos en sentido general, las principales responsabilidades son conocer (atributos, relaciones con otros objetos) y hacer (tareas que debe cumplir cada objeto). (40)

Entre los principales patrones GRASP utilizados se encuentran:

➤ Experto

Este patrón indica cual es la clase que debe asumir una determinada responsabilidad (clase experta), teniendo en cuenta la información que porta para cumplirla. Este se evidencia en la definición de las clases según las funcionalidades que realizan, como por ejemplo la clase GestionarActividadesModel que es la encargada de realizar las funcionalidades de insertar, actualizar y eliminar actividades.

➤ Creador

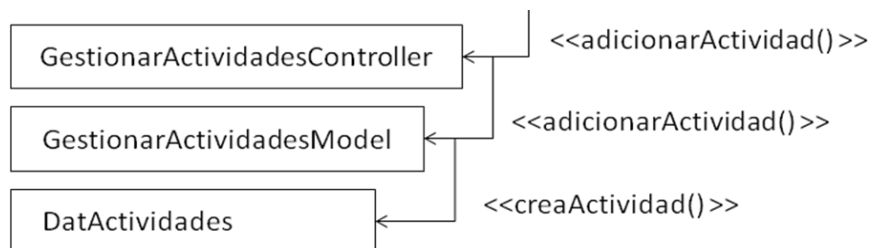
El patrón Creador resuelve el problema de quien debe crear una instancia de alguna clase, recomienda que una clase **B** debe crear una instancia de **A** si:

- ✓ B agrega los objetos de A.
- ✓ B contiene a los objetos de A.
- ✓ B registra las instancias de los objetos de A.
- ✓ B utiliza específicamente los objetos de A.
- ✓ B tiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado.

Este patrón se evidencia en la clase GestionarActividadesController, la cual es la responsable de crear instancias de la clase GestionarActividadesModel, para así utilizar las funcionalidades de esta.

➤ **Bajo Acoplamiento**

Este patrón es muy importante para el sistema, debido a que resuelve el problema de cómo lograr baja dependencia, alta reutilización entre las clases y reducir el impacto de los cambios. La siguiente figura ejemplifica como se empleó este patrón en el diseño de clases.



Se puede evidenciar que solo se le atribuye la responsabilidad de crear objetos de DatActividad a la clase GestionarActividadesModel evitando así que el número de recurrencias de la clase GestionarActividadesController a otras clases sea mayor.

➤ **Alta Cohesión**

Con una alta cohesión dentro de cada clase definida se puede obtener una medida de la fuerza que une a las responsabilidades de una clase, resolviendo el problema de cómo lograr que la complejidad sea lo más manejable posible. Como consecuencia del uso de este patrón se logra un incremento de la claridad y comprensión, lo que simplifica el mantenimiento. Este patrón se aplica de manera general en el diseño del Subsistema PDO, al agrupar las clases en dependencia de los requerimientos y los procesos que se necesitaban informatizar, de modo que cada clase implementa las operaciones que se encuentran en su misma área funcional, de esta forma GestionarActividadesModel solamente se encarga de la creación y modificación de actividades, así como otras operaciones que se realizan sobre esta área y no de otro tipo de operación.

➤ **Controlador**

El patrón Controlador define quién debe responder a determinados eventos. Una clase controladora debe ser la encargada de manejar los eventos dentro del componente, así la aplicación del patrón conlleva a separar la lógica de negocios de la capa de presentación, al aplicar estos principios, el controlador no realiza las actividades mencionadas sino que las delega en otras clases, los modelos, con las que mantiene un modelo de alta cohesión. Las clases

GestionaractividadesController y GestionarCalendarioController son un ejemplo de la aplicación de este patrón.

3.3.2 Patrones GoF

GoF: (Gang of Four) llamados así por los cuatro autores del libro Patrones de Diseño que recoge alrededor de 23 patrones de los más utilizados. Están clasificados según su propósito:

- ✓ Creación: Tratan la creación de instancias.
- ✓ Estructura: Tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad.
- ✓ Comportamiento: Tratan la interacción y cooperación entre clases. (41)

Entre los principales patrones GoF utilizados se encuentran:

➤ Fachada

Este patrón simplifica el acceso a un conjunto de objetos relacionados mediante una interfaz de alto nivel que esconde la complejidad del sistema. Se utiliza para proporcionar un acceso sencillo a subsistemas muy complejos. 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 Actividades y que se encuentran, tanto en otros componentes pertenecientes al Subsistema Planificación por Objetivo, como en otros subsistemas externos. La utilización de la clase ConfiguracionServices como fachada para el acceso a determinadas funcionalidades del componente Configuración del Subsistema PDO es una muestra del empleo de este patrón.

➤ Cadena de Responsabilidades

El patrón Cadena de Responsabilidades permite establecer la línea que deben llevar los mensajes para que los objetos realicen la tarea indicada. Su aplicación principal en el diseño del sistema es en el tratamiento de Excepciones. Está concebido que ante la ocurrencia de un error al realizarse una determinada consulta a la base de datos el mismo sea manejado por el Modelo, creando una nueva excepción de tipo ZendExt_Exception. Dicha excepción debe ser propagada al Controlador, el cual será el encargado de capturarla y enviarla a la Vista ya traducida. Esta última por su parte mostrará un mensaje al usuario en un lenguaje entendible notificando el error y sin especificar detalles del mismo, distribuyéndose de esta manera las responsabilidades entre los diferentes componentes.

3.4 Diseño de la solución.

En el presente acápite se exponen elementos del modelo de diseño que describen la estructura en términos de componentes y clases de la solución que se pretende formular.

3.4.1 Diseño de la solución en términos de componentes.

El Subsistema Planificación por Objetivos del Sistema Integral de Gestión de Entidades CedruX es el encargado de interrelacionar objetivos de trabajo, actividades y recursos en tiempo real; garantizando el seguimiento del desarrollo y cumplimiento de los objetivos y tareas principales en las entidades. Como componente de dicho subsistema se integraría el Gestor de actividades, que abarcaría todas funcionalidades del paquete de requisitos Gestionar Actividad incluyendo además las del paquete Repetición de Actividad. Mientras que el componente Calendario se integraría como facilitador de este y comprendería las funcionalidades del paquete Gestionar actividades en Calendario. En el caso de los paquetes de requisitos auxiliares Gestionar Categoría de la Actividad y Gestionar Tipo de Actividad se anexan al componente Nomencladores.

La siguiente figura responde al modelo de componentes del Subsistema Planificación por Objetivos en el que se resaltan los dos componentes que conforman la solución que se propone.

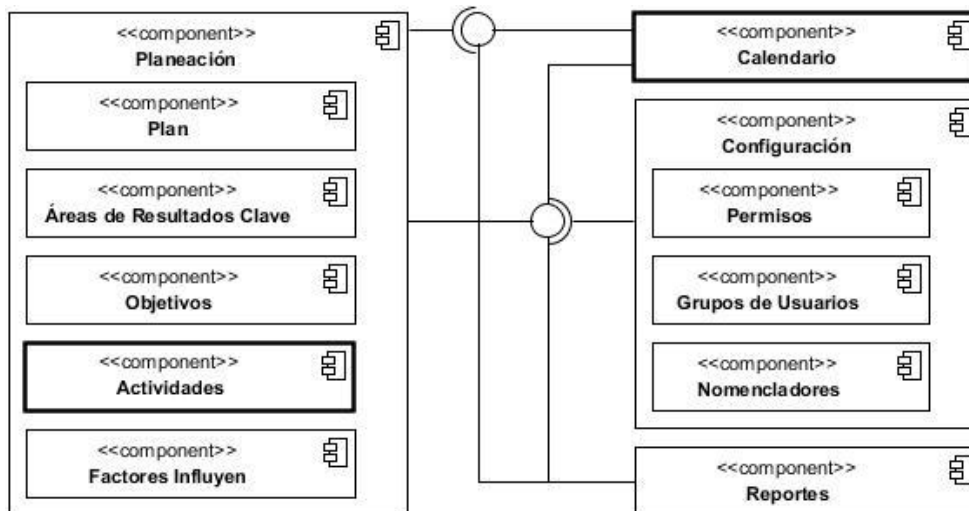


Figura 8: Diagrama de componentes del Subsistema Planificación por Objetivos.

Breve explicación del modelo de componentes del Subsistema Planificación por Objetivos.

El macro-componente Planeación es el encargado de la gestión de los elementos que inciden en la planificación y de establecer las relaciones entre ellos, mientras que el referido como Configuración es el responsable tanto de la gestión de los grupos de usuarios del sistema y los permisos respectivos como de la gestión de los nomencladores. Estos macro-componentes a través de sus respectivas interfaces de comunicación establecen un estrecho vínculo, debido a que no es posible concebir una gestión de los elementos que inciden en la planificación si previamente no se realizó una configuración de los grupos de usuarios que intervendrá en la planificación de la entidad y los permisos asociados a estos, así como de los nomencladores que se emplearán durante el periodo que se desea planear. Por otra parte los componentes medulares Reportes y Calendario adoptan una condición receptora en cuanto al intercambio de información se refiere.

3.4.2 Diseño de clases.

La definición de las clases que intervendrán en el desarrollo de la solución que se propone y las relaciones entre ellas constituye un elemento significativo para la transición a la fase de construcción. Por tal motivo en el posterior acápite se presentan los diagramas de clases que según la notación UML son una abstracción del dominio donde a partir de una estructura estática se representan los requerimientos a través de las clases del sistema y sus interrelaciones, mostrando en términos generales qué debe hacer el sistema.

3.4.2.1 Diagramas de clases del diseño

A continuación se muestran las propiedades y procedimientos asociados a cada clase del diseño y las relaciones entre ellas, facilitando su posterior implementación en los lenguajes adoptados para el desarrollo de la solución propuesta. Se especifica el diagrama de clases de componente Gestor de Actividades.

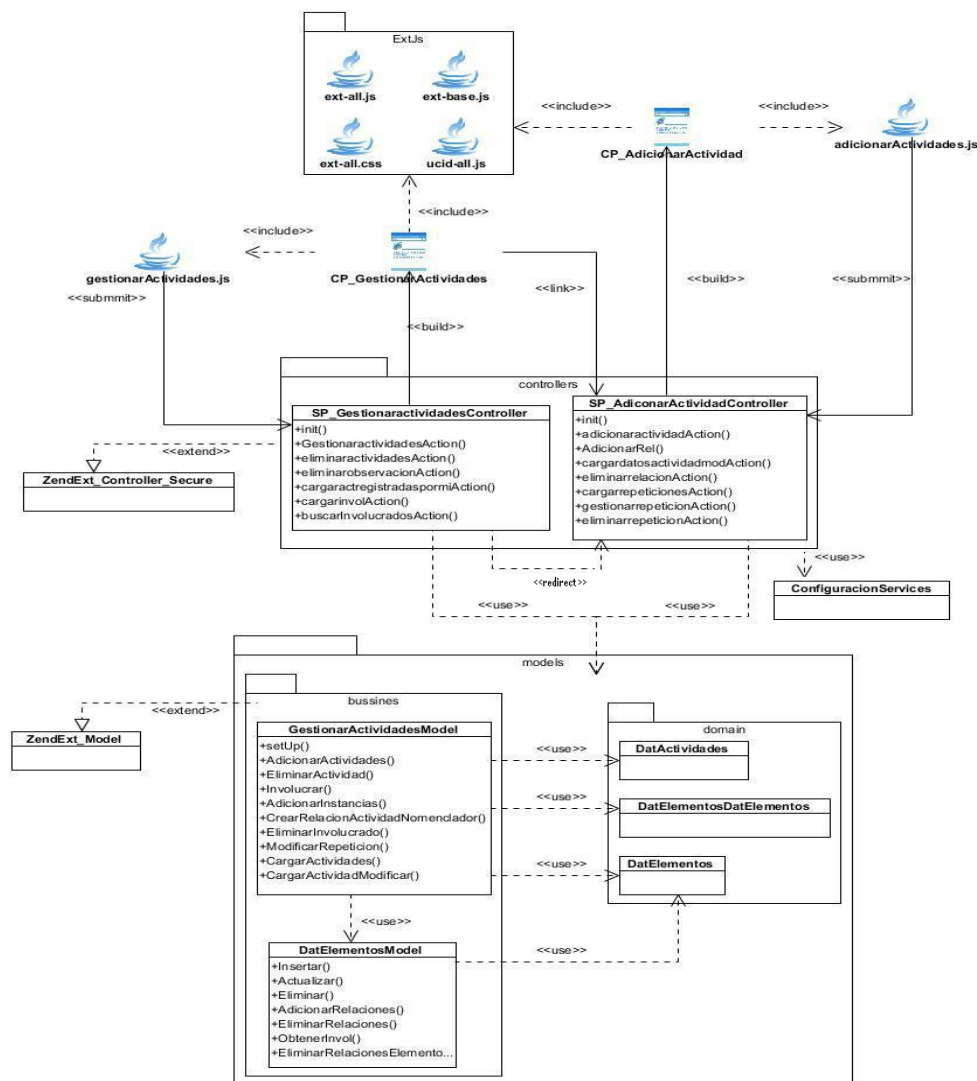


Figura 9: Diagrama de clases del componente Gestor de Actividades.

Tabla 3: Descripción del diseño de clases asociadas al componente Gestor de Actividades.

Elementos representados	Descripción
Librería ExtJS	Contiene los componentes generados a través de la librería JavaScript ExtJS necesarios para visualizar el contenido de la presentación.
CP_GestionarActividades	Página encargada de visualizar, a través de los archivos de extensión <i>js</i> que debe incluir, la información relacionada con la gestión de las actividades que son asignadas a los integrantes de una organización.
CP_AdicionarActividad	Página encargada de visualizar al usuario los datos necesarios para adicionar y modificar las actividades que se le son asignadas.
gestionarActividades.js	Script encargado de generar de forma dinámica a través del DOM y utilizando la librería ExtJS, los componentes de la vista necesarios para mostrar las actividades a los usuarios, así como las operaciones que se pueden realizar sobre ellas. Debe enviar y recibir los datos de la controladora utilizando tecnología AJAX.
adicionarActividades.js	Script encargado de generar de forma dinámica a través del DOM y utilizando la librería ExtJS, los componentes de la vista necesarios para que el usuario adicione y modifique las actividades que le son asignadas. Debe enviar y recibir los datos de la controladora utilizando tecnología AJAX.
SP_GestionaractividadesController	Contiene la clase controladora encargada de ejecutar las diferentes funcionalidades que devuelven los datos para visualizar los elementos relacionados con las actividades, según las peticiones del usuario.
SP_AdiconarActividadController	Contiene la clase controladora encargada de ejecutar las distintas funcionalidades relacionadas con la inserción y modificación de las actividades.
ZendExt_Controller_Secure	Clase encargada de gestionar acciones personalizadas y está integrada a la seguridad.
GestionarActividadesModel	Clase del Negocio encargada de manejar los datos persistentes dentro del componente, contiene las funcionalidades referentes a la gestión de las actividades.
DatElementosModel	Clase del Negocio encargada de manejar los datos persistentes dentro del componente, contiene las funcionalidades que permite tratar a las actividades como un Elemento de la Planificación.
ZendExt_Model	Modelo gestor del negocio que permite entre otras funcionalidades iniciar las conexiones a la base de datos.
Paquete Domain	Contiene las clases que representan las tablas de la base de datos, incluyen todas las funcionalidades referentes al acceso a los datos mediante un lenguaje de consultas propio del Doctrine.

3.4.2.2 Modelos de datos

Un modelo de datos es un lenguaje utilizado para la descripción de una Base de Datos. Por lo general, permite describir las estructuras de datos de la base, las restricciones de integridad y las operaciones de manipulación de los datos. (42)

El modelo de datos diseñado para el Subsistema PDO se ajusta totalmente a las necesidades de almacenamiento de la información de la solución que se propone. Esto es posible debido a la dependencia del proceso de negocio Registro, Control y Seguimiento a las actividades con los macro-procesos Concepción y Ejecución del plan. Estos últimos fueron diseñados bajo una visión integracionista, por lo tanto se previó durante el desarrollo de dicho subsistema la inclusión de la gestión de las actividades como elemento de planificación. Para su construcción se tuvo en cuenta la persistencia de campos resúmenes para agilizar recuperaciones frecuentes de algunos datos. Además de garantizar mediante su estructura relacional una fácil desagregación de los datos que se manejan en el proceso de negocio en cuestión.

Dicho modelo de datos se encuentra en tercera forma normal contando con un total de 31 tablas. Las tablas resaltadas con el borde oscuro en la siguiente figura son las responsables de almacenar la información principal referente a los elementos básicos que inciden en la planificación, dígase Plan, Objetivo, FIP y Actividades. Específicamente la tabla Dat_elementos aglutina las propiedades comunes de estos elementos de planificación. En el conjunto de tablas con el borde discontinuo se destacan las que se relacionan con la gestión grupos de usuarios y la asociación de los permisos respectivos. El resto de las tablas corresponden a características y relaciones propias de los elementos de la planificación.

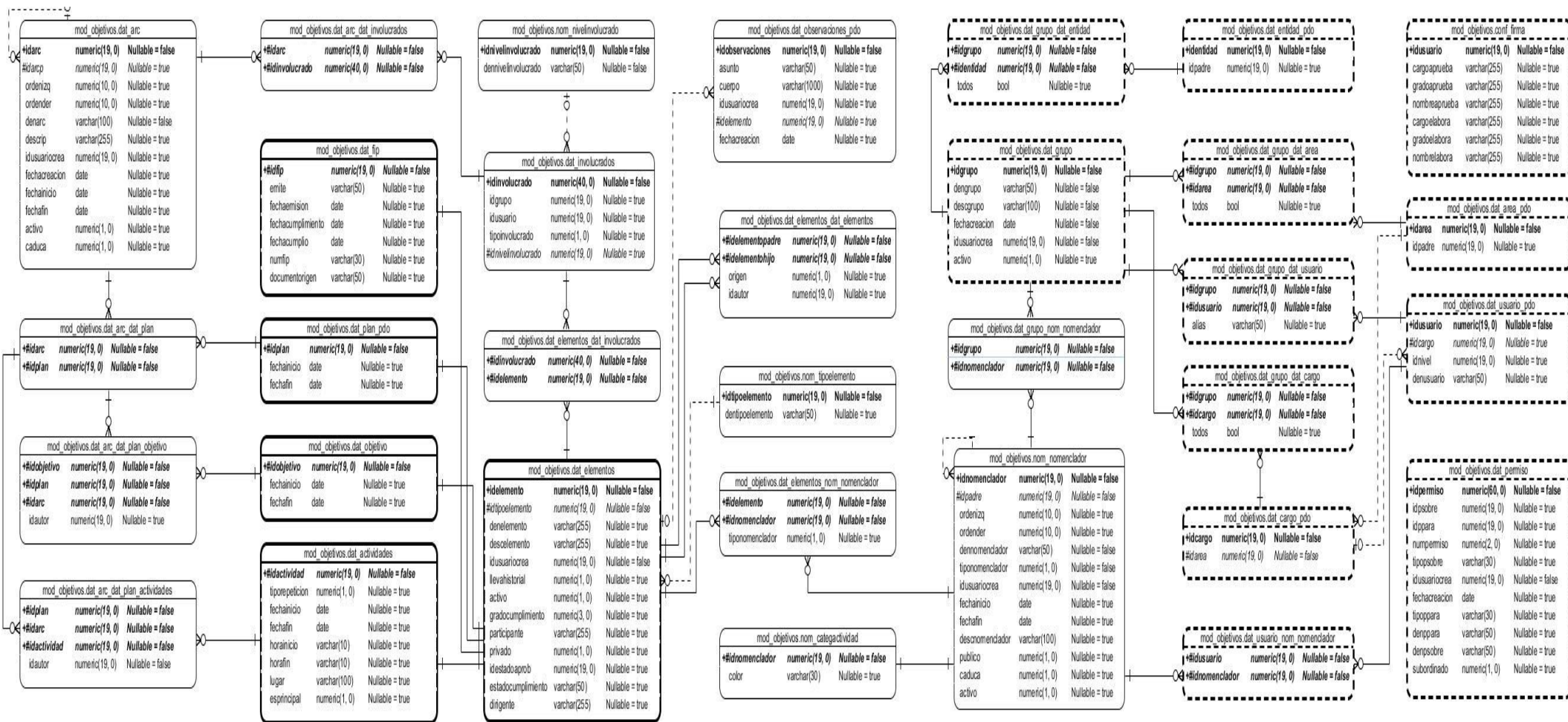


Figura 10: Modelo de datos del Subsistema Planificación por Objetivos.

3.4.3 Estrategia de integración.

La comunicación entre las distintas partes del sistema es un elemento de vital importancia, esta debe ser definida y estandarizada para garantizar la integridad y seguridad del flujo de datos que va desde la vista hacia el modelo y viceversa, así como la integración entre componentes, esta responde completamente a una estrategia de integración vertical concebida sobre 4 nodos y a partir de cada uno de los elementos arquitectónicos definidos.

A continuación se describe en cada uno de los lugares donde se ubican dichos nodos:

Nodo 1 (Vista – Controlador): Este se sitúa entre la vista y el controlador. Los datos recogidos en un formulario son enviados al Controlador haciendo uso del protocolo de comunicación HTTP a través del método “post” para ser procesados y los resultados son enviados por el controlador a la vista en un JSON a través del método “echo”.

Nodo 2 (Controlador – Modelo): Este está entre el controlador y el modelo. El Controlador toma los datos recibidos desde la vista, instancia una determinada clase del modelo y llama a uno de sus métodos, pasándole como parámetros los datos recibidos.

Nodo 3 (Modelo – Doctrine): Vincula el modelo con el marco de trabajo Doctrine. El Modelo utiliza llamadas a métodos de Doctrine que le permitan crear, modificar, eliminar o actualizar los datos almacenados en las tuplas de la base de datos.

Nodo 4 (Doctrine – Base de Datos): Se encuentra entre el Doctrine y la base de datos. Doctrine ejecuta las consultas a la Base de Datos utilizando programación orientada a objetos.

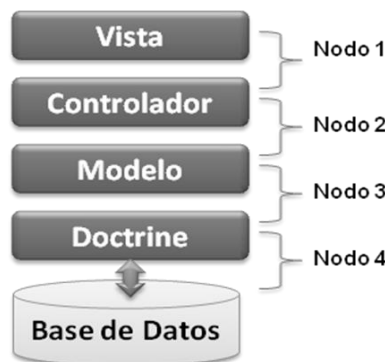


Figura 11: Estrategia de integración.

Dentro de un mismo componente la comunicación se establece de manera directa (a partir de la instanciación de las clases que lo integran), pero a la hora de crearla entre los distintos componentes y subsistemas se ejecuta a través del empleo de un registro de datos de los subsistemas que contiene la referencia a las distintas funcionalidades o servicios que estos brindan. Este registro es un fichero.xml llamado Inversión de Control (IoC) que es mapeado por la clase ZendExt_IoC. Este registra las funcionalidades que ofrecen los métodos de las clases *services* de los componentes del sistema, especifica respuestas deseadas a sucesos o solicitudes de datos concretas, orden

necesario y el conjunto de sucesos que tienen que ocurrir según los parámetros requeridos para poder hacer uso de un determinado servicio.

Tabla 4: Servicios internos del Subsistema PDO gestión de las actividades y el calendario

Servicios que se reciben para la gestión de las actividades y el calendario		
Servicios internos del Subsistema PDO		
Para la gestión de las actividades		
Componente	Servicio	Descripción
Configuración	DameGruposDondeEstaUsuario	Dado un identificador de usuario los debe devolver los grupos al que pertenece.
	DameSubordinados	Debe devolver los subordinados del usuario en sesión.
	DameInvolucradosAdicionar	Debe retornar los involucrados agrupados por usuarios, subordinados, grupos y usuarios de nivel superior sobre los que se tienen permiso para adicionar actividades.
	GetTodosNomencladoresPorTipo	Debe devolver un listado de nomencladores según el tipo.
	DameDatosNomenclador	Debe retornar los datos de un nomenclador dado su identificador.
	DameDatosDeGrupo	Debe retornar los datos de un grupo dado un identificador.
	TienePermisoParaSobre	Debe retornar los permisos que tiene el usuario en sesión.
Para el calendario		
Componente	Servicio	Descripción
Configuración	DameArbolBuscarPorInvolucradoCalendario	Debe retornar un listado de involucrados dado un identificador de usuario.
	DameGruposDondeEstaUsuario	Debe devolver dado un identificador de usuario los grupos al que pertenece.
	GetTodosNomencladoresPorTipo	Debe devolver un listado de nomencladores según el tipo.
	DameSubordinados	Debe retornar los subordinados del usuario en sesión.
	DamePermisosLeer	Debe retornar un listado de usuarios con permiso de lectura

		sobre las actividades adicionales por el usuario en sesión.
Plan	DameActividadesPorInvolucrados	Debe devolver un listado de actividades asociadas a un conjunto de usuarios involucrados.
	DameActvPorUsuarioTipoNomenclador	Debe devolver un listado de actividades según el usuario y el tipo de nomenclador.
	EliminarActividad	Servicio que brindará la posibilidad de eliminar una actividad pasado su identificador.
	ModificarActivacionActividad	Servicio que brindará la posibilidad de activar o desactivar una actividad.
	ModificarHoraActividad	Servicio que brindará la posibilidad de modificar la hora de inicio y de fin de una actividad.

Servicios que se consumen de otros subsistemas

Para la gestión de las actividades

Componente	Servicio	Descripción
Configuración(ERP)	EstadosVerDoc	Devolverá una lista de estados en los que se puede encontrar un documento.
	DenomEstado	Devolverá la denominación de un estado según su identificador.
Seguridad	nombreUsuario	Devolverá el nombre de un determinado usuario

Para el calendario

Componente	Servicio	Descripción
Configuración(ERP)	EstadosVerDoc	Devolverá una lista de estados en los que se puede encontrar un documento.

Servicios que se brindan como parte de la integración.

Servicio	Descripción
DameDatosActividad	Servicio que permitirá conocer los datos de una actividad dado su identificador.
VerificarInvolucradoAct	Servicio que permitirá conocer si un usuario es involucrado en una determinada actividad.
DameActividadesPorInvolucrados	Devolverá un listado de actividades asociadas a un conjunto de usuarios involucrados.
DameActvPorUsuarioTipoNomenclador	Devolverá un listado de actividades según el usuario y el tipo

	de nomenclador.
EliminarActividad	Servicio que brindará la posibilidad de eliminar una actividad pasado su identificador.
ModificarActivacionActividad	Servicio que brindará la posibilidad de activar o desactivar una actividad.
ModificarHoraActividad	Servicio que brindará la posibilidad de modificar la hora de inicio y de fin de una actividad.
ObtenerActividadesPorDia	Devolverá un listado de actividades dada una fecha de inicio y de fin.
ObtenerActividadesReporte	Devolverá un listado de actividades para presentar en un reporte.
ObtenerActividadesPorAreas	Devolverá un listado de actividades pertenecientes a un área de resultados clave determinada.
ActividadesDirectasAlPlan	Devolverá un listado de actividades pertenecientes a un plan determinado.

3.4.4 Prototipos de interfaz de usuario.

Prototipos de interfaz de usuarios asociados al componente Gestor de Actividades.

Figura 12: Prototipo de interfaz de usuario del requisito Adicionar Actividad.

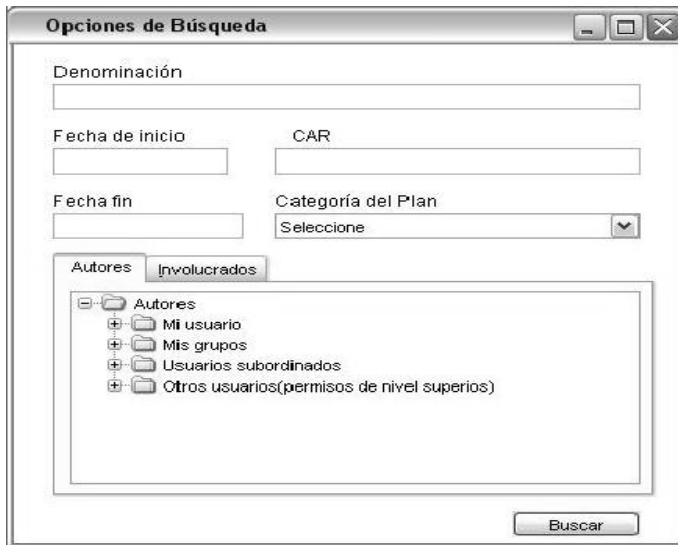


Figura 13: Prototipo de interfaz de usuario del requisito Buscar Actividad.

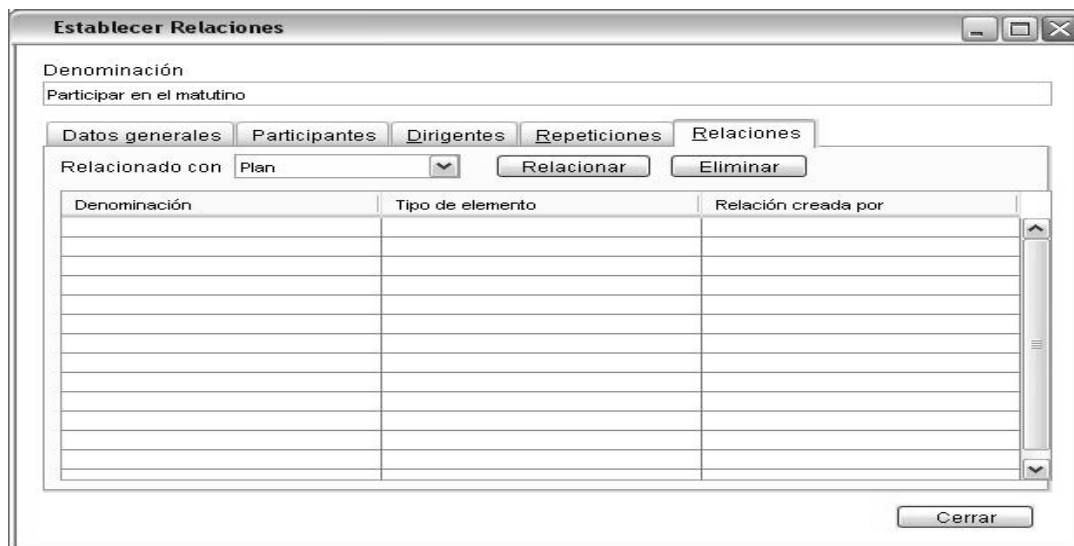


Figura 14: Prototipo de interfaz de usuario del requisito Establecer Relaciones.

3.5 Conclusiones del capítulo.

El presente capítulo fue crucial para la definición de la arquitectura de la solución que se propone y el diseño de los requerimientos funcionales señalados en el capítulo anterior. Se fundamentaron los patrones de diseño empleados. Estos permitieron especificar la estructura y comportamiento de las clases, garantizando presentar una solución que atiende los niveles de reutilización y mantenimiento. El diseño de la solución en términos de componentes y la estrategia de integración de los mismos permitió establecer las bases por la cuales se registrarán los desarrolladores para poder pasar a la implementación de la solución. Por consiguiente, una vez precisadas las funcionalidades y la estructura de la solución es posible el comienzo de la fase contigua a la Modelación: la Construcción del sistema.

CAPÍTULO IV: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

4.1 Introducción.

Como parte de la implementación de la solución propuesta, en el presente capítulo se expondrá la estructura física de la misma, los estándares que regirán el código fuente a implementar y los componentes necesarios para su despliegue. También se abarcará todo lo concerniente a la realización de las pruebas de aplicación y se expondrán las de métricas aplicadas al diseño propuesto en el capítulo anterior, en aras de garantizar calidad de dicha solución.

4.2 Implementación.

La Implementación comienza con el resultado obtenido del diseño detallado. Su objetivo principal es desarrollar la arquitectura y el sistema como un todo. De forma más específica, los propósitos de la Implementación son:

- ✓ Planificar las integraciones de sistema necesarias en cada iteración. Siguiendo para ello un enfoque incremental.
- ✓ Implementar clases, componentes y subsistemas encontrados durante el diseño.
- ✓ Integrar componentes. (34)

En los contiguos acápites se presenta la estructura física de la solución propuesta que coincide con la que define el marco de trabajo Sauxe y el diagrama de despliegue asociado a la misma.

4.2.1 Estándares de código.

Un estándar de código se basa en la estructura y apariencia física del código de un programa con el fin de facilitar la lectura, comprensión, mantenimiento del código, reutilización a lo largo del proceso de desarrollo de un software y no en la lógica del programa. Su objetivo fundamental es definir la nomenclatura de las variables, objetos, métodos y funciones, así como también tiene que ver con el orden y legibilidad del código escrito.

Partiendo de esto, se definen 3 partes principales dentro de un estándar de programación:

- ✓ Convención de nomenclatura: Es la forma de nombrar las variables, funciones y clases.
- ✓ Convenciones de legibilidad de código: Es la forma de organizar el código.
- ✓ Convenciones de documentación: Es la manera de establecer comentarios, la ayuda, entre otros.

➤ Nomenclatura de las clases:

Los nombres de las clases comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación *PascalCasing*, la cual define que los identificadores y nombres de variables, métodos y clases están compuestos por múltiples palabras

juntas, iniciando cada palabra con letra mayúscula y con sólo leerlo se reconoce el propósito de la misma.

1. Nomenclatura según el tipo de clases:

Clases controladoras: Las clases controladoras después del nombre llevan la palabra: "Controller".

Ejemplo: GestionarActividadesController

Clases de los modelos:

- ✓ Business (Negocio): Las clases que se encuentran dentro de Business después del nombre llevan la palabra: "Model". Ejemplo: GestionarActividadesModel.
- ✓ Domain (Dominio): Las clases que se encuentran dentro de Domain el nombre que reciben es el de la tabla en la base de datos. Ejemplo: DatActividades.
- ✓ Generated (Dominio base): Las clases que se encuentran dentro de Generated el nombre comienza con la palabra: "Base" y seguido el nombre de la tabla en la base de datos. Ejemplo: BaseDatActividades.

2. Nomenclatura de las funcionalidades:

El nombre a emplear para las funciones se escribe con la inicial del identificador en minúscula, en caso de que sea un nombre compuesto se empleará notación *CamelCasing* que es similar a la *PascalCasing* con la excepción de la primera letra.

Nomenclatura según la clase donde se encuentren las funciones:

En la clase controladora: Las principales funcionalidades de las clases controladoras se les pone el nombre y seguida la palabra: "Action". Ejemplo: eliminarActividadesAction().

En las clases de los modelos: Las funcionalidades se nombran de manera que al leerlo se identifique su propósito. Ejemplo de función: eliminarActividad().

3. Nomenclatura de los comentarios:

Los comentarios deben ser lo bastante claros y precisos de forma tal que se entienda el propósito de lo que se está desarrollando. En caso de ser una función complicada se debe comentar para lograr una mejor comprensión del código.

4.2.2 Diagrama de despliegue.

Seguidamente se describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de computo, dígame procesadores, dispositivos y protocolos.

Existen dos posibles escenarios para el despliegue de la aplicación:

- ✓ Diagrama de despliegue de escenario para PC cliente con disco.

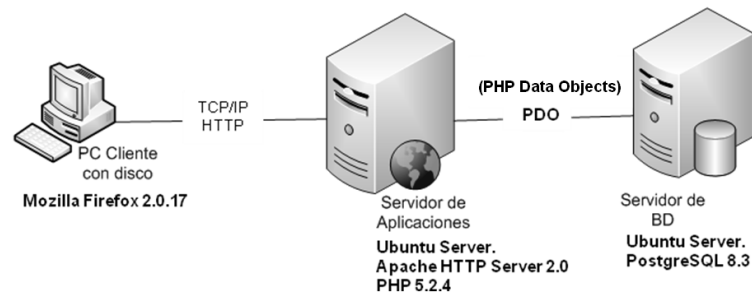


Figura 15: Diagrama de despliegue de escenario para PC cliente con disco.

- ✓ Diagrama de despliegue de escenario para PC cliente sin disco.

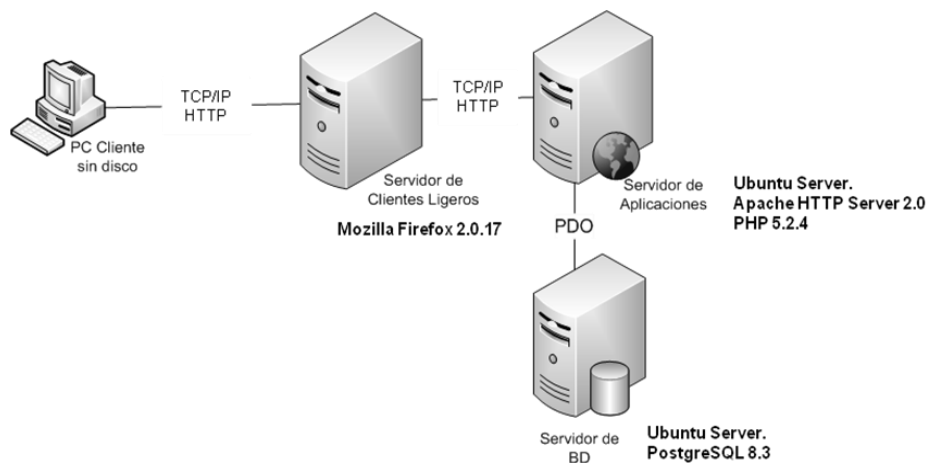


Figura 16: Diagrama de despliegue de escenario para PC cliente sin disco.

4.3 Validación de la solución propuesta.

A continuación se exponen los procedimientos empleados para la validación de la solución propuesta. Primeramente se presentan las métricas del diseño en los componentes lo que permite juzgar la calidad del diseño de los mismos. Luego se describen las pruebas de aplicación realizadas. Estas fueron ejecutadas con el objetivo de detectar y corregir el máximo de errores posibles.

4.3.1 Validación del diseño propuesto.

Las métricas para aplicaciones informáticas, no son perfectas; muchos expertos argumentan que se necesita más experimentación hasta que se puedan emplear bien las métricas de diseño. Sin embargo el diseño sin medición, en opinión de la Licenciada en Ingeniería en Sistemas Computacionales Heidi González Doria, es una alternativa inaceptable. A continuación se presentarán las métricas Tamaño operacional de clase (TOC) y Relaciones entre clases (RC), necesarias para evaluar la calidad del diseño propuesto a nivel de componentes. Ambas métricas no fueron aplicadas durante el diseño procedimental, sino que fueron retrasadas hasta tener disponible

el código fuente, pues la complejidad del negocio exigía la incorporación de diversas subrutinas necesarias para la implementación de cada procedimiento descrito en el diseño de clases.

Las métricas TOC y RC incluyen medidas de los siguientes atributos de calidad:

Responsabilidad: Responsabilidad que posee una clase en un marco conceptual correspondiente al modelado de la solución propuesta.

Complejidad del mantenimiento: Nivel de esfuerzo necesario para sustentar, mejorar o corregir el diseño de software propuesto. Puede influir significativamente en los costes y la planificación del proyecto.

Complejidad de implementación: Grado de dificultad que tiene implementar un diseño de clases determinado.

Reutilización: Significa cuán reutilizada es una clase o estructura de clase dentro de un diseño de software.

Acoplamiento: Dependencia o interconexión de una clase o estructura de clase respecto a otras.

Cantidad de pruebas: Número o grado de esfuerzo necesario para realizar las pruebas de calidad al producto (componente) diseñado. (43)

➤ **Métrica TOC: Tamaño operacional de clase.**

Se refiere al número de procedimientos existentes en una clase. Determina una relación directa entre los atributos Responsabilidad y Complejidad de implementación, sin embargo establece una relación inversa entre estos últimos y el atributo Reutilización.

Como resultado de la evaluación de la métrica TOC se obtuvo lo siguiente:

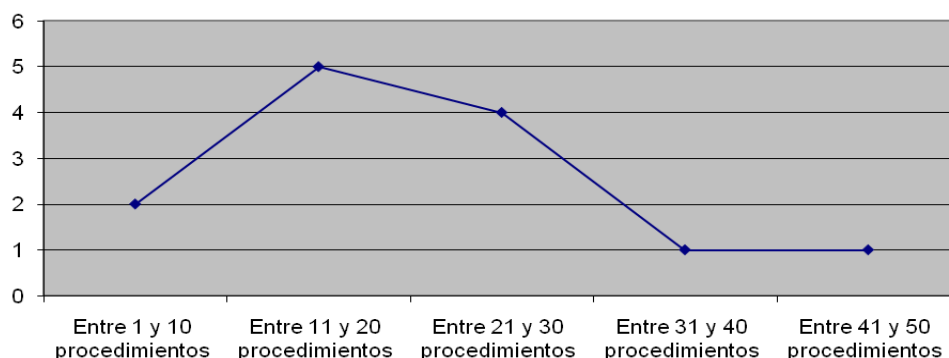


Figura 17: Representación de la cantidad de clase y el número procedimientos que contienen.

Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos:

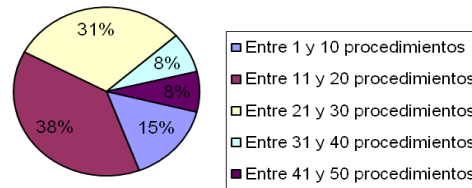


Figura 18: Representación en % de la cantidad de clase y el número procedimientos que contienen.

Representación en % de la incidencia de los resultados obtenidos en el atributo **Responsabilidad:**

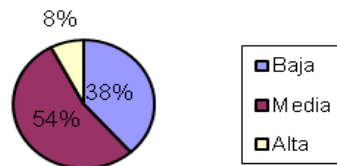


Figura 19: Representación del valor en % del atributo Responsabilidad.

Representación en % de la incidencia de los resultados obtenidos en el atributo **Complejidad de Implementación.**

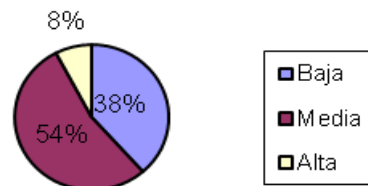


Figura 20: Representación del valor en % del atributo Complejidad de Implementación.

Representación en % de la incidencia de los resultados obtenidos en el atributo **Reutilización.**

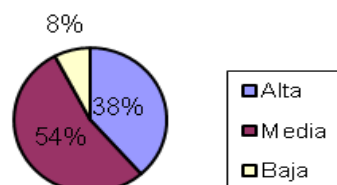


Figura 21: Representación del valor en % del atributo Reutilización.

Análisis de los resultados obtenidos en la evaluación de la métrica TOC

Considerando que el 53% de las clases contienen un número de funcionalidades inferior a la media de procedimientos por clases (cuyo valor asciende a 20) se puede afirmar que el diseño de clases elaborado se encuentra dentro de los límites de calidad. Es apreciable, dados los valores de los atributos de calidad, la complejidad de la implementación de las clases que intervienen en el diseño. Esto se debe a que muchos de los procedimientos no son atómicos lo que incrementa el

número de subrutinas necesarias para poder ejecutarlos. Por consiguiente, el 54% de las clases que intervienen en el diseño tienen un nivel medio de Reutilización.

➤ **Métrica RC: Relaciones entre clases.**

Se refiere al número de relaciones de uso de una clase. Determinada por los atributos: Acoplamiento, Complejidad de mantenimiento, Reutilización y Cantidad de pruebas, existiendo una relación directa con los tres primeros e inversa con el último antes mencionado.

Como resultado de la evaluación de la métrica RC se obtuvo lo siguiente:

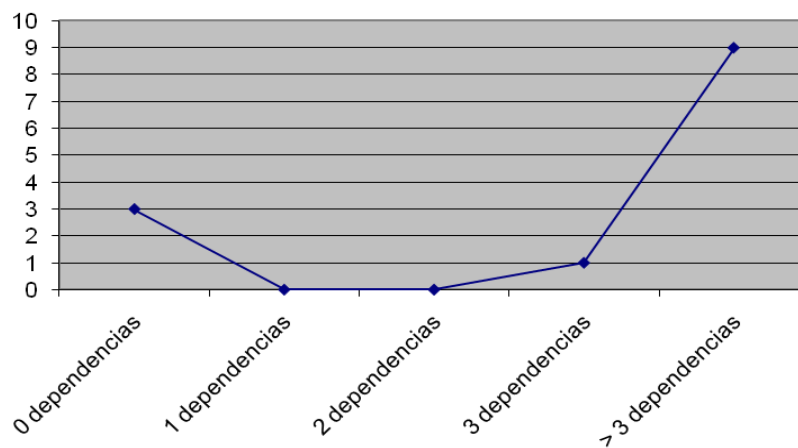


Figura 22: Representación de las asociaciones de uso por cantidad a de clases.

Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos:

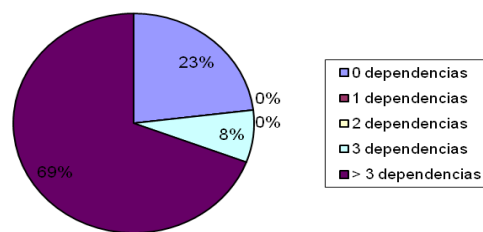


Figura 23: Representación en % de las asociaciones de uso por cantidad a de clases.

Representación en % de la incidencia de los resultados obtenidos en el atributo **Acoplamiento**:

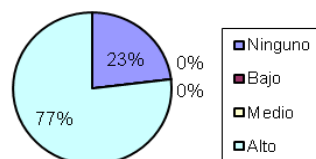


Figura 24: Representación del valor en % del atributo Acoplamiento.

Representación en % de la incidencia de los resultados obtenidos en el atributo **Complejidad de Mantenimiento**.

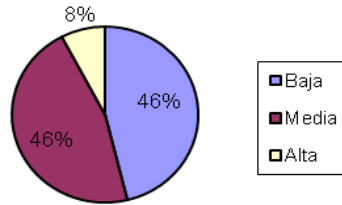


Figura 25: Representación del valor en % del atributo Complejidad de Mantenimiento.

Representación en % de la incidencia de los resultados obtenidos en el atributo **Cantidad de Pruebas**.

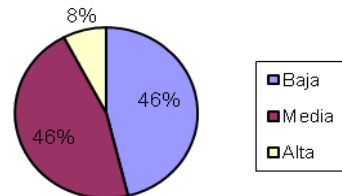


Figura 26: Representación del valor en % del atributo Cantidad de Pruebas.

Representación en % de la incidencia de los resultados obtenidos en el atributo **Reutilización**.

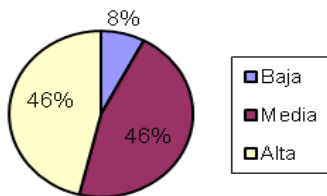


Figura 27: Representación del valor en % del atributo Reutilización.

Análisis de los resultados obtenidos en la evaluación de la métrica RC

Considerando que el promedio de asociaciones de uso entre las clases diseñadas es 16 y que el 63% de dichas clases contienen más de 3 asociaciones se puede afirmar que la solución que se propone presenta un elevado nivel de colaboraciones por clases. Por consiguiente, si el 77% de las clases presentan un alto nivel de acoplamiento, mayor será la complejidad de mantenimiento y la cantidad de comprobación que se produzcan cuando se efectúen modificaciones. En consecuencia el grado de reusabilidad será menor.

4.3.2 Pruebas de aplicación.

Las pruebas constituyen una etapa imprescindible durante el proceso de desarrollo del software. Su objetivo principal es asegurar que el software cumpla con las especificaciones requeridas y eliminar

los posibles defectos que este pudiera tener. De forma más específica los propósitos de las pruebas son:

- ✓ Realizar la validación del software desarrollado, entendiendo como validación el proceso que determina si el software satisface los requisitos.
- ✓ Realizar la verificación del software desarrollado, entendiendo como verificación el proceso que determina si los productos de una fase satisfacen las condiciones de la misma.

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. (44)

Niveles de Prueba:

A la hora de evaluar dinámicamente un sistema se debe comenzar por los componentes más simples y pequeños e ir avanzando progresivamente hasta probar todo el software en su conjunto. Las pruebas se aplican en distintos niveles de trabajo, dentro de estos se distinguen:

- 1. Pruebas de Unidad:** Prueba individual a las unidades separadas de un sistema de software.
- 2. Pruebas de Integración:** Los componentes individuales son combinados con otros componentes para asegurar que la comunicación, enlaces y los datos compartidos ocurran apropiadamente.
- 3. Pruebas del Sistema:** Son usualmente conducidas para asegurar que todos los módulos trabajan como sistema sin error. Es similar a la prueba de integración pero con un alcance mucho más amplio.
- 4. Pruebas de Aceptación:** Son realizadas principalmente por los usuarios con el apoyo del equipo del proyecto. El propósito es confirmar que el sistema está terminado, que desarrolla puntualmente las necesidades de la organización y que es aceptado por los usuarios finales.

Métodos de Prueba:

Un método de prueba es un enfoque sistemático, independiente del nivel en que se enmarque la prueba, que ayuda a encontrar buenos conjuntos de casos de prueba ²⁸ para detectar diferentes tipos de errores. (45)

- 1. Pruebas de Caja Blanca:** Se comprueban los componentes internos. Comprueba los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coinciden con el esperado o mencionado.
- 2. Pruebas de Caja Negra:** Se comprueban las funcionalidades sin tener en cuenta la estructura interna. Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, a través de los casos de prueba se demuestra que las funciones del software son operativas, que la entrada

²⁸ **Casos de prueba:** especifican una forma de probar el componente, incluye: la entrada, las condiciones bajo las cuales ha de probarse y los resultados esperados.

se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

Este tipo de pruebas permite encontrar:

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

Dentro de la prueba de caja negra se incluyen varias Técnicas de Pruebas tales como:

Partición de Equivalencia: Divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.

Análisis de Valores Límites: Prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.

Grafos de Causa-Efecto: Permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Pruebas realizadas a los componentes:

Para verificar el correcto funcionamiento de los componentes Gestor de actividades y Calendario se realizaron pruebas funcionales aplicando el método de caja negra, y específicamente la técnica de partición de equivalencia. Esta última divide el dominio de entrada de un programa en un número finito de variables de equivalencia. Las variables de equivalencia representan un conjunto de estados válidos y no válidos para las condiciones de entrada de un programa. Se definen dos tipos de variables de equivalencia, las válidas, que representan entradas válidas al programa, y las no válidas, que representan valores de entrada erróneos, aunque pueden existir valores no relevantes a los que no sea necesario proporcionar un valor real de dato.

Actualmente se están realizando pruebas de aceptación a los componentes desarrollados en el MINFAR y en el Consejo de Estado, con el propósito de confirmar que el sistema desarrolla puntualmente las necesidades de la organización. Hasta el momento solo se ha pedido incorporar nuevos requisitos para la segunda versión del producto. Sin embargo la versión actual, aún sin implementar la totalidad de los requerimientos suficientes para desarrollar a plenitud el proceso de Registro, Control y Seguimiento a las actividades, en opinión de los usuarios satisface la ejecución del proceso.

Casos de pruebas de caja negra generados al aplicar la técnica Partición de equivalencia:

Requisito a probar: 1.2 Adicionar actividad

Tabla 2: Caso de prueba “Adicionar actividad”.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
3.1. Adicionar actividad.	Se adiciona una actividad, con los siguientes atributos: Denominación Fecha de inicio Fecha de fin Porcentaje de cumplimiento Hora de inicio Hora de fin Lugar Categoría de la actividad Tipo de actividad Descripción Involucrados Participantes Dirigentes Principal	EP 1.1: Adicionar una nueva actividad introduciendo datos válidos.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar. - Se insertan todos los datos. - Se presiona el botón Aceptar
		EP 1.2: Adicionar una nueva actividad dejando campos requeridos en blanco.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar. - Se introducen los datos dejando campos requeridos en blanco. - Se presiona el botón Aceptar.
		EP 1.3: Adicionar una nueva actividad introduciendo errores en los datos.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar. - Se introducen los datos de la actividad, que se desea adicionar, introduciendo errores en los mismos. - Se presiona el botón Aceptar.
		EP 1.4: Cancelar	<ul style="list-style-type: none"> - Se presiona el botón Adicionar. - Se introducen, o no, datos. - Se presiona el botón Cancelar.

Tabla 6: Descripción de las variables.

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Denominación	Campo de texto	No.	Cadena de caracteres alfanuméricos. No permite caracteres especiales.
2	Fecha de inicio	Campo de fecha	No.	Es de tipo fecha con el formato: dd/mm/aaaa. La fecha de inicio debe ser menor o igual que la fecha de fin.
3	Fecha de fin	Campo de fecha	No.	Es de tipo fecha con el formato: dd/mm/aaaa. La fecha de fin debe ser mayor o igual que la fecha de inicio.
4	Porcentaje de cumplimiento	Campo de texto	Si	Combinación de números. Permite el Vacío.
5	Hora de inicio	Lista desplegable	Si	Combinación de números, letras y el carácter especial ":". Permite el Vacío.
6	Hora de fin	Lista desplegable	Si	Combinación de números, letras y el carácter especial ":". Permite el Vacío.
7	Lugar	Campo de texto	Si	Combinación de letras y números. Permite el Vacío.
8	Categoría de la actividad	Lista desplegable	Si	Combinación de letras. Permite el Vacío.
9	Tipo de actividad	Lista desplegable	Si	Combinación de letras. Permite el Vacío.
10	Dirigentes	Campo de texto	Si	Combinación de caracteres. Permite el Vacío.
11	Descripción	Campo de texto	Si	Combinación de letras. Permite el Vacío.
12	Involucrados	Campo de selección múltiple	Si	Permite la selección múltiple de los elementos que muestra el sistema. No permite la entrada de datos. Permite el Vacío.
13	Participantes	Campo de texto	Si	Combinación de caracteres. Permite el Vacío.
14	Principal	Campo de selección	Si	Campo de selección para un atributo booleano. Permite el Vacío.

Resultados de las pruebas realizadas:

Luego de aplicados los métodos de prueba a los componentes implementados, es válido señalar que los resultados obtenidos hasta el momento han sido satisfactorios. Cada una de las no conformidades detectadas fue debidamente atendida logrando un correcto comportamiento de los componentes ante diferentes situaciones (entradas válidas y no válidas). A continuación se presenta el número de no conformidades detectadas tras cada iteración de pruebas realizadas:

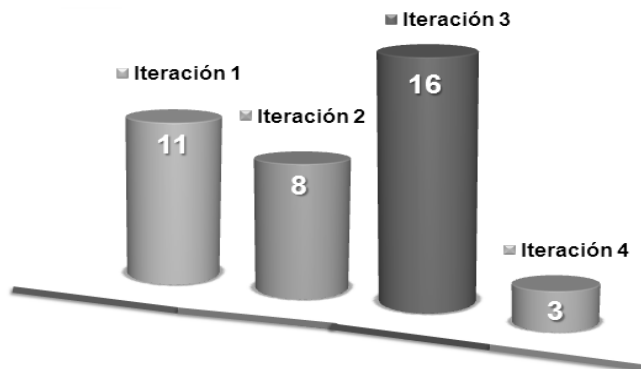


Figura 28: No conformidades por iteración detectadas durante las pruebas funcionales.

4.4 Conclusiones del capítulo.

El presente capítulo fue crucial para definir algunos elementos que intervienen en el desarrollo de la solución propuesta. En ese sentido se presentó la estructura física de componente Calendario, se definieron los estándares de código para su implementación y los componentes necesarios para su despliegue. Se aplicaron además las métricas: Tamaño Operacional de la Clase y Relaciones entre Clases para evaluar la calidad el diseño propuesto. Los resultados arrojados por dichas métricas permitieron valorar principalmente la complejidad en la implementación de la solución que se presenta con la actual investigación.

También se efectuaron pruebas de caja negra a los componentes Gestor de actividades y Calendario para verificar su correcto funcionamiento, aplicándose pruebas de partición de equivalencia. Tras 4 iteraciones de realización de pruebas funcionales se detectaron un total de 38 no conformidades. Estas fueron resultas permitiendo cumplir con los requerimientos capturados en la primera etapa de desarrollo.

Las pruebas de aceptación realizadas hasta el momento por los usuarios con el apoyo del equipo del proyecto han permitido verificar y validar los requerimientos identificados. Dichas pruebas arrojan que la solución propuesta, en opinión de los usuarios, proporciona un avance en la ejecución de proceso de Registro, Control y Seguimiento a las actividades en las entidades de prueba; posibilitando la gestión de las actividades en concordancia con lo que define el modelo actual de planificación.

CONCLUSIONES GENERALES

El desarrollo de la presente investigación y, en consecuencia, la obtención de los resultados generados por la misma, han permitido arribar a las siguientes conclusiones:

1. Con el análisis del curso actual del proceso de Registro, Control y Seguimiento a las actividades como subproceso de la Planificación por Objetivos se pautaron los aspectos significativos que impiden su desarrollo acertado en las entidades nacionales. Esto permitió reconocer la no existencia de una solución informática que responda cabalmente a lo que en realidad requieren las entidades cubanas.
2. La Modelación y Construcción de la solución propuesta viabiliza el registro, control y seguimiento a las actividades en las entidades cubanas, posibilitando la gestión dinámica de las mismas como elemento de planificación a partir de una jerarquía de roles. Esto resuelve el problema de asignación simultánea y dirigida a los responsables y participantes de las actividades.
3. Permite a diferencia de las alternativas adoptadas hasta el momento, establecer relaciones múltiples entre las actividades con los planes definidos por la entidad, los objetivos, los factores que influyen en el plan y las áreas de resultados clave, como define el modelo actualizado de Planificación por Objetivos.
4. Proporciona un soporte confiable para el almacenamiento centralizado de la información que se genera durante la ejecución del proceso de Registro, Control y Seguimiento a las actividades, resolviendo de esta manera el problema del registro inestable de la documentación referente al proceso.
5. A partir de la implementación de los requerimientos capturados para la primera etapa de desarrollo se obtuvo una solución integrada, probada y estable que satisface los acuerdos establecidos en el contrato.
6. Se destaca con la solución propuesta, la incorporación de principios por los que se mide la factibilidad del diseño de una aplicación informática. La utilización de patrones y métricas es un ejemplo fehaciente de ello.
7. Por consiguiente, el desarrollo de los componentes Gestor de actividades y Calendario como solución que informatizar el proceso de Registro, Control y Seguimiento a las actividades en el Subsistema PDO del Sistema Integral de Gestión de Entidades CedruX, constituye una mejora en la realización del proceso de Planificación por Objetivos en las entidades del país.

RECOMENDACIONES

Se recomienda considerando cumplidos todos los objetivos trazados en el presente trabajo de diploma lo siguiente:

1. Extender las funcionalidades relacionadas con los mecanismos de control, la exportación e importación de las actividades, en aras de ampliar las posibilidades de trabajo con la solución desarrollada.
2. Realizar integración con el subsistema Planificación Material y Financiera para garantizar la relación entre los recursos financieros y las actividades.
3. Proporcionar una ayuda que claramente indique el curso de las operaciones durante el trabajo con la solución propuesta.

BIBLIOGRAFÍA REFERENCIADA

1. **Coello González, Sayda y Hernández León, Rolando Alfredo. 2002.** El paradigma cuantitativo de la investigación científica. Habana : EDUNIV, 2002. ISBN: 959-16-0343-6.
2. **Bernal Vidal, Néstor y Galán Ramírez, Yuliet. 2008.** Proceso de Planificación por Objetivos en las entidades de las FAR. Habana : UCI, 2008.
3. **Filgueiras Sainz de Rozas, Miriam L. 2010.** Planeación Estratégica y Dirección por Objetivos. Habana : s.n., 2010.
4. **2009.** OpenBravo. [En línea] 2009. [Citado el: 10 de 09 de 2010.] <http://www.openbravo.com/es>.
5. **2010.** OpenERP. [En línea] 2010. [Citado el: 10 de 09 de 2010.] <http://www.openerpsite.com/erp-openerp-modulos>.
6. **2011.** Portal de ayuda del SAP. [En línea] 2011. [Citado el: 5 de 06 de 2011.] <http://www.sap.com/spain/solutions/business-suite/erp/keycapabilities/index.epx>
7. **Cabrera González, Miguel P., Obregón Rodríguez, Guillermo, Cárdenas Negrin, Margarita y Carralero Silva, Luis Mario.** XV FORUM DE CIENCIA Y TECNICA. SISTEMA ECONOMICO INTEGRADO.
8. **Machado Scull, Sandy y Galán Ramírez, Yuliet. 2009.** Glosario de términos del Subsistema Dirección por objetivos. Habana : UCI, 2009.
9. **Drucker, Peter. 1954.** The Practice of Management. 1954.
10. **Mintzberg, Henry y Waters, James. 1993.** De las estrategias deliberadas y emergentes. Diario de Gestión Estratégica. 1993.
11. **Hernández Ponce, Lidiarys y Montes de Oca Montero, Arnoldo. 2009.** 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 de Dirección por Objetivo. Habana : UCI, 2009.
12. **Ferré, Grau Xavier. 2011.** "Tutorial UML, Desarrollo Orientado a Objetos con UML." *Clikear.com*, 2011. [Citado el: 14 de 11 de 2010.] <http://www.clikear.com/manuales/uml/index.aspx>.
13. **Milestone Consulting. 2011.** Milestone Consulting. "Curso Modelado de Negocio con BPMN 2.0 y UML, Curso de BPMN, Capacitación-Entrenamiento en BPMN 2.0." [Citado el: 14 de 11 de 2010.] <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>.
14. **2003.** Extensible Markup Language (XML). [En línea] 2003. [Citado el: 27 de 01 de 2011.] <http://www.w3.org/XML/>.
15. **2002.** Lenguaje HTML. [En línea] 2002. [Citado el: 27 de 01 de 2011.] <http://www.desarrolloweb.com/articulos/711.php>.
16. **Pérez, Iván Nieto. 2008.** Diccionario de siglas CSS (Cascading Style Sheets). [En línea] 2008. [Citado el: 27 de enero de 2011.] <http://www.elcodigo.net/tutoriales/diccionario.html>.
17. **2008.** CodeBox. *Glosario php*. [En línea] 2008. [Citado el: 27 de 01 de 2010.] <http://www.codebox.es/glosario>.
18. **2008.** CodeBox. *Glosario framework*. [En línea] 2008. [Citado el: 27 de 01 de 2011.] <http://www.codebox.es/glosario>.
19. **Gómez Baryolo, Oiner, Morejón Borbón, Yoandry y García Tejo, Darien.** ARQUITECTURA TECNOLÓGICA PARA EL DESARROLLO DE SOFTWARE. Habana : UCI.
20. **'Cutter' Blades, Steve, Ramsay, Colin y Frederick, Shea. 2008.** Learning Ext JS. s.l. : Packt Publishing Ltd, 2008. ISBN 978-1-847195-14-2.
21. **2010.** ZEND FRAMEWORK. [En línea] 2010. [Citado el: 27 de 01 de 2011.] <http://framework.zend.com/manual/en/>.
22. **2008.** Doctrine. [En línea] 2008. [Citado el: 28 de 01 de 2011.] <http://www.doctrine-project.org>.
23. **Pérez, Javier Eguíluz. 2011.** "Introducción a AJAX." *LibrosWeb.es*, 2011. <http://www.librosweb.es/ajax/capitulo1.html>.

24. **2010.** Ingeniería De Software I. *Capítulo I HERRAMIENTAS CASE*. [En línea] 14 de abril de 2010. [Citado el: 27 de 11 de 2010.] <http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>.
25. **2005.** Visual-paradigm. [En línea] 2005. [Citado el: 28 de 09 de 2010.] <http://www.visual-paradigm.com>.
26. **2010.** Subversion. [En línea] 2010. [Citado el: 28 de 10 de 2010.] <http://subversion.apache.org>.
27. **2009.** Documentación del Servidor HTTP Apache 2.0. [En línea] 2009. [Citado el: 28 de 10 de 2010.] <http://httpd.apache.org/docs/2.0/es/>.
28. **2010.** PostgreSQL. *The world's most advanced open source database PostgreSQL*. [En línea] 2010. [Citado el: 28 de 01 de 2011.] <http://www.postgresql.org>.
29. **2010.** DISEÑO DE BASES DE DATOS RELACIONALES. [En línea] 2010. [Citado el: 28 de 01 de 2011.] <http://usuarios.multimania.es/cursosgbd/UD4.htm>.
30. **Gómez, Santiago Pavón. 2011.** Java Swing. [En línea] 2011.[Citado el: 27 de 01 de 2011.] <http://jungla.dit.upm.es/~santiago/docencia/apuntes/Swing/index.htm>.
31. **2010.** Los datos de objetos de PHP. [En línea] 2010. [Citado el: 28 de 1 de 2010.] <http://php.net/manual/es/intro.pdo.php>
32. **2011.** Introducing JSON. [En línea].2011. [Citado el: 28 de enero de 2011.] <http://www.json.org/>.
33. **2007.** Metodologías de sistemas. [En línea] 2007. [Citado el: 28 de 01 de 2011.] <http://metodologiasdesistemas.blogspot.com/2007/10/que-es-un-orm-object-relational-mapping.html>.
34. **UCID. 2009.** Proceso de desarrollo y gestión de proyectos de software. Habana : UCID, 2009.
35. **Díez, A. 2001.** IRqA y el desarrollo de proyectos: Experiencias Prácticas. I Jornadas de Ingeniería de Requisitos JIRA 2001. Sevilla, España : s.n., 2001.
36. **Sommerville, Ian. 2005.** Ingeniería de Software. Séptima edición. Madrid : Pearson Educación S.A., 2005. 84-7829-074-5.
37. **González, Dianelys Brito. 2011.** ARQUITECTURA DE SOFTWARE. Habana : CEIGE, 2011. CEIGE-N-00-i001.
38. **Chaviano Gómez, Enrique y Carrascoso Puebla, Yoan Arlet. 2009.** Propuesta de arquitectura orientada a servicios para el módulo de inventario del ERP cubano. Habana : UCI, 2009.
39. **Salazar, Ing. Tte. Leonel. 2008.** Curso de Capacitación por Roles: Generalidades de los Patrones de Diseño. Habana, Cuba : s.n., 2008.
40. **2010.** El mundo informático. [En línea] [Citado el: 8 de enero de 2010.] <http://jorgesaavedra.wordpress.com/2006/08/17/patrones-grasp-craig-larman/>.
41. **Prieto, Félix. 2009.** Patrones de diseño.
42. **2008.** Definicion.de. [En línea] 2008. <http://definicion.de/modelo-de-datos/>.
43. **Férrandez González, Mairelys y Zorrilla Rivera, Osley. 2010.** Diseño e implementación del componente Ajuste al Costo del Subsistema Costos y Procesos del Sistema Integral de Gestión de Entidades CEDRUX. Habana : UCI.
44. **Pressman, Roger. 1998.** Ingeniería de Software. Un enfoque práctico. 1998.
45. **Ramírez, Jaime.** Unidad de Programación. Métodos de Prueba del Software.
46. **Núñez Parada, Andrés M. 2011.** El prisma. Administración de empresas. [Online] 2011. http://www.elprisma.com/apuntes/administracion_de_empresas/planificacion/default5.asp.
47. **2010.** PERU-ERP.com. ¿Qué son los sistemas ERP? [En línea] 2010. <http://www.peru-erp.com/erp.html>.
48. **Porteiro, MSc. Marisel Sosa y Cobo Morales, Lic. Pedro H. 2011.** REVISTA BETSIME-LA REVISTA DEL EMPRESARIO CUBANO. El VERSAT-Sarasola: Sistema cubano de Gestión Contable-Financiero. [En línea] http://www.betsime.disaic.cu/secciones/eco_enemar_07.htm.

BIBLIOGRAFÍA CONSULTADA

1. **Bernal Vidal, Nestor, Galán Ramírez y Pereira, Misary (21 de 04 de 2010).** Modelo de Planificación por Objetivos. (A. R. Artola, Entrevistador)
2. **Bernal, Nestor (2008).** Proceso de Planificación por Objetivos en las entidades de las FAR. La Habana.
3. **Ponce, D. (20 de 04 de 2010).** Componentes de gestión de actividades. (M. A. Castellanos, Entrevistador)
4. **Cornejo González, José Enrique.** DocIRS. [En línea] [Citado el: 6 de 10 de 2010.] <http://www.docirs.cl/uml.htm>.
5. **Definición.org.** [En línea] [Citado el: 6 de Octubre de 2010.] <http://www.definicion.org>.
6. **Alvarez, Miguel Angel.** Desarrollo Web. [En línea] [Citado el: 6 de 10 de 2010.] <http://www.desarrolloweb.com/articulos/que-es-html.html>.
7. **Asesores, Imdea. Imdea.** [En línea] [Citado el: 5 de 10 de 2010.] http://www.imdea.com/www/index.php?option=com_content&view=article&id=83%3Aventajas-de-un-bpm&catid=57%3Afaqs-gestion-documental&Itemid=84&lang=es.
8. **Corzo, Giancarlo.** Desarrollo en Web. [En línea] 22 de 10 de 2008. [Citado el: 5 de 09 de 2010.] <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.
9. **Espinoza, Humberto.** PostgreSQL Una Alternativa de DBMS Open Source. 2005.
10. Parakey, Inc. Brothersoft. [En línea] 6 de 08 de 2008. [Citado el: 5 de 10 de 2010.] <http://es.brothersoft.com/FireBug-131343.html>.
11. **Superuser. 2003.** Tienda Linux. [En línea] 31 de 05 de 2003. [Citado el: 5 de 10 de 2010.] http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html.
12. **Lage Dávila, Carlos. 2007.** Discurso en la clausura del Seminario Nacional con los directores de empresas que aplican el perfeccionamiento empresarial. 29 de 08 de 2007.
13. **Caldera Mejía, Rodolfo. Diciembre 2004.** Planeación Estratégica de Recursos Humanos. NICARAGUA : EDUCACIÓN CONTINUA. UNIVERSIDAD POLITÉCNICA DE NICARAGUA., Diciembre 2004.
14. **Núñez Parada, Andrés M. 2011.** El prisma. Administración de empresas. [En línea] 2011. http://www.elprisma.com/apuntes/administracion_de_empresas/planificacion/default5.asp.
15. **Gómez Baryolo, Oiner, Morejón Borbón, Yoandry y García Tejo, Darien.** ARQUITECTURA TECNOLÓGICA PARA EL DESARROLLO DE SOFTWARE. Habana : UCI.
16. **Pressman, Roger. 1998.** Ingeniería de Software. Un enfoque práctico. 1998.
17. **González, Dianelys Brito. 2011.** ARQUITECTURA DE SOFTWARE. Habana : CEIGE, 2011. CEIGE-N-00-i001.
18. **Chaviano Gómez, Enrique y Carrascoso Puebla, Yoan Arlet. 2009.** Propuesta de arquitectura orientada a servicios para el módulo de inventario del ERP cubano. Habana : UCI, 2009.
19. **Fernández González, Mairelys y Zorrilla Rivera, Osley. 2010.** Diseño e implementación del componente Ajuste al Costo del Subsistema Costos y Procesos del Sistema Integral de Gestión de Entidades CEDRUX. Habana : UCI.
20. **UCID. 2009.** Proceso de desarrollo y gestión de proyectos de software. Habana : UCID, 2009.
21. **Bernal Vidal, Néstor y Galán Ramírez, Yuliet. 2008.** Proceso de Planificación por Objetivos en las entidades de las FAR. Habana : UCI, 2008.
22. **Filgueiras Sainz de Rozas, Miriam L. 2010.** Planeación Estratégica y Dirección por Objetivos. Habana : s.n., 2010.

23. **Fernández, Osmar Leyet. 2010.** Documento Línea Base de proyecto-CEDRUX-1.0. 2010.
24. **González Doria, Heidi. 2010.** Métricas en el desarrollo del Software. Capitulo 4. México : Universidad de las Américas Puebla.. ISBN/ISSN.
25. **Visconti, Marcelo y Astudillo, Hernán. 2010.** Fundamentos de Ingeniería de Software. Valparaiso.España. : Departamento de Informática. Universidad Técnica Federico Santa María.
26. **2010.** Los datos de objetos de PHP. [En línea] 2010. [Citado el: 28 de 1 de 2010.] <http://php.net/manual/es/intro.pdo.php>
27. **Pacios Lozano, Ana R. 2007.**Administración por Objetivos.
28. **Centro de Informatización y Gestión de Entidades. 2010.**CedruX.Desarrollo Tecnológico. Habana. : UCI.
29. **Borroto Cruz, Dr. Eugenio Radamés y Aneiros R, Ramón. 2003.** La Dirección por Objetivos. La Habana : Escuela Nacional de Salud Pública.
30. **Hamilton Reyna, Yaremis. 2010.** Plan de producto.Planificación por Objetivos v1.0. Habana. : Centro de soluciones de gestión.Departamento de productos.
31. **Borroto Cruz, Eugenio Radamés, Castell-Florit serrate, Pastor y Díaz Rojas, Pedro A. 2010.** Planeación Estratégica y Dirección por Objetivos. Habana : Escuela Nacional de Salud Pública.
32. **Rodríguez Pomedá, Jesús. 1996.** PLANIFICACIÓN Y CONTROL DE LA EMPRESA: LOS SISTEMAS DE APOYO A LA DIRECCIÓN. Madrid : IADE-UAM.
33. **IEEE-SA. 2010.** "IEEE SA - 1471-2000 - IEEE Recommended Practice for Architectural Description for Software-Intensive Systems". [En línea]. <http://standards.ieee.org/findstds/standard/1471-2000.html>.
34. **Microsoft. 2011.** "Capítulo 3: Patrones de arquitectura y estilos." [En línea]. *MSDN Magazine*. <http://msdn.microsoft.com/en-us/library/ee658117.aspx>.
35. **Milestone Consulting. 2011.** "Curso Modelado de Negocio con BPMN 2.0 y UML, Curso de BPMN, Capacitación-Entrenamiento en BPMN 2.0." [En línea]. Milestone Consulting. <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>.
36. **Xavier Ferré Grau. 2011.** "Tutorial UML, Desarrollo Orientado a Objetos con UML." *Clikear.com*. <http://www.clikear.com/manuales/uml/index.aspx>.
37. **UCID. 2008.**Vista de datos de la Arquitectura de ERP. Habana : UCID.
38. **Thinking People. 2010.** "DPO Gestión Retribución Variable por Objetivos ", <http://www.thinkingpeoplerecursoshumanos.es/software-de-rr-hh/dpo-gestion-retribucion-variable-por-objetivos/>.
39. **2010. Dirección EICMA.** Versat Sarasola, Un efectivo sistema cubano de contabilidad. [En línea] noviembre de 2010. <http://www.eimagr.cu/index.php/component/content/article/60-portada/101-versat-sarasola-un-sistema-para-trabajar>.