

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 3



*Trabajo de Diploma para Optar por el Título de
Ingeniero en Ciencias Informáticas*

*Título: Módulo Configuración del Sistema para la
Planificación de Actividades (SIPAC).*

*Autor(s): Lucrecia de la Caridad Valiente Orbera.
Juana Dailiana León Lastre.*

Tutor: Ing. Néstor Bernal Vidal.

Ciudad de La Habana, noviembre de 2012

"Año del 54 Aniversario de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ____ días del mes de _____ del año_____.

Autor

Tutor

Autor

Tutor



Resumen

La planificación de actividades en las entidades cubanas se elabora siguiendo el modelo de Planificación por Objetivos que surge de la unión de la Dirección por Objetivos y la Planeación Estratégica. Un proceso vital de este modelo es el proceso de Organización, que sienta las bases para los procesos de Planificación, Dirección y Control. Sistema para la Planificación de Actividades es la herramienta informática destinada a informatizar este proceso que actualmente se encuentra en prueba experimental en el Ministerio de las Fuerzas Armadas y en algunos Organismos de Administración Central del Estado. Esta herramienta implementa parte del proceso de Organización, pero no tiene en cuenta la asignación de responsabilidades para una correcta comunicación entre los niveles de la planificación sin violar lo establecido. Es por ello que tampoco esta herramienta proporciona el aseguramiento de un lenguaje común entre dichos niveles a la hora de clasificar los planes, objetivos y actividades, permitiendo operaciones no atribuidas que influyen en la integridad de la información, la pérdida de recursos y tiempo en la reelaboración de la misma, durante el proceso de concepción y ejecución de la planificación. El presente trabajo de diploma está dirigido a la obtención del módulo Configuración para el Sistema para la Planificación de Actividades, de manera que el proceso de planificación vigente en las entidades cubanas se ejecute sobre las bases de una correcta organización, compartimentación y entendimiento común de la información entre y hacia los distintos niveles de la planificación.

Palabras clave:

Compartimentación, Nomencladores, Organización, Permisos, Planificación, Planificación por Objetivo.



Índice de Contenidos

Introducción.....	1
Capítulo 1. Fundamentación Teórica.....	6
Introducción del Capítulo.	6
1.1. Estado del Arte.	6
1.1.1. Planificación.....	6
1.1.2. Dirección por Objetivos, Planeación Estratégica y Planificación por Objetivos.....	7
1.2. Proceso de planificación en las entidades cubanas.	8
1.3. El proceso organizativo dentro de la planificación en las entidades cubanas.	9
1.3.1. Medios y herramientas para planificar actividades y objetivos.	11
1.4. Lenguajes propuestos para el desarrollo de la solución.	15
1.4.1. Lenguajes de Modelado.	15
1.4.2. Estándar de Modelado.	15
1.4.3. Lenguaje de Marcado de Hipertexto.	16
1.4.4. Lenguajes de Desarrollo del lado del cliente.	16
1.4.5. Lenguajes de Desarrollo del lado del servidor.	17
1.4.6. Frameworks o Marcos de Trabajo.	17
1.5. Tecnologías y Herramientas de Desarrollo.	18
1.6. Estándares de Código.....	21
1.7. Modelo de Desarrollo.	22
1.8. Conclusiones del Capítulo.	23
Capítulo 2. Propuesta de Solución.	25
Introducción del Capítulo.	25
2.1. Propuesta de Solución.	25
2.2.1. Diagramas de procesos de negocio (BPMN).	26
2.2.2. Especificación de los procesos de negocio.....	28
2.3. Conceptos manejados en el negocio.....	32
2.3.1. Modelo Conceptual.	32
2.4. Requisitos.....	32
2.4.1. Técnica de Captura de Requisitos.....	33

2.4.2. Listado de Requisitos Funcionales Identificados.....	33
2.4.3. Especificación de Requisitos Funcionales.....	34
2.4.4. Prototipos de Interfaz de Usuarios.....	36
2.4.5. Validación de los requisitos funcionales.	37
2.4.6. Requisitos no funcionales.	37
2.5. Arquitectura de la solución propuesta.	38
2.5.1. Principales aspectos arquitectónicos.....	39
2.6. Patrones de diseños.	41
2.6.1. Patrones GRASP.....	41
2.6.2. Patrones GoF.	43
2.7. Diseño de la solución.	44
2.7.1. Diseño de componentes.....	44
2.7.2. Servicios consumidos y brindados por el módulo Configuración.....	45
2.7.3. Diseño de clases.....	45
2.7.4. Diagramas de clases del diseño.	46
2.7.5. Modelos de datos.	48
2.1.1. Otros modelos aplicados.....	50
2.8. Estructura Física del Sistema.....	52
2.9. Implementación.....	55
2.10. Diagrama de Despliegue.....	55
2.11. Conclusiones del capítulo.	56
Capítulo 3. Validación de la Solución Propuesta.	57
Introducción del Capítulo.	57
3.1. Validación del Diseño Propuesto.....	57
3.1.1. Métrica TOC: Tamaño operacional de clase.....	58
3.1.2. Análisis de los resultados obtenidos en la evaluación de la métrica TOC.	59
3.1.3. Métrica RC: Relaciones entre clases.....	59
3.1.4. Análisis de los resultados obtenidos en la evaluación de la métrica RC.	61
3.2. Pruebas de aplicación.....	61
3.2.1. Niveles de Prueba:	62
3.2.2. Métodos de Prueba:	62

Índice de Contenidos

3.2.3. Pruebas realizadas a los componentes:.....	63
3.2.4. Casos de pruebas de caja negra generados al aplicar dicha técnica.....	64
3.3. Conclusiones del Capitulo	69
Recomendaciones	71
Referencia Bibliográfica	72

Índice de Figuras

Figura 1.1: Principales procesos de la planificación de actividades guiadas al cumplimiento de objetivos.	9
Figura 1.2: Procesos del Modelo de Desarrollo a utilizar.	23
Figura 1.3: Fases o etapas del ciclo de vida de un proyecto.	23
Figura 2.1: Definición de la Organización.	27
Figura 2.2: Generalización.	27
Figura 2.3: Establecimiento de las Responsabilidades.	28
Figura 2.4: Modelo Conceptual correspondiente al módulo Configuración.	32
Figura 3.1: Representación en % de la cantidad de clase y el número procedimientos que contienen.	58
Figura 3.2: Representación del valor en % del atributo Responsabilidad.	59
Figura 3.3: Representación del valor en % del atributo Complejidad de Implementación.	59
Figura 3.4: Representación del valor en % del atributo Reutilización.	59
Figura 3.5: Representación en % de las asociaciones de uso por cantidad a de clases.	60
Figura 3.6: Representación del valor en % del atributo Acoplamiento.	60
Figura 3.7: Representación del valor en % del atributo Complejidad de Mantenimiento.	60
Figura 3.8: Representación del valor en % del atributo Cantidad de Pruebas.	61
Figura 3.9: Representación del valor en % del atributo Reutilización.	61
Figura 3.10: No conformidades por iteración detectadas durante las pruebas funcionales.	66
Figura 3.11: Estimación de la compartimentación y sus dimensiones.	68
Figura 3.12: Estimación de la variable Entendimiento Común.	68
Figura 4.1: Prototipo de interfaz de usuario del requisito Modificar grupo de usuarios por rol.	¡Error! Marcador no definido.
Figura 4.2: Prototipo de interfaz de usuario del requisito Buscar Grupos de usuarios por rol.	¡Error! Marcador no definido.

Índice de Tablas

Tabla 1: Descripción del proceso de negocio correspondiente a la Definición de la Organización.	28
Tabla 2: Descripción del proceso de negocio correspondiente a la Generalización.	29
Tabla 3: Descripción del proceso de negocio correspondiente al Establecimiento de Responsabilidades.	30
Tabla 4: Listado de Requisitos Funcionales Identificados.	33
Tabla 5: Adicionar grupo de usuarios por rol.	35
Tabla 6: Servicios consumidos y brindados por el módulo Configuración.	45
Tabla 7: Descripción del diseño de clases asociadas al componente Nomencladores.	48
Tabla 8: Relación de columnas indexadas por tablas.	50
Tabla 9: Caso de Prueba Adicionar Permiso.	64
Tabla 10: Descripción de las variables.	65
Tabla 11: Variables con sus dimensiones.	66
Tabla 12: Modificar grupo de usuarios por rol.	¡Error! Marcador no definido.
Tabla 13: Eliminar grupo de usuarios por rol.	¡Error! Marcador no definido.
Tabla 14: Buscar grupo de usuarios por rol.	¡Error! Marcador no definido.
Tabla 15: Listar grupo de usuarios por rol.	¡Error! Marcador no definido.
Tabla 16: Rango de valores de para la evaluación técnica de los atributos de calidad relacionados con la métrica TOC.	¡Error! Marcador no definido.
Tabla 17: Rango de valores de para la evaluación técnica de los atributos de calidad relacionados con la métrica RC.	¡Error! Marcador no definido.

Introducción

La evolución es un proceso que tiene lugar todo el tiempo, es sinónimo de adaptarse al medio según el término al que se aplica del éxito y la supervivencia. Precisamente la evolución ha estado presente también en la organización, planificación, dirección y control ejecutados en las entidades cubanas. Las primeras técnicas adoptadas provenientes de la antigua Unión de Repúblicas Socialistas Soviéticas (URSS) fueron adaptadas al entorno socio-económico de nuestro país, estas se introdujeron en la década de los '90 y estaban estrechamente relacionados con la Planeación Estratégica y la Dirección por Objetivos.

Unos años después, tomando como base las experiencias obtenidas durante el perfeccionamiento empresarial aplicado principalmente a las entidades de las Fuerzas Armadas Revolucionarias (FAR) y a algunas entidades civiles, surgen las estrategias para propiciar una mejora y adecuar las bases teóricas, metodológicas y organizativas a la realidad de ese entonces del país (2). Con el auge que empezó a tomar la introducción de la informática y las tecnologías y su uso en la informatización de las entidades cubanas, se empiezan a dar los primeros pasos en la elaboración de sistemas para informatizar los procesos comprendidos en la Planificación por Objetivos. Se toma como guía el desarrollo de dicho proceso en entidades pioneras en la planificación, mayormente entidades pertenecientes a las FAR. El primer resultado fue la obtención del PTRAB¹ colocando a estas entidades a la vanguardia en ese momento, ya que este sistema permitía elaborar y ejecutar los planes anuales correspondientes a un periodo de 2 años con sus objetivos y actividades. Toda la información que se introducía en el sistema se podía traspasar o exportar mediante ficheros con extensión dbf, ya que no contaba con la prestación de una base de datos centralizada a la cual se pudiese acceder, leer y compartir información mediante conexiones en red. Aún así permitía compartir e intercambiar la información entre los niveles de la planificación a pesar de que la información podía ser ampliamente modificable por parte de los receptores de la misma, violando la integridad de la información, y limitando el control de las acciones sobre la información existente por parte de los jefes. Estas desventajas junto a otras, como su arquitectura pobremente escalable, su poca percepción de la planificación en el espacio-tiempo deseado e implementado sobre un lenguaje de desarrollo privativo, no le permitieron evolucionar a la par del modelo para la planificación orientada al cumplimiento de objetivos, el cual ha expandido su visión y nivel de exigencias en

¹ **PETRAB:** Diminutivo de Plan de Trabajo, sistema informático desarrollado en FoxPro para la planificación por objetivos implantado en 1997 en el MINFAR y entidades pertenecientes a las FAR, y desarrollado por el Tte. Coronel Javier Cocurill Villamonte.

cuanto a organización, planificación, dirección y control. Este modelo de planificación se encuentra establecido la Instrucción No.1² (**Ver Anexo1**) para la planificación de los objetivos y actividades en los órganos, organismos de la administración central del estado, entidades nacionales y las administraciones locales del poder popular y por la cual también se rige el plan de actividades del Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR). Actualmente, a pesar de que PETRAB aún sigue siendo utilizada, es una herramienta destinada y marcada a ser reemplazada.

Por otra parte, las entidades pertenecientes a los Organismos de Administración Central del Estado (OACE) optaron por la utilización de aquellas herramientas que más conocían, las que propone Microsoft en su paquete Office, el Outlook y el Excel, así como pequeñas aplicaciones en Access. Con ellas aunque de manera poco convencional, se lograban en cierta medida buenos resultados. La utilización de estas herramientas en los niveles más altos de la planificación, más que una solución, a veces se tornaba en un problema mayor, porque se tenía que invertir más tiempo del reglamentado en la corrección de la información, pues muchos elementos planificados no se correspondían con su verdadero propósito o simplemente se violaban.

Es así como entonces surge la necesidad de contar con una herramienta que informatice estos procesos al nivel requerido, y nace de la unión y cooperación entre la Unidad de Compatibilización, Integración y Desarrollo de software para las FAR (UCID) y Centro de Informatización de Gestión de Entidades (CEIGE) el proyecto Sistema para la Planificación de Actividades (SPA), proyecto encargado de proporcionar una herramienta que facilite el proceso de planificación de las entidades del país, y que sea capaz de adaptarse a cualquier método de planificación con bases a actividades, que sea escalable, y por sobre todas las cosas, que permita llevar el registro, seguimiento y control de los elementos planificados. A finales del 2010 SPA libera una primera versión del producto conocido como Sistema para la Planificación de Actividades (SIPAC), con el cual se procede a efectuar pruebas en varios ministerios del país, dígase Ministerio de Educación (MINED) y MINFAR con el objetivo de explorar el sistema en un ambiente lo más real posible como base para una futura generalización en el país.

Esta primera versión de SIPAC cuenta con módulos encargados de informatizar de forma general, el mecanismo que establece la modelación de los niveles de planificación mediante la

² **No1** : Documento archivado en el Protocolo de Disposiciones Jurídicas que obra en la Secretaria del Consejo de Ministros, y expedido en el Palacio de la Revolución.

representación de estructuras con su composición, así como los dominios en los cuales se encuentran estas estructuras, y los niveles de acceso a la gestión de la información. Aún así, esto solo representa el primer momento dentro del proceso de la definición de los conceptos estructurales para el establecimiento de la organización necesaria para planificar las actividades y objetivos de la entidad o nivel de planificación. Lo mismo sucede con el proceso de asignación de responsabilidades y restricción de acciones o eventos en la gestión de la información, pues estas funcionalidades solo limitan el acceso a módulos donde se gestiona determinado tipo de información, pero no permite restringir el acceso de lectura o escritura en dependencia del nivel en que se está planificando y el origen o fuente de la planificación visualizada. No obstante, el módulo destinado a la planificación estratégica y operativa cuenta con ciertas funciones que permiten definir aspectos organizativos como algunos nomencladores. Pero este no presenta funcionalidades que limiten el uso incorrecto de términos o clasificaciones de los elementos de la planificación, la atribución indebida de establecer planificaciones no autorizadas donde se involucre a personal ajeno a la entidad o nivel de planificación³, tampoco limita el acceso a la información planificada dentro, y a veces hasta fuera de su propio nivel, violando lo establecido en la Instrucción No1.

De forma general estas no conformidades detectadas en la prueba arrojaron como resultado la ineficiencia de SIPAC en restringir el acceso a la información y la falta de conceptos necesarios en el ámbito organizacional propio de la planificación. Influyendo igualmente en el incompleto nivel de compartimentación de la planificación y la necesidad de generalizar las clasificaciones de los elementos de la planificación de forma jerárquica. Debido a que el proceso de retroalimentación de la planificación se hace muy complejo, no es entendido puesto que no existe un vocabulario común para clasificar las actividades y los planes, la información tenía que volver a reelaborarse mediante un proceso engorroso de confirmación e identificación de la misma entre dichos niveles.

Por lo anteriormente planteado surge el siguiente **problema a resolver**:

¿Cómo contribuir a que el proceso de planificación vigente en las entidades cubanas se ejecute sobre las bases de una correcta organización, compartimentación y entendimiento común de la información entre y hacia los distintos niveles deseados dentro del Sistema para la Planificación de Actividades?

³**Nivel de Planificación:** Puede ser una persona, cargo, área, departamento o entidad que responda por determinadas funciones dentro del ejercicio de la planificación.

El problema planteado anteriormente se enmarca en el **objeto de estudio**: proceso de planificación en las entidades cubanas; trazándose como **objetivo general**: desarrollar el módulo Configuración para el Sistema para Planificación de Actividades, de modo que se contribuya a que el proceso de planificación vigente en las entidades cubanas se ejecute sobre las bases de una correcta organización, compartimentación y entendimiento común de la información entre y hacia los distintos niveles deseados dentro del Sistema para la Planificación de Actividades.

Se desglosan como **objetivos específicos**:

1. Fundamentar la investigación, mediante la elaboración del Marco Teórico.
2. Realizar el análisis, diseño e implementación del módulo Configuración.
3. Validar el módulo mediante las pruebas.

La presente investigación está enmarcada en el **campo de acción**: El proceso organizativo dentro de la planificación en las entidades cubanas.

Manteniendo como base los elementos expuestos anteriormente, se define como **idea a defender**: El desarrollo del Módulo Configuración contribuirá a que el proceso de planificación vigente en las entidades cubanas se ejecute sobre las bases de una correcta organización, compartimentación y entendimiento común de la información entre y hacia los distintos niveles deseados dentro del Sistema para la Planificación de Actividades.

Esto permitirá como **posibles resultados** de la presente investigación:

1. Modelado de los procesos.
2. Modelo conceptual.
3. Especificación de requisitos.
4. Diagrama de componentes.
5. Modelo de datos.
6. Componente para la gestión de grupos de usuarios por rol.
7. Componente para el manejo y control de los permisos de los usuarios sobre la información planificada.
8. Componente para la gestión de los nomencladores horizontales y verticales.
9. Módulo Configuración para el Sistema de Planificación de Actividades.

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

Los **métodos de investigación** científica empleados en la investigación son:

Dentro del conjunto de **Métodos Teóricos**:

Modelación: Donde se lleva a cabo una modelación del proceso de actualización e intercambio de información, permitiendo predecir la respuesta de dicho proceso a variaciones de algunos de sus parámetros, sin tener que ejecutar el proceso en la realidad.

Analítico – sintético: Para el análisis, estudio y comprensión de la bibliografía usada, la formulación del problema y los objetivos.

Histórico- Lógico: Para la administración de los datos relacionados con la instauración del proceso de Planificación por Objetivos en el país.

Para obtener los datos se utilizó el **Método Empírico**:

La Observación: Para la recogida de la información de cada uno de los conceptos o variables definidas en la idea a defender con la intención de acumular la mayor cantidad de información posible.

Entrevista: Con el propósito de construir una concepción inicial de la problemática. Estas entrevistas fueron realizadas a profesionales del Sistema de Planificación de Actividades y a profesionales del Grupo Planificación del Comité Ejecutivo del Consejo de Ministros (CECM) y profesionales del Ministerio de las Fuerzas Armadas Revolucionarias.

Estructura Capitular

Capítulo 1: Fundamentación Teórica:

Se expone el estado del arte referente al proceso de planificación vigente en las entidades del país, sus principales características y su origen, así como el proceso de organización y establecimiento de prioridades y niveles de subordinación dentro del proceso de planificación. Se realiza la fundamentación teórica del tema. Se describen y valoran algunas de las soluciones existentes. Se señalan las características principales del modelo de desarrollo propuesto y se

justifican las herramientas y tecnologías para la modelación e implementación de la solución que se propone.

Capítulo 2: Propuesta de Solución:

En este capítulo se describen las características esenciales que debe tener la solución propuesta, sustentadas por la modelación de los procesos de negocio y la definición de los requisitos. Se plasman algunos de los artefactos propuestos en la fase de Modelación definida en la metodología de software aplicada, el diseño e implementación de los componentes y la arquitectura que soportan los módulos de software.

Capítulo 3: Validación de la solución propuesta:

Este capítulo incluye lo referente a la validación del diseño, pruebas de aplicación y pruebas en base a las variables que rigen el problema a resolver.

Capítulo 1. Fundamentación Teórica.

Introducción del Capítulo.

En este capítulo se exponen los principales procesos de la planificación, así como sus características más esenciales que se ponen de manifiesto durante la ejecución del mismo en las entidades nacionales. Se abordan las principales estrategias adoptadas por dichas entidades para una mejor organización, dirección y control de la planificación. Se expone una panorámica de las principales herramientas utilizadas hasta el momento como medio de automatización del proceso de planificación de actividades, sus bondades y desventajas, así como un resumen de las principales técnicas y tecnologías que se utilizan en el desarrollo y obtención de la solución al problema que encausa el presente trabajo de diploma.

1.1. Estado del Arte.

1.1.1. Planificación.

La planificación es una parte indispensable de la vida y de la sociedad que nos rodea. Se puede decir que funciona como un reflejo incondicional, desde el mismo momento en que una persona se levanta pensando en las cosas que tiene que hacer durante ese día y la mejor manera de lograrla, ubicándose mejor en el tiempo y espacio necesario para ello. Las ideas concebidas con el fin de lograr una meta, alcanzar un propósito, de forma consciente o no, se traducen sencillamente en “planificación”.

Uno de los exponentes más prominente en materia de planificación, y considerado como el padre de la planificación en los tiempos actuales, James A. Finch, plantea que “la planificación no es más que el proceso de establecer metas y definir medios para alcanzar las mismas” (3) . Por otra parte, el Máster en Administración y Dirección de Empresas Andrés M. Núñez Parada propone que la planificación “consiste en determinar las metas u objetivos a cumplir seleccionando las misiones, objetivos y acciones para alcanzarlos” (4). Partiendo de la esencia de las ideas anteriormente expuestas, este trabajo se acoge al concepto de que la planificación es una de las funciones básicas del Proceso de Gestión que permite establecer un nivel de estabilidad en las entidades y proporciona condiciones favorables ante los cambios avizorados o no en un futuro.

1.1.2. Dirección por Objetivos, Planeación Estratégica y Planificación por Objetivos.

Muchas concepciones de la planificación plasman “metas” y “objetivos” como el principal eslabón, razón por la cual surge un método de planificación un poco más profundo y mejor definido, se trata de la Dirección por Objetivos (DpO). Peter Drucker considerado padre de la DpO, alega que es un proceso mediante el cual la organización define objetivos comunes entre todos sus miembros, para los que se definen áreas de responsabilidad para poder medir sus resultados y utilizar esas medidas como vía para la operación de las acciones. (5)

Por otra parte, como resultado natural de la evolución del concepto de Planificación surge la Planeación Estratégica como sistema de gerencia en los ‘70 y el proceso de desarrollo de planes para alcanzar propósitos u objetivos; desplaza el énfasis del “qué lograr” (objetivos) y el “qué hacer” (estrategias), concentrándose solo en objetivos factibles de lograr y en qué negocio o área competir, según las oportunidades y amenazas que ofrece el entorno.

Teniendo en cuenta esta evolución mundial de la Planificación, Cuba adopta un modelo de planificación personalizado pero matizado por las bondades de la Dirección por Objetivos y la Planeación Estratégica, denominada Planificación por Objetivos. Esta propone una adecuada combinación de recursos, tiempo y personas como parte del mecanismo de la entidad u organización para obtener resultados relevantes. Esta definición es acogida por la presente investigación, agregándole también la definición de metas o fines estratégicos pero implementados operativamente mediante planes proyectados sobre Áreas de Resultados Claves (ARC) las cuáles enmarcan los objetivos a alcanzar y actividades a cumplir con su funcionamiento estatal⁴. Este modelo está guiado por 4 aspectos fundamentales que rigen la elaboración y ejecución con éxito de la planificación: la organización, la planificación, la dirección y el control.

- ✚ **Organización:** (¿Cómo se va a hacer y con quién?) Es el proceso para ordenar y distribuir el trabajo, asignar responsabilidades de tal manera que estos puedan alcanzar las metas de la organización. Comprende la estructura y la administración de recursos humanos.
- ✚ **Planificación:** (¿Qué es lo que se quiere hacer? ¿Qué es lo que se va a hacer?) Define objetivos organizacionales y propone medios para lograrlos. Comprende las metas, objetivos, estrategias y planes.

⁴ **Funcionamiento Estatal:** Actividades por la cual responde una persona u organización al estado.

- ✚ **Dirección:** (Verificar que se haga). Guiar, liderar la realización de las tareas necesarias para lograr los objetivos de la organización. Comprende la motivación, el liderazgo, la comunicación y el comportamiento individual y de grupo.
- ✚ **Control:** (¿Cómo se ha hecho?) Es aquel en el que una persona, un grupo o una organización vigilan el desempeño y emprende acciones correctivas. Comprende las normas, medidas, comparaciones y acciones correctivas.

Estos procesos mantienen una alta dependencia uno de los otros, ya que la correcta implementación de cada uno de ellos influye en el éxito de las acciones ejecutadas. La planificación debe de ser primeramente organizada, luego dirigida y finalmente controlada. De esta manera se obliga a ejecutar con objetividad cada uno de los procesos, formando una cadena de responsabilidades y dependencias.

1.2. Proceso de planificación en las entidades cubanas.

El proceso de planificación en las entidades nacionales está avalado legalmente por lo planteado en la Instrucción No.1 donde se plasma todo el curso que debe de llevar la misma, las responsabilidades de los involucrados, desde su inicio en la etapa de concepción hasta la ejecución, incluyendo el control y seguimiento. El núcleo del proceso de planificación consiste en llevar un correcto registro, control y seguimiento de los elementos planificados: planes, Áreas de Resultado Claves (ARC), objetivos, criterios de medidas, actividades y Factores que Influyen en el Plan (FIP). Existen tres procesos fundamentales: Concepción, Ejecución y Aprobación-Conciliación. En la Concepción del plan se elaboran todos aquellos elementos que se desean cumplir, se elaboran planes con las ARC que comprenden los objetivos con sus criterios de medidas y las actividades que guían el cumplimiento de dichos objetivos. Una vez elaborados estos elementos pasan por el proceso de Aprobación-Conciliación, donde se decide por determinados niveles de la planificación qué elementos son posibles de alcanzar o son aceptados teniendo en cuenta su envergadura. Aquellos elementos que son rechazados durante este proceso, vuelven a reelaborarse por los niveles correspondientes, o simplemente se abandonan en caso de que no sea verdaderamente factible. Por otra parte, aquellos que fueron aprobados pasan inmediatamente a formar parte de la planificación de las entidades involucradas en dichos elementos planificados. Una vez llegada la fecha de efectuar el cumplimiento del plan, ya iniciado el proceso de Ejecución, se procede a controlar por el correcto cumplimiento de las actividades y los objetivos del propio nivel de planificación y de aquellos niveles que se subordinan a él. En caso de que exista alguna desviación o una

incorrecta implementación, detectarla y tomar medidas correctivas. Todo este proceso se muestra de forma concisa en la figura que se presenta a continuación:

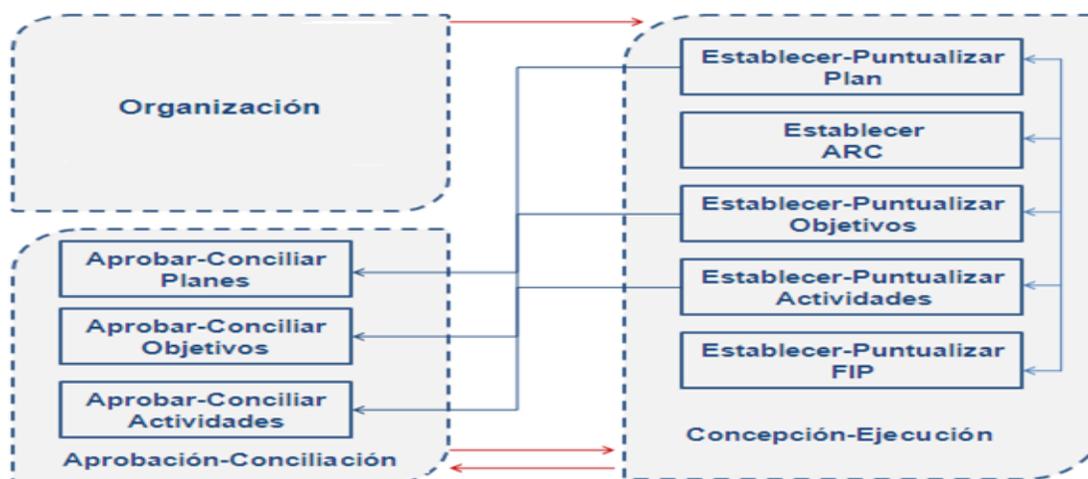


Figura 1.1: Principales procesos de la planificación de actividades guiadas al cumplimiento de objetivos.

Como se puede apreciar los procesos Concepción, Ejecución y Aprobación-Conciliación llevan implícitos los 3 últimos aspectos o procesos de la Planificación por Objetivos: Planificación, Dirección y Control. Una planificación, sin una buena organización, no puede ser bien dirigida y mucho menos controlada, y conlleva al desperdicio de esfuerzos y medios. Por esta razón, la concepción y ejecución de los elementos de la planificación en las entidades cubanas está precedida por un proceso de organización o pre-concepción.

1.3. El proceso organizativo dentro de la planificación en las entidades cubanas.

El proceso de Organización de la planificación posee una gran influencia sobre el resto de los procesos, por lo que saber implementarlo de la forma correcta requiere buena capacidad de mando y visión estratégica.

Está guiado por las siguientes actividades:

- ✚ Definición de la estructura y su composición.
- ✚ Establecimiento de los niveles de la planificación.
- ✚ Definición de las responsabilidades.
- ✚ Generalización de las indicaciones para realizar la planificación.

Capítulo 1. Fundamentación Teórica

En este proceso se define, inicialmente, cuáles son las estructuras funcionales necesarias para ejercer un correcto funcionamiento, con sus plantillas, áreas de funcionamiento, departamentos y los cargos disponibles que responderán al proceso de planificación. Esta definición parte de la estructura de la entidad emitida por un área especializada, casi siempre denominada Organización y Personal (OyP), donde está representada la plantilla económicamente (los cargos que realmente se pagan). En la mayoría de las veces, la estructura para planificar coincide en un 80% con la definida por OyP, aunque también se representan plantillas ficticias necesarias para poder efectuar una mejor distribución de las responsabilidades en la planificación. En esta actividad se genera una propuesta de estructura para la planificación según la visión estratégica del responsable en responder por el funcionamiento de la misma. Esta propuesta es sometida a valoración con el resto de los implicados para detectar si pueden surgir escenarios que atenten contra el correcto funcionamiento de la misma.

Conocida ya la estructura, se tiene una primera visión de los niveles de planificación existentes dentro de la entidad, donde cada uno de estos niveles puede ser un nivel superior e inferior al mismo tiempo, según el flujo que debe de tener la información planificada por cada uno de ellos y el rango de acción. El rango de acción define quienes son subordinados y jefes, para establecer las responsabilidades de cada uno y los permisos de interacción con aquellos planificadores fuera del rango de acción, estableciendo así una jerarquía de procesos desde el nivel superior hasta el nivel más bajo dentro de la entidad. Estas actividades son ejecutadas desde el macro nivel entidad hacia cada uno de los niveles dentro de la misma, y está guiada por el jefe que aprueba o por el que responde por la planificación en ese nivel mediante un documento emitido en ese mismo orden, que contiene las indicaciones generales para realizar la planificación. Dentro de estas indicaciones generales también se incluyen las clasificaciones generales a utilizar en la planificación de actividades, objetivos y otros elementos. Estas clasificaciones se conforman con un procedimiento parecido: se establece una primera proposición según la visión del nivel más alto de la planificación dentro de la entidad, se circula la propuesta para conciliarla y detectar si es necesario incluir otras utilizadas por determinados niveles y que pueden ser concebidas como generales. Una vez conciliada la propuesta general y las propuestas de cada uno de los niveles, se procede a elaborar las clasificaciones generales definitivas y se circulan mediante las indicaciones generales.

Este proceso organizativo, con cada una de sus actividades, permite establecer las bases para la planificación, seguimiento y control de los objetivos con sus actividades. Permite conocer al planificador sus responsabilidades, a quién le debe de planificar, qué debe o no planificar, y cuál

es el flujo que debe de seguir esa información planificada por él. De esta manera se garantiza un proceso de concepción y ejecución de la planificación ordenada, más centralizado y con mayor probabilidad de alcanzar con éxito los objetivos de la entidad.

1.3.1. Medios y herramientas para planificar actividades y objetivos.

Existen diversas herramientas para efectuar la planificación del plan de trabajo, administración y seguimiento de actividades regulares. Se pueden mencionar muchas opciones de sistemas destinados a la planificación de actividades, en ambientes escritorio, web, dirigidos a agendas electrónicas y a la telefonía móvil. A continuación se mencionan las más utilizadas.

GanttProject.

Es una iniciativa de código abierto con ambiente escritorio para representaciones esquemáticas de la distribución de las tareas a corto, mediano y largo plazo. Permite distribuir las actividades por personas o recursos enmarcados en metas o proyectos. Esta herramienta no permite planificar por objetivos de acuerdo al modelo de planificación cubano, pues aunque permite la representación de metas, no contiene el potencial para representar los objetivos con sus criterios de medidas, las ARC y menos permite las operaciones sobre las actividades que dependen de las relaciones entre estos elementos.

OpenProj.

Otra aplicación libre de código abierto y ambiente escritorio en las que se pueden planificar detalladamente cada tarea, con tareas sucesoras y predecesoras, pero destinada también a la gestión de proyectos de forma fácil, con buena interoperabilidad con Microsoft Project. Al igual que GanttProject, es utilizada ampliamente en este sentido pero tampoco se ajusta al modelo cubano de planificación.

eGroupware.

Solución de trabajo en grupo con ambiente web, de código abierto. Está escrita en PHP utilizando bases de datos, tales como PostgreSQL o MySQL. Incluye un calendario, una libreta de direcciones, un gestor de contactos, un cliente de correo electrónico IMAP⁵, un InfoLog,

⁵ **IMAP**: Internet Message Access Protocol es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor.

funciones de CRM⁶, un gestor de proyectos, un gestor de recursos, un gestor de ficheros, una plantilla de tiempos, un wiki, una base de conocimiento y un motor de flujos de trabajo.

Estas herramientas no permiten la gestión de la configuración necesaria para garantizar el proceso de Organización y simular las estructuras. Se basan en el trabajo directo con los usuarios, y la asignación de responsabilidades no parte de la especificación del rango de acción o dominio de planificación, lo que no admite la recuperación de la información planificada en base a la entidad, pues al dejar de existir el usuario se pierde la información planificada generada por él. Estas herramientas tampoco permiten el manejo de objetivos, criterios de medidas e indicadores de la manera establecida en la Instrucción No1, ni el registro de órdenes, indicaciones, acuerdos o resoluciones. Es válido destacar algunos aspectos positivos, los cuales son referencias para cualquier sistema que desee automatizar la planificación de actividades. Estos sistemas brindan funcionalidades para la visualización gráfica de eventos y actividades, mecanismos de interoperabilidad mediante estándares internacionales para la interacción con Microsoft Outlook y Excel. Igualmente permite la generalización de los nomencladores o clasificaciones de los elementos de la planificación. En el caso del eGroupware, fue el primer prototipo analizado y personalizado por el UCID y puesto en prueba en el MINFAR en el 2009, y aunque si permitía planificar objetivos con sus principales aspectos, no respondía a la operatividad que demanda el plan de la organización, aunque si lo hacía para el plan personal.

Excel, Word, Outlook y Access.

Herramientas del paquete Office de Microsoft de fácil utilización. En el caso de Outlook gestionaba las actividades como eventos, luego de ser recibidas las mismas en una plantilla Excel con un formato determinado con tablas y filtros. Con estas herramientas la información planificada que se recibía no coincidía con las clasificaciones establecidas, y el filtrado de la información no se podía ejecutar adecuadamente. Como medio de planificación, tiene sus ventajas comparado al proceso manual en hojas, pero no permite elaborar la organización, y menos llevar el seguimiento y control de la planificación como debe de ser.

Redmine.

Es una herramienta para la gestión de proyectos que incluye un sistema de seguimiento de incidentes con seguimiento de errores. Es utilizado ampliamente en la Universidad de las Ciencias Informáticas para la planificación de proyectos y el plan de trabajo, pero tampoco se ajusta como debe de ser al modelo de planificación establecido. En el caso del CEIGE, se toman

⁶ **CRM**: customer relationship management es un modelo de gestión de toda la organización, basada en la orientación al cliente (u orientación al mercado según otros autores).

Capítulo 1. Fundamentación Teórica

las fases como ARC, las iteraciones como objetivos y las actividades pueden ser peticiones, indicadores o actividad netamente según el tipo configurado. Aun así, las personalizaciones realizadas al mismo han permitido su utilización en un escenario más amplio, pero no responde a la Planificación por Objetivos ni a sus principales procesos, y no permite modelar las estructuras, trabaja directamente con usuarios, grupos y roles de acceso a funcionalidades.

PTRAB.

Primera herramienta orientada al modelo de Planificación por Objetivos cubano. Esta herramienta con ambiente de escritorio, entre sus principales ventajas tiene las funcionalidades para registrar planes, ARC, objetivos, criterios de medidas y actividades, permite la recuperación de la información en los formatos establecidos, posibilita llevar un control de las modificaciones a la planificación aprobada, tratamiento de niveles de planificación, entre otras. Ha estado utilizándose hasta el presente, pero muestra grandes desventajas pues no ha sabido evolucionar a la par del modelo de planificación cubano, que ha crecido en cuanto al nivel de exigencia y control, pues posee grandes limitaciones tecnológicas como una arquitectura no escalable. Esta herramienta no permite la reutilización a gran escala de los elementos planificados (6). Respecto al proceso de Organización, no trabaja con la modelación de entidades ni dominios de planificación, cada dominio es el computador donde está instalado, y aunque permite exportar la información en ficheros para garantizar el intercambio de la información, no garantiza la compartimentación requerida en esa información destinada a la configuración.

Hasta ahora se puede apreciar que las desventajas principales de estas herramientas radican, entre otras, en la gestión del proceso de organización, pues no eliminan el usuario-dependencia de la información, en su mayoría no representan los niveles de planificación y el grado de asignación de responsabilidades que brindan, limitan la realización de la planificación conjunta. Teniendo en cuenta esto, surge la primera versión de SIPAC, liberada en el 2010, y puesta en prueba experimental en entidades de las FAR. Sus principales ventajas son:

-  Presenta una arquitectura orientada a componentes, que posibilita, dentro de la misma plataforma tecnológica, generalizar la solución e ir agregando nuevos componentes según la necesidad que presente el cliente.

Capítulo 1. Fundamentación Teórica

- ✚ Utiliza las mismas tecnologías que Cedrux⁷, por lo que puede ser fácilmente integrada con otras herramientas de ese paquete para la obtención de recursos planificables y contables.
- ✚ Brinda todo un mecanismo de gestión de estructuras, dominios, roles, entre otros aspectos que son la base para la compartimentación de la información.
- ✚ Responde totalmente a lo establecido por la Instrucción No1, y se puede utilizar también para planificar el plan personal, con o sin objetivos, criterios e indicadores.
- ✚ Permite ejecutar procesos de aprobación-conciliación en tiempo real o “justo a tiempo”, pues está pensado para funcionar en red, con bases de datos centralizadas.

En cambio, no responde totalmente como debe de ser al proceso de Organización dentro de la planificación en las entidades cubanas. SIPAC no cuenta con la personalización de las estructuras modeladas hacia conceptos más operativos y más fáciles de utilizar en determinados niveles de la planificación. La compartimentación generada por el módulo encargado de gestionar la seguridad, basado en dominios y usuarios con roles dentro de ese dominio restringe el acceso a funcionalidades, pero no a los niveles de información. Por otra parte, a pesar de que SIPAC utiliza los módulos o subsistemas de Cedrux y este tiene un módulo para la configuración general del sistema, éste no permite manejar la compartimentación de las clasificaciones de los elementos de la planificación como necesitan los planificadores, ni permite establecer las responsabilidades y atribuciones en materia de planificación. El módulo Configuración de Cedrux está diseñado para realizar la configuración de aspectos generales relacionados con la contabilidad, finanzas, logística, etc. que le puedan servir de manera general al resto de los procesos de negocios informatizados dentro del mismo paquete.

A modo de conclusión, se puede afirmar que el proceso de Organización dentro de la Planificación por Objetivos en las entidades cubanas no está informatizado con todos los requerimientos necesarios y establecidos en la Instrucción No1, que proponen un adecuado tratamiento a la definición de los niveles de la planificación, a la compartimentación y a la generalización de aspectos claves a utilizar dentro del proceso de Concepción-Ejecución de la planificación. Las herramientas existentes y descritas anteriormente no responden a todas estas necesidades, a pesar de que SIPAC representa el camino hacia la evolución de la informatización de la Planificación por Objetivos en las entidades cubanas, aún no cuenta con funcionalidades que permitan que este proceso de planificación se ejecute sobre las bases de

⁷ **Cedrux:** Sistema Integral de Gestión, novedosa aplicación informática que se construye especialmente adaptado al control de los recursos empresariales de las entidades, conocido como el ERP cubano. Fue desarrollado siguiendo los principios de independencia tecnológica.

una correcta organización, compartimentación y entendimiento común de la información entre y hacia los distintos niveles deseados.

1.4. Lenguajes propuestos para el desarrollo de la solución.

Los lenguajes propuestos para el desarrollo de la solución son los lenguajes de modelado y desarrollo establecidos por la dirección de CEIGE y la UCID como resultado de acuerdo mutuo en aras de la integración y cooperación, de acuerdo a los principios de libertad tecnológica y tecnologías a utilizar por ambos.

1.4.1. Lenguajes de Modelado.

Los lenguajes de modelado se conocen como aquellos que estandarizan símbolos de modo que sirvan para ordenar un diseño del software, por lo que se propone para el desarrollo de la solución:

UML

UML (Unified Modeling Language) es un lenguaje de modelado unificado que permitirá modelar, construir y documentar los elementos que forman un sistema de software orientado a objetos, de un modo estándar incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema. Los elementos concretos como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reutilizables posibilitarán la división del proyecto en un número de diagramas que representarán las diferentes vistas del proyecto. Estos diagramas juntos se encargarán de representar la arquitectura del proyecto, ya que 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 (7).

1.4.2. Estándar de Modelado.

BPMN

BPMN (Business Process Modeling Notation) es un estándar de modelado de procesos de negocio donde se presentan gráficamente las diferentes etapas del proceso del mismo. Esta notación ha sido diseñada 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, por lo que la

combinación de 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 (8).

1.4.3. Lenguaje de Marcado de Hipertexto.

HTML.

HTML (HyperText Markup Language) es el lenguaje de marcado que se utilizará para el desarrollo de páginas clientes de la aplicación. Permitirá describir la estructura y el contenido en forma de texto, además de complementar el texto con objetos tales como imágenes. HTML se escribe mediante etiquetas, puede incluir uno o varios scripts, como por ejemplo JavaScript o PHP, los cuáles pueden afectar el comportamiento del HTML. Su principal desventaja es que todos los navegadores no interpretan el código HTML de la misma manera (9).

1.4.4. Lenguajes de Desarrollo del lado del cliente.

JavaScript.

Java Script es un lenguaje interpretado, que no requiere compilación, se utilizará integrado en un navegador web para el desarrollo de páginas web dinámicas ya que está basado en prototipos. Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web por lo que este lenguaje de programación permitirá que las páginas web sean interactivas y con elementos mucho más dinámicos. Y una de sus principales ventajas es que interactúa perfectamente con códigos basados en HTML (10).

CSS (Hojas de estilo en Cascada).

CSS (Cascading Style Sheets) es el lenguaje que describirá la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación y cómo se va a mostrar un documento en pantalla (dígase tamaños, colores, iconos, imágenes, tipografías, espacios y bordes). Se basa en una serie de reglas que regirán el estilo de los elementos en los documentos estructurados y que formarán la sintaxis de las hojas de estilo. Constituirán 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 (11).

XML (Extensive Markup Language).

XML se utilizará para proveer un conjunto de reglas, normas y convenciones para diseñar formatos de texto para datos estructurados que van desde las hojas de cálculo o las libretas de

direcciones de Internet, hasta parámetros de configuración, transacciones financieras o dibujos técnicos. Permite que los diseñadores creen sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones. Facilita el análisis y procesamiento de los documentos XML creados por terceros. XML se trata de un estándar del **W3C**⁸ 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 (12).

1.4.5. Lenguajes de Desarrollo del lado del servidor.

PHP v5.2.

PHP permitirá el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto y a que es de código abierto, es el más popular y extendido en la web. Es capaz de realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas en un lenguaje distinto al HTML. Esto se debe a que PHP ofrece un extenso conjunto de funciones para la explotación de bases de datos sin complicaciones. 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 (13).

1.4.6. Frameworks o Marcos de Trabajo.

Es la estructura base sobre la cual otro proyecto de software puede ser desarrollado. Del mismo modo se define como una agrupación de criterios, prácticas y consejos para contrarrestar alguna problemática en particular, el cual sirve como bandera para resolver otros problemas de índole similar. Puede incluir apéndices como bibliotecas, soportes de programas y lenguajes interpretados para ayudar a construir y unir los diferentes componentes o partes de un proyecto (14).

Sauxe.

Entre sus principales características figura su arquitectura en capas, proporcionando una mayor integración, flexibilidad y agilidad en el proceso de desarrollo. Se ha elegido gracias al desarrollo de tecnologías propias que tienen como base otros marcos de trabajo como Extjs para la capa

⁸ **W3C:** El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web.

de presentación, ZendFramework para la lógica de negocio y Doctrine para el acceso a datos. Fue desarrollado en la UCI como resultado de los principios de independencia tecnológica (15).

ExtJs 2.2.

Basado en la Programación Orientada a Objetos (POO) donde cada instancia posee sus atributos, métodos y eventos. Puro JavaScript en el que mediante el uso extenso del DOM⁹ genera y carga todos los objetos HTML. Hace uso extensivo de Ajax para el intercambio de datos desde el cliente hacia el servidor, que viene implementado en componentes predefinidos los cuáles permiten utilizar, reutilizar y adaptar según el nivel de complejidad de nuestras aplicaciones y nuestra abstracción o proyección (16).

ZendFramework.

Posee funcionalidades en potencia por ser precisamente de código abierto, además de facilidades de uso como mecanismos de filtrado y validación de entrada de datos, conversión de estructuras de datos PHP a JSON¹⁰ y viceversa para ser utilizadas en aplicaciones AJAX. Con su sistema de cache proporciona el almacenamiento de diferentes datos y los componentes que forman la infraestructura del patrón Modelo-Vista-Controlador. Fue diseñado para la versión 5 de PHP, pero posee excelentes capacidades de ampliación (17).

Doctrine.

Potente sistema ORM¹¹ para PHP 5.2 o superior que tiene una capa de abstracción a base de datos, permitiendo escribir si se desea las preguntas o consultas a base de datos en POO, eliminando la duplicación del código y aumentando la flexibilidad. 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 (18).

1.5. Tecnologías y Herramientas de Desarrollo.

AJAX.

AJAX (JavaScript asíncrono y XML). Se trata de una técnica de desarrollo web que permitirá crear aplicaciones que se ejecutan en el cliente. Esta tecnología permitirá aumentar la

⁹ **DOM:** Document Object Model (en español Modelo de Objetos del Documento) es esencialmente una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

¹⁰ **JSON:** Acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

¹¹ **ORM:** Mapeo objeto-relacional (Object-Relational mapping) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia.

interactividad, velocidad y usabilidad en las aplicaciones, gracias a la posibilidad de realizar cambios sobre las páginas sin necesidad de recargarlas. JavaScript es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de Ajax, mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales (19).

Subversion.

Subversion es la herramienta que se utilizará para el control de versiones libre y de código abierto, manejará ficheros y directorios a través del tiempo. Esto le permitirá recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos. Subversión puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos ordenadores. Se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones, puesto que el trabajo se encuentra bajo el control de versiones, no hay razón para temer por que la calidad del mismo vaya a verse afectada por la pérdida de ese conducto único si se ha hecho un cambio incorrecto a los datos, simplemente deshace ese cambio (20).

Visual Paradigm.

Se empleará Visual Paradigm en su versión 1.6.5 ya que es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y documentación. La herramienta UML CASE¹² proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML (21).

Netbeans.

Se utilizará como entorno de desarrollo el Netbeans en su versión 6.8 ya que es un reconocido entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. Este IDE¹³ está formado por código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones web, empresariales, de escritorio y móviles utilizando múltiples

¹² **CASE:** (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

¹³ **IDE:** Un entorno de desarrollo integrado (integrated development environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien poder utilizarse para varios.

plataformas entre la que se encuentra PHP, JavaScript y Ajax. Se integra con varias herramientas como el Subversion y servidores web, presenta una gran estabilidad y hace uso de plugins¹⁴ para ampliar sus funcionalidades, lo que le da una gran facilidad de uso (22).

Apache.

Se utilizará como servidor web Apache pues actualmente es el más implantado entre los distintos servidores que ofertan servicios web en Internet y uno de los más populares del mercado. Una de las principales motivaciones para utilizar éste servidor es su condición de aplicación libre y descargable de forma gratuita de la web. Su instalación es bastante sencilla con una alta configurabilidad en la creación y gestión de registros de actividad. Apache permite la creación de ficheros de registro a medida del administrador y se puede tener un mayor control sobre lo que sucede en el servidor (23).

PostgreSQL.

Se utilizará como gestor de base de datos PostgreSQL en su versión 8.3.3. Este es un sistema de gestión de bases de datos relacional orientada a objetos. Tiene prácticamente todo lo que tienen los gestores comerciales, haciendo de él una muy buena alternativa GPL¹⁵. Es una herramienta de código abierto, de bajo coste y multiplataforma. Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc. y permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. Además se destaca en ejecutar consultas complejas, consultas sobre vistas, subconsultas y uniones de gran tamaño (24).

Mozilla Firefox.

Mozilla Firefox es un navegador de Internet libre y de código abierto y es un navegador multiplataforma. Su código fuente es software libre, publicado bajo una triple licencia GPL/LGPL¹⁶/MPL¹⁷. Es un programa independiente del sistema operativo, con un código mejorado por decenas de programadores que lo hacen más sólido y estable. Permite la navegación por pestañas, una forma increíblemente eficaz e intuitiva para tener varias páginas disponibles al mismo tiempo, sin saturar la barra de tareas. Además tiene gran compatibilidad

¹⁴ **Plugins:** es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

¹⁵ **GPL:** La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License. está orientada principalmente a proteger la libre distribución, modificación y uso de software.

¹⁶ **LGPL:** Lesser General Public License pretende garantizar la libertad de compartir y modificar el software cubierto por ella, asegurando que el software es libre para todos sus usuarios.

¹⁷ **MPL:** La licencia pública de Mozilla (Mozilla Public License o MPL) es una licencia de código abierto y de software libre.

con los estándares web (señálese HTML, XML, CSS y JavaScript) y permite la búsqueda directa e incorpora un espacio en su interfaz para realizar búsquedas no sólo en uno, sino en varios motores de búsquedas (25).

1.6. Estándares de Código.

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. 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. Siguiendo esta idea, podemos definir 3 partes principales dentro de un estándar de programación:

- Convención de nomenclatura: Como nombrar variables, funciones, métodos, etc.
- Convenciones de legibilidad de código: Como indentar el código, etc.
- Convenciones de documentación: Como establecer comentarios, archivos de ayuda, etc.

✚ 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.

✚ Nomenclatura según el tipo de clases:

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

Clases de los modelos:

- **Business (Negocio)**: Las clases que se encuentran dentro de Business después del nombre llevan la palabra: "Model". Ejemplo: GestionarGruposModel.
- **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: DatGrupo.
- **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: BaseDatGrupos.

✚ 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: **eliminarGruposAction()**.
- En las clases de los modelos: Las funcionalidades se nombran de manera que al leerlo se identifique su propósito. Ejemplo de función: **eliminarGrupos()**.

✚ 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.

1.7. Modelo de Desarrollo.

El proyecto SPA, como anteriormente se había mencionado en la introducción del presente trabajo de diploma, surge como fruto de la cooperación entre el CEIGE y la UCID. Teniendo en cuenta que la UCID lleva alrededor de 9 años de experiencia obteniendo resultados satisfactorios. La presente investigación se acoge al empleo del modelo de desarrollo utilizado en dicha entidad. Este modelo se encuentra descrito en el documento Proceso de Desarrollo y Gestión de Proyectos de Software, en su primera versión en el cual se logra mediante la combinación de los modelos basados en Componentes, el Iterativo y el Incremental. Como principales características presenta el incremento del resultado parcial del sistema ya probado, integrado y estable, así como una mayor reutilización de las soluciones. Este modelo está definido por procesos los cuales se representan a continuación:



Figura 1.2: Procesos del Modelo de Desarrollo a utilizar.

Cada proyecto tiene un ciclo de vida regido por una serie de fases en forma de secuencia, una a continuación de la otra, las cuales se representan a continuación:

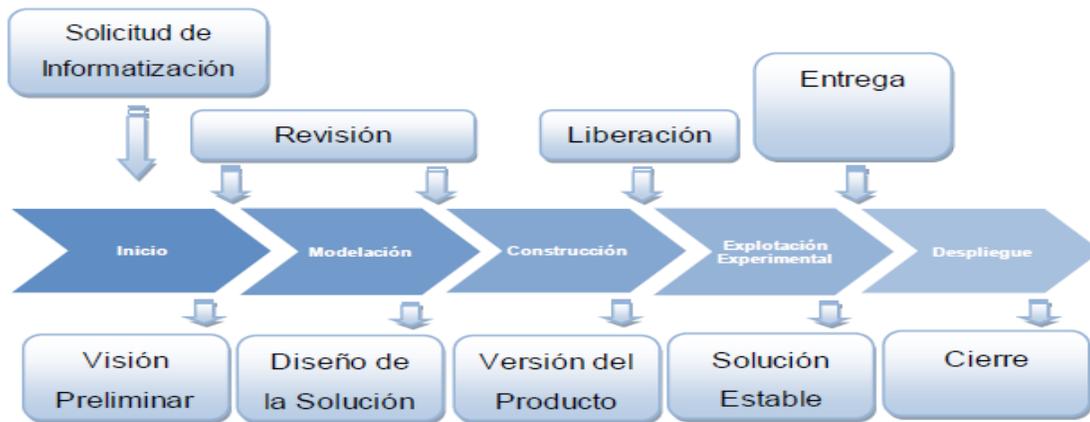


Figura 1.3: Fases o etapas del ciclo de vida de un proyecto.

1.8. Conclusiones del Capítulo.

Por todo lo expuesto en este capítulo se puede afirmar que el proceso de planificación en las entidades cubanas ha tomado un nivel de madurez de la aplicación del modelo cubano para dicho proceso, modelo híbrido obtenido mediante la unión de la Dirección por Objetivos y la Planeación Estratégica. Dentro de este modelo, la fase de Organización juega un papel importante como la base del resto de los procesos, es la piedra angular para una correcta dirección y control de la planificación. Se expusieron además las principales características del

Capítulo 1. Fundamentación Teórica

estado del arte referente al tema en cuestión, partiendo del análisis del comportamiento que ha tenido la planificación, su evolución, su proyección en escenarios nacionales e internacionales y su materialización en herramientas informáticas con el fin de proporcionar mayor fortaleza en la ejecución de dichos procesos. Se presentaron las técnicas y tecnologías propuestas para el desarrollo de la solución y de manera general un breve resumen del modelo de desarrollo a seguir.

Capítulo 2. Propuesta de Solución.

Introducción del Capítulo.

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, la definición de los requisitos y la arquitectura. Se figuran algunos de los artefactos propuestos en la fase de Modelación definida en la metodología de desarrollo aplicada y se fundamentan los patrones de diseños y modelos empleados en su elaboración. Igualmente se expondrá la estructura física del módulo Configuración y el Diagrama de Despliegue de la solución propuesta así como los estándares de codificación utilizados en la aplicación y la implementación de dicho módulo.

2.1. Propuesta de Solución.

SIPAC es la solución informática que se encuentra en estos momentos en prueba experimental, para informatizar el proceso de planificación de actividades y objetivos en las entidades cubanas bajo el modelo cubano de Planificación por Objetivos. En esta prueba experimental se arrojaron una serie de no conformidades relacionadas con el proceso de Organización, sobre todo en la parte concerniente a la particularidad de cada uno de los planificadores según el nivel, de poder agrupar a los subordinados en conceptos sobre los cuales se lograra depositar cualquier configuración e inclusive cualquier planificación. Asimismo poder definir permisos y niveles de interacción a la hora de establecer planificaciones conjuntas y hacer global al resto de los niveles aquellos atributos más utilizados en la planificación, de acuerdo al dominio de planificación y el flujo que debe de seguir la información generada en los procesos Concepción y Ejecución.

El presente trabajo de diploma está dirigido a aumentar el nivel de usabilidad y robustez de SIPAC en este aspecto, mediante la incorporación del módulo Configuración que contará con prestaciones para:

- ✚ La personalización de las estructuras modeladas hacia conceptos más operativos y más fáciles de utilizar en determinados niveles de la planificación.
- ✚ La asignación de responsabilidades con el fin de establecer la compartimentación sobre los niveles de la información y permitir realizar operaciones de planificación conjunta.
- ✚ Las funcionalidades que permitan heredar las clasificaciones a utilizar en la planificación de los elementos elaborados en el nivel superior, permitirá

reimplementarlas e implementar las propias del nivel, y generalizarlas hacia los niveles subordinados propiciando así un mecanismo de entendimiento común entre los niveles de la planificación.

2.2. Modelación del negocio.

A continuación se modela el proceso de negocio correspondiente a la Definición de la Organización, Generalización y Establecimiento de las Responsabilidades empleando la notación BPMN para la representación de procesos de negocio, la cual posibilita una representación visual de éste, especificando la relación entre los roles, artefactos y eventos que intervienen en su desarrollo.

2.2.1. Diagramas de procesos de negocio (BPMN).

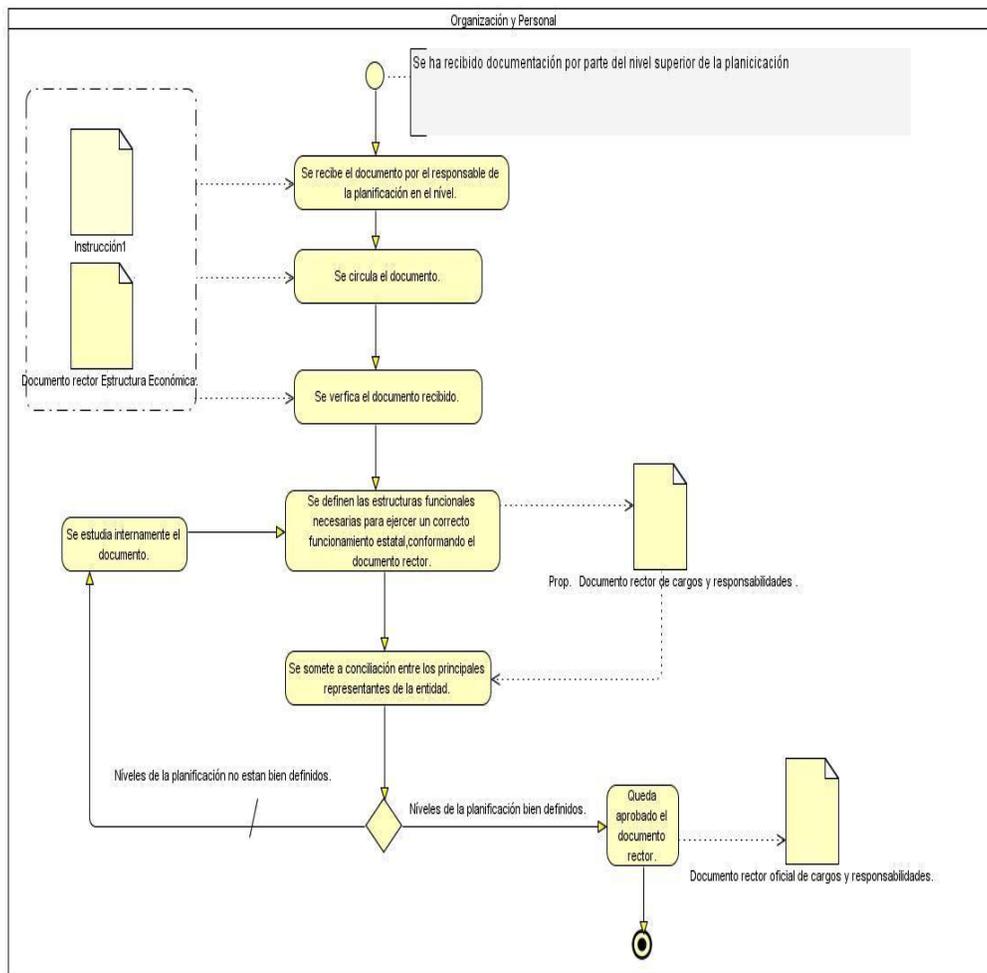


Figura 2.1: Definición de la Organización.

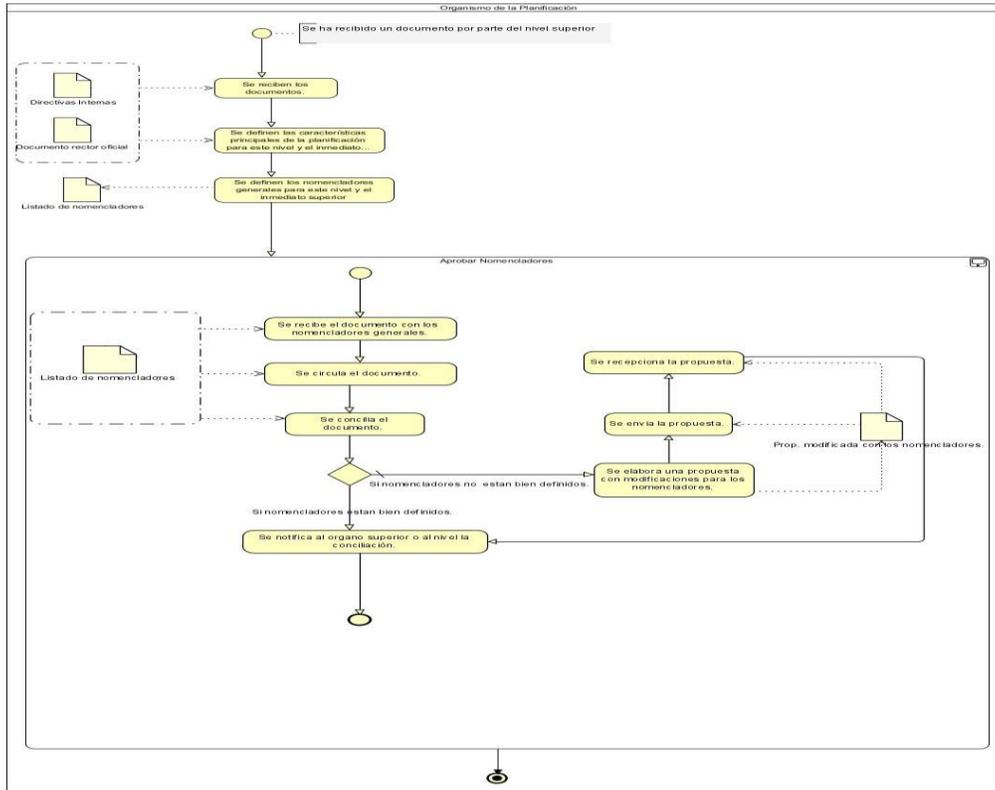


Figura 2.2: Generalización.

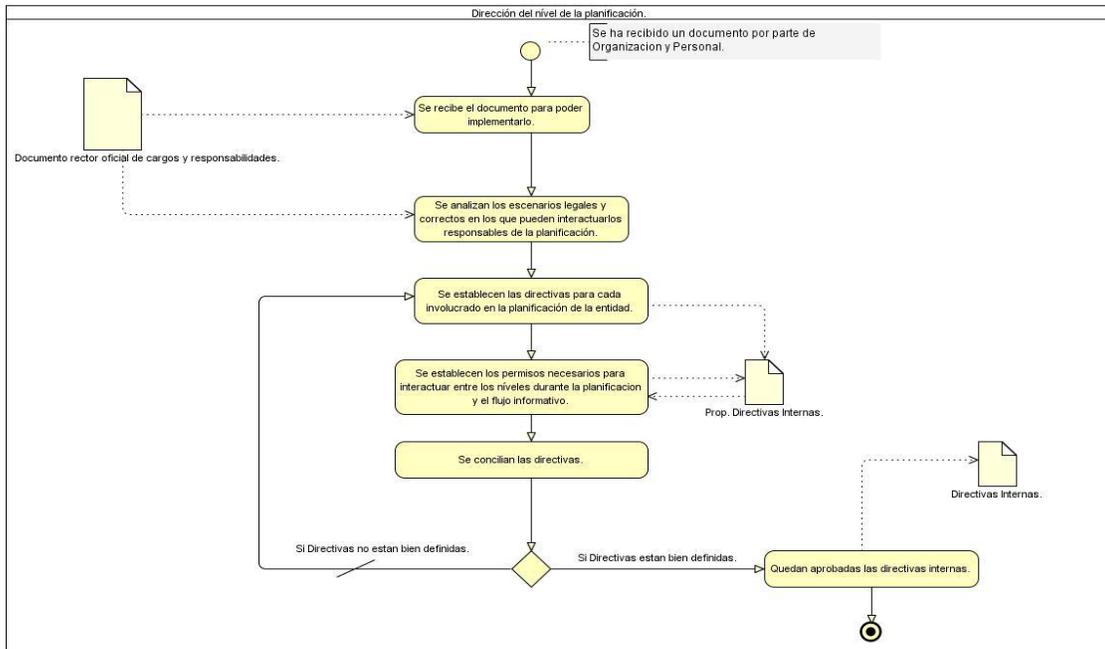


Figura 2.3: Establecimiento de las Responsabilidades.

2.2.2. Especificación de los procesos de negocio.

Tabla 1: Descripción del proceso de negocio correspondiente a la Definición de la Organización.

Objetivo	Definir y aprobar los niveles de la planificación (departamento, áreas y cargos para el proceso).
Evento(s) que lo genera(n)	Se ha recibido documentación por el responsable de la planificación en organización y personal.
Pre condiciones	Se ha emitido una documentación que orienta la planificación en el nivel superior.
Marco legal	N/A.
Cientes internos	N/A.
Cientes externos	N/A.
Entradas	Documento Rector (Documento Word). Instrucción1 (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 de los niveles de planificación.	
2. Se verifica el documento recibido. Este documento se identifica para conocer que sea el Documento Rector de Estructura Económica y el documento de Instrucción1.	
3. Se definen las estructuras funcionales necesarias para ejercer un correcto funcionamiento estatal, conformando la propuesta del documento rector de cargos y responsabilidades.	
4. Se somete a conciliación entre los principales representantes de la entidad. Si los niveles de la planificación están bien definidos se ejecuta el flujo básico <u>Queda aprobado el documento rector</u> . Si por el contrario los niveles de la planificación no están bien definidos ver el flujo alternativo 1.	
5. Queda aprobado el documento rector.	
6. Concluye el proceso.	
Pos-condiciones	
1. N/A.	
Salidas	
1. Documento rector oficial de cargos y responsabilidades.	
Flujo alterno	

Flujo alternativo 1. Niveles de la planificación no están bien definidos.

1. 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 los niveles de la planificación. Se ejecuta el flujo básico 3.

Pos-condiciones

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

Salidas

2. Documento rector oficial de cargos y responsabilidades.

Asuntos pendientes

1. N/A.

Tabla 2: Descripción del proceso de negocio correspondiente a la Generalización.

Objetivo	Definir y aprobar los nomencladores generales para este nivel del organismo de la planificación y el inmediato superior.
Evento(s) que lo genera(n)	Se ha recibido los documentos de las Directivas Internas.
Pre condiciones	Se ha emitido una documentación en el organismo de la planificación.
Marco legal	N/A.
Clientes internos	N/A.
Clientes externos	N/A.
Entradas	Directivas Internas (Documento Word). Documento Rector Oficial de cargos y responsabilidades (Documento Word).
Flujo de eventos	
Flujo básico	
7.	Se reciben los documentos. Se reciben los documentos en formato impreso o digital y se circula a todos los involucrados del mismo nivel de dirección que participarán en el establecimiento de los nomencladores generales.
8.	Se definen las características principales de la planificación para este nivel y el inmediato superior.
9.	Se definen los nomencladores generales para este nivel y el inmediato superior. Se ejecuta el flujo paralelo 1.
10.	Concluye el proceso.

Pos-condiciones

2. N/A.

Salidas

3. Listado de Nomencladores.

Flujos paralelos

Flujos paralelo 1 Modificar Nomencladores.

1. Se recibe el documento con los nomencladores generales.
2. Se circula el documento en formato impreso o digital a todos los involucrados del mismo nivel de dirección.
3. Se concilia el documento. Los nomencladores que han sido propuestos por los implicados se llevan a un proceso de concertación entre ellos para definir los que finalmente serán incluidas en la nueva propuesta. Si los nomencladores están bien definidos se notifica al nivel u órgano superior la conciliación. Si por el contrario los nomencladores no están bien definidos se ejecuta el flujo alterno 3.
4. Concluye el proceso.

Pos-condiciones

1. Se ha establecido el listado de nomencladores.

Salidas

1. Listado de Nomencladores.

Flujos alternos

Flujo alternativo 3. Nomencladores no están bien definidos.

2. Se elabora una propuesta con las modificaciones para los nomencladores.
3. Se envía la propuesta.
4. Se recepciona la propuesta. Se ejecuta el flujo básico1.
5. Concluye el proceso.

Pos-condiciones

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

Salidas

1. Propuesta modificada con los nomencladores.

Tabla 3: Descripción del proceso de negocio correspondiente al Establecimiento de Responsabilidades.

Objetivo	Definir y aprobar las directivas para todos los involucrados en la entidad.
Evento(s) que lo genera(n)	Se ha recibido documentación por el responsable de la planificación en el nivel superior.

Capítulo 2: Propuesta de Solución

Pre condiciones	Se ha emitido una documentación que orienta la planificación en el nivel superior.
Marco legal	N/A.
Cientes internos	N/A.
Cientes externos	N/A.
Entradas	Documento Rector Oficial de cargos y responsabilidades (Documento Word).
Flujo de eventos	
Flujo básico	
1.	Se recibe el documento para poder implementarlo. Se circula el documento en la dirección del nivel de la planificación.
2.	Se analizan los escenarios legales y correctos en los que pueden interactuar los responsables de la planificación.
4.	Se establecen las directivas para cada involucrado en la planificación de la entidad.
5.	Se establecen los permisos necesarios para interactuar entre los niveles durante la planificación y el flujo informativo.
6.	Se concilian las directivas. Si directivas están bien definidas se ejecuta el flujo básico <u>Quedan aprobadas las directivas internas</u> . Si por el contrario las directivas no están bien definidas ver el flujo alternativo 2.
7.	Quedan aprobadas las directivas internas.
8.	Concluye el proceso.
Pos-condiciones	
N/A.	
Salidas	
1. Directivas Internas.	
Flujo alterno	
Flujo alternativo 2.Directivas Internas no están bien definidas.	
1.	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 restablecerla propuesta de las directivas. Se ejecuta el flujo básico 3.
Pos-condiciones	
3.	Se ha rechazado el documento y devuelto al órgano de dirección que lo originó.
Salidas	
9. Directivas Internas.	
Asuntos pendientes	

10. N/A.

2.3. Conceptos manejados en el negocio.

En el negocio concerniente al proceso de Organización se manejan varios conceptos, los cuales tienen una estrecha relación entre sí y contribuyen a la conformación de otros perteneciente al resto de los procesos. Entre los principales conceptos se encuentran: Grupo, Usuario, Elemento de seguridad, Permiso, Lenguaje común, Nomenclador entre otros. A continuación se representan estos conceptos, sus relaciones entre ellos y con el resto.

2.3.1. Modelo Conceptual.

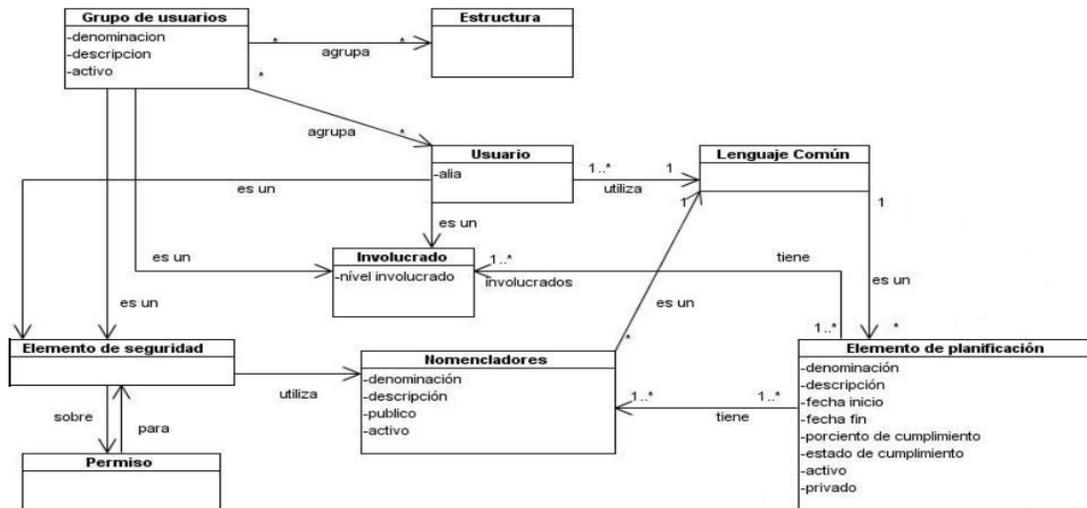


Figura 2.4: Modelo Conceptual correspondiente al módulo Configuración.

2.4. Requisitos.

Un requisito en cualquier plano no es más que la capacidad para desarrollar determinadas actividades y funciones con el fin de resolver un problema o alcanzar determinado objetivo. En el plano de la informática un requisito se puede definir como:

- ✚ 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. (IEEE 610.12-1990)

Este trabajo se acoge a esta definición. Para saber que requerimientos debe de poseer el sistema es necesario realizar una captura de requisitos, de manera que se pueda extraer de la fuente, que en este caso son los clientes y la petición inicial del proyecto, las necesidades que debe de cubrir la solución propuesta.

2.4.1. Técnica de Captura de Requisitos.

Para llevar a cabo esta actividad fue necesario aplicar varias técnicas, las que se listan a continuación:

- ✚ **Tormenta de ideas:** Consisten en 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.
- ✚ **Entrevistas:** Son entrevistas con el cliente, donde se realizan preguntas con el objetivo de obtener toda la información posible sobre la visión que el cliente tiene de los requisitos y comprender los propósitos de la solución buscada.
- ✚ **Talleres:** Son reuniones muy parecidas a las ejecutadas en la tormenta de ideas, pero estas son sesiones más intensivas y estructuradas donde se realiza una preparación previa y es dirigida por un experto para poder detallar cada requisito.

2.4.2. Listado de Requisitos Funcionales Identificados.

Los requisitos funcionales son declaraciones de las prestaciones que debe proporcionar el sistema, así como la reacción que éste debe de tener ante situaciones particulares. En algunos casos, los requisitos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer.

Seguidamente se muestran cuáles fueron los requisitos funcionales capturados y que deben de ser implementados por el sistema¹⁸.

Tabla 4: Listado de Requisitos Funcionales Identificados.

Gestionar grupos de usuarios por rol	Gestionar permisos	Gestionar categorías de la actividad	Gestionar categorías del plan	Gestionar procedencias del FIP	Gestionar tipos de FIP	Gestionar tipos de actividad	Gestionar unidades de medida del criterio	Gestionar niveles de acceso del FIP
Adicionar grupo de usuarios por rol.	Listar permisos por defecto	Adicionar categoría de la actividad.	Adicionar categoría del plan.	Adicionar procedencias del FIP.	Adicionar tipos de FIP.	Adicionar tipos de actividad.	Adicionar unidades de medida del criterio.	Adicionar niveles de acceso del FIP.
Modificar grupo de usuarios	Listar permisos otorgados	Modificar categoría de la actividad.	Modificar categoría del plan.	Modificar procedencias del FIP.	Modificar tipos de FIP.	Modificar tipos de actividad.	Modificar unidades de medida	Modificar niveles de

¹⁸ La descripción de los requisitos Modificar Grupos de Usuarios por rol y Eliminar Grupos de Usuarios por rol se encuentran en los anexos del presente trabajo, el resto se encuentra archivados dentro del expediente de proyecto perteneciente a SIPAC.

Capítulo 2: Propuesta de Solución

por rol.	por el nivel superior.						del criterio.	acceso del FIP.
Eliminar grupo de usuarios por rol.	Listar usuarios subordinados	Eliminar categoría de la actividad.	Eliminar categoría del plan.	Eliminar procedencias del FIP.	Eliminar tipos de FIP.	Eliminar tipos de actividad.	Eliminar unidades de medida del criterio.	Eliminar niveles de acceso del FIP.
Desactivar grupo de usuarios por rol.	Configurar permisos sobre los subordinados.	Activar categoría de la actividad.	Activar categoría del plan.	Activar procedencias del FIP.	Activar tipos de FIP.	Activar tipos de actividad.	Activar unidades de medida del criterio.	Activar niveles de acceso del FIP.
Activar grupo de usuarios por rol.	Listar permisos otorgados a subordinados.	Desactivar categoría de la actividad.	Desactivar categoría del plan.	Desactivar procedencias del FIP.	Desactivar categoría del plan.	Desactivar tipos de actividad.	Desactivar unidades de medida del criterio.	Desactivar niveles de acceso del FIP.
Cambiar alias a usuarios.	Adicionar permiso a subordinado.	Buscar categoría de la actividad.	Buscar categoría del plan.	Buscar procedencias del FIP.	Buscar tipos de FIP.	Buscar tipos de actividad.	Buscar unidades de medida del criterio.	Buscar niveles de acceso del FIP.
Buscar grupo de usuarios por rol.	Modificar permiso a subordinado.	Listar categoría de la actividad.	Listar categoría del plan.	Listar procedencias del FIP.	Listar tipos de FIP.	Listar tipos de actividad.	Listar unidades de medida del criterio.	Listar niveles de acceso del FIP.
Listar grupos de usuarios por rol.	Eliminar permiso a subordinado	Activar categoría de la actividad desde el sistema.	Activar categoría del plan desde el sistema.	Activar procedencias del FIP desde el sistema.	Activar tipos de FIP desde el sistema.	Activar tipos de actividad desde el sistema.	Activar unidades de medida del criterio desde el sistema.	Activar niveles de acceso del FIP desde el sistema.
Listar usuarios de grupo de usuarios por rol.		Desactivar categoría de la actividad desde el sistema.	Desactivar categoría del plan desde el sistema.	Desactivar procedencias del FIP desde el sistema.	Desactivar tipos de FIP desde el sistema.	Desactivar tipos de actividad desde el sistema.	Desactivar unidades de medida del criterio desde el sistema.	Desactivar niveles de acceso del FIP desde el sistema.
		Generalizar categoría de la actividad	Generalizar categoría del plan.	Generalizar procedencias del FIP	Generalizar tipos de FIP.	Generalizar tipos de actividad.	Generalizar unidades de medida del criterio.	Generalizar niveles de acceso del FIP.

2.4.3. Especificación de Requisitos Funcionales.

Para la especificación de requisitos se utiliza la plantilla definida por el Departamento de Calidad de CEIGE, que consta con diferentes secciones para describir los flujos de eventos que conllevan a la ejecución la funcionalidad en el sistema y que se presentan a continuación:

Especificación de requisitos de Adicionar grupo de usuarios por rol.

Tabla 5: Adicionar grupo de usuarios por rol.

Precondiciones	Ha sido creada una estructura de trabajo, la cual está compuesta por entidades, áreas, cargos y usuarios.
Flujo de eventos	
Flujo básico	
1.	Se introducen los datos del grupo de usuarios: Denominación Descripción
2.	Se seleccionan en la estructura definida los usuarios que pertenecerán al grupo de usuarios.
3.	El sistema valida (ver validación 1) los datos introducidos.
4.	Si los datos son correctos el sistema los registra.
5.	El sistema confirma el registro de los datos.
6.	Concluye el requisito.
Pos-condiciones	
1	Se registró en el sistema un nuevo grupo de usuarios integrado por usuarios.
2	El grupo de usuarios registrado está activado.
Flujos alternativos	
Flujo alternativo 2.a Crear un grupo formado por otros grupos.	
1	Se selecciona de los grupos definidos los que van a pertenecer al nuevo grupo de usuarios.
2	Volver al paso 3 del flujo básico.
Pos-condiciones	
1	Se registró en el sistema un nuevo grupo de usuarios integrado por otros grupos de usuarios.
Flujo alternativo 4.a Información errónea.	
1	El sistema señala los datos erróneos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 3 del flujo básico.
Pos-condiciones	
1	N/A
Flujo alternativo 4.b Información incompleta.	

1 El sistema señala los datos vacíos y permite corregirlos.

2 El usuario corrige los datos.

3 Volver al paso 3 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	N/A.
-------------------	-----------------------------	------

	Extensiones	N/A.
--	--------------------	------

Conceptos	Grupo de usuarios	Visibles en la interfaz: Denominación Descripción Activado Utilizados internamente: N/A
	Usuario	Visibles en la interfaz: N/A Utilizados internamente: N/A

Requisitos especiales	N/A.
------------------------------	------

Asuntos pendientes	N/A.
---------------------------	------

2.4.4. Prototipos de Interfaz de Usuarios.

A continuación los prototipos de Interfaz de usuarios asociados al componente Grupos de usuarios por rol.

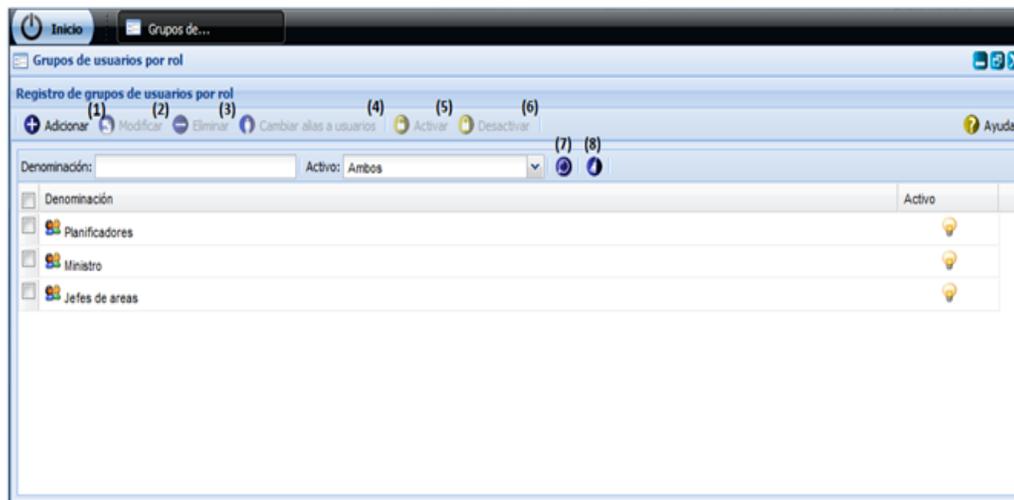


Figura 2.5: Prototipo de interfaz de usuario de Grupos de usuarios por rol.

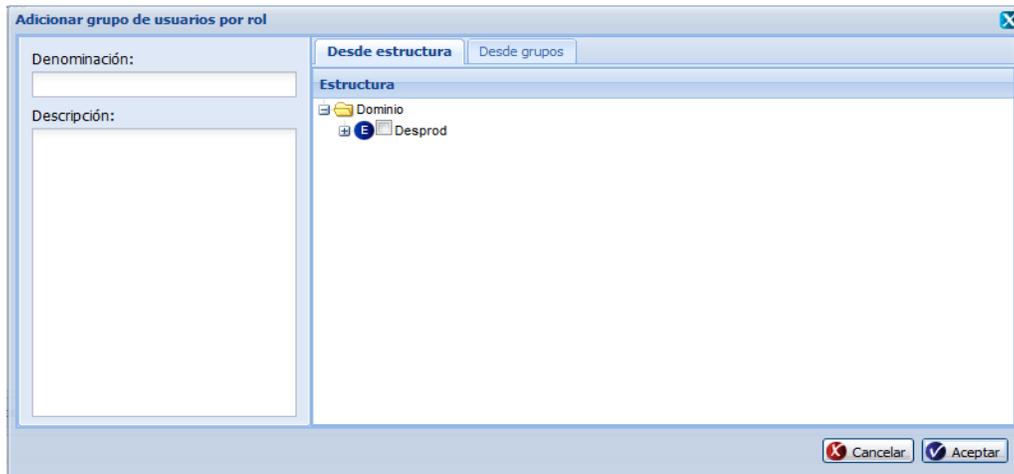


Figura 2.6: Prototipo de interfaz de usuario del requisito Adicionar grupo de usuarios por rol.

2.4.5. Validación de los requisitos funcionales.

Estos requisitos se encuentran validados por la dirección del proyecto SPA. Los documentos del Centro de Soluciones de Gestión concernientes al proceso de Planificación por Objetivos constituyen una constancia de validación de este proceso. **(Ver Anexo 8)**

2.4.6. Requisitos no funcionales.

Los requisitos no funcionales constituyen restricciones de las funciones ofrecidas por el sistema, como las restricciones de tiempo, el proceso de desarrollo y estándares. Este tipo de

requisitos se aplican al sistema en su totalidad y muy poco a características o servicios individuales. (26)

Los requisitos no funcionales de la solución son los definidos por el equipo de Arquitectura para el producto SPA, tomando como base los definidos para el producto Cedrux, definidos igual por el mismo equipo.

Usabilidad (USB)

El sistema podrá ser usado por cualquier usuario con conocimientos básicos de computación.

Rendimiento (REN)

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.

Seguridad (SEG)

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.

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.

Interfaz (INU)

El sistema debe contar con una interfaz fácil, amigable y sencilla que permita a los usuarios finales del mismo interactuar con este aun teniendo conocimientos básicos.

2.5. Arquitectura de la solución propuesta.

La estructura arquitectónica que se propone para la solución se encuentra definida según lo establecido por el departamento de Tecnología del proyecto ERP-Cuba, que es el encargado del diseño y evolución de la arquitectura del sistema Cedrux, del cual se hereda toda la visión arquitectónica por la naturaleza tan semejante de las soluciones.

Esta se organiza fundamentalmente en tres vistas, las cuáles a su vez pueden desagregarse en otras. 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.



Figura 2.7: Estructura Arquitectónica.

2.5.1. Principales aspectos arquitectónicos.

La característica principal de esta arquitectura es que surge como resultado de la combinación de varios estilos arquitectónicos, un híbrido, orientado a componentes para el desarrollo horizontal del sistema, debido a la complejidad que presenta el dominio de los negocios a incidir con el desarrollo de la aplicación (27). Esto posibilita alcanzar un mayor nivel de reutilización de las soluciones, y es una de las piezas fundamentales para la escalabilidad del sistema, permitiendo también la prueba por separado de cada uno de sus componentes sin afectar el funcionamiento del resto, solo en el área de integración, en caso de que tengan.

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 reinención (28). Esta arquitectura es también basada en capas aunque 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:

- ✚ **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.
- ✚ **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.
- ✚ **Capa de Acceso a Dato:** En esta capa estará presente el ORM (Mapeo de Objetos Relacionales) Doctrine, como Marco de Trabajo de persistencia para la comunicación con el servidor de datos mediante **PDO**¹⁹, también consta de un Persistidor de Configuración que es el encargado de comunicarse vía XML con los Ficheros de Configuración del sistema denominado FastResponse.
- ✚ **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.
- ✚ **Capa de Servicio:** En esta última capa se encuentran todos los subsistemas que prestan y consumen servicios entre sí. (29)

La combinación de todos estos estilos arquitectónicos permite el desarrollo en paralelo, incluso por capas, robustez mediante el encapsulamiento, rapidez y agilidad en el soporte, hace la solución muy escalable, ya que la idea es ir agregando componentes a medida que se desee generalizar el producto para otros clientes con características específicas que deseen hacerse de SIPAC sobre la misma base tecnológica.

Precisamente, la agrupación de funcionalidades en componentes, y la interacción entre estos generan estructuras o niveles de empaquetamiento:

¹⁹ **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.

- ✚ **Nivel Sistema.** El nivel de sistema es el TODO. Está compuesto por todos los subsistemas, y es la vista global del producto.
- ✚ **Nivel Subsistema.** Corresponde con la abstracción de las áreas de proceso del negocio, que posee objetos propios y operaciones asociadas a esos objetos. Tiene implicación en el diseño de instalación de la solución, está compuesto por módulos u otros subsistemas que a su vez están compuestos igualmente por módulos. La colaboración entre subsistemas será gestionada por el nivel más alto, nivel de sistema.
- ✚ **Nivel Módulo.** Este nivel surge para empaquetar 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.
- ✚ **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.

2.6. Patrones de diseños.

Los patrones de diseño especifican la estructura y el comportamiento de una sociedad de clases. Los Patrones de Diseño buscan 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.

2.6.1. Patrones GRASP.

GRASP: (General Responsibility Assignment Software Patterns) se encargan de asignar responsabilidades, las principales responsabilidades son conocer (atributos, relaciones con otros objetos) y hacer (tareas que debe cumplir cada objeto). (30)

Los principales patrones GRASP utilizados fueron:

✚ **Experto**

Indica cual es la clase que debe asumir una determinada responsabilidad (clase experta), teniendo en cuenta la información que contiene para cumplirla. Este se evidencia en la definición de las clases según las funcionalidades que realizan, como por ejemplo la clase **GestionarGruposModel** que es la encargada de realizar las funcionalidades de insertar, actualizar, eliminar y buscar grupos de usuarios, entre otras.

✚ **Creador**

El patrón Creador 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 **GestionargruposController**, la cual es la responsable de crear instancias de la clase **GestionarGruposModel** cada vez que desee utilizar funcionalidades de esta.

✚ Bajo Acoplamiento.

Este patrón resuelve el problema de cómo lograr baja dependencia, alta reutilización entre las clases y minimizar el impacto de los cambios. La siguiente figura ejemplifica como se empleó este patrón:

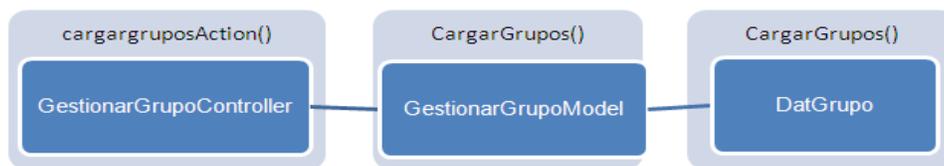


Figura 2.8: Patrón Creador evidenciado en la clase **GestionarGrupoController**.

En este caso la responsabilidad de crear objetos de **DatGrupo** la tiene la clase **GestionarGrupoModel** disminuyendo el número de recurrencias de la clase **GestionarGrupoController** a **DatGrupo**.

✚ Alta Cohesión.

Este patrón se aplica para agrupar las clases en dependencia de los requerimientos y los procesos que se necesitan informatizar, de modo que cada clase implementa las operaciones que se encuentran en su misma área funcional, de esta forma **GestionarGrupoModel** solamente se encarga de crear y modificar grupos de usuarios, así como otras operaciones que se realizan sobre esta área, limitándola a otras operaciones de otra naturaleza.

✚ Controlador.

Este patrón define quién responde ante determinados eventos. En este caso la clase controladora de cada componente, aparte de separar la lógica de negocios de la capa de presentación, delega en otras clases (los modelos de negocio y dominio) la responsabilidad de

efectuar operaciones complejas sobre el área, funcionando como agente de tráfico para las operaciones, donde recibe peticiones y devuelve respuestas, mientras que otras clases se encargan de gestionarlas realmente. **GestionarGrupoController** y **GestionarPermisosController** son ejemplos de cómo se aplica este patrón.

2.6.2. Patrones GoF.

GoF o Gang of Four, llamados así por los cuatro autores del libro Patrones de Diseño que recoge alrededor de 23 patrones de los más utilizados que están clasificados según su objetivo:

- **Creación:** Para la creación de instancias.
- **Estructura:** Para la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad.
- **Comportamiento:** Para la interacción y cooperación entre clases. (31)

Los más utilizados son:

✚ Fachada.

Simplificando el acceso a un conjunto de objetos relacionados mediante una interfaz de alto nivel que esconde la complejidad del sistema este patrón se utiliza para proporcionar un acceso sencillo a subsistemas muy complejos. La relación entre componentes y/o módulos mediante servicios consumidos y prestados, en este caso el módulo **Configuración**, mediante la clase **ConfiguracionServices**, actúa como fachada brindando una serie de servicios a utilizar por los componentes del módulo **Planeación**.

✚ 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.

2.7. Diseño de la solución.

2.7.1. Diseño de componentes.

El módulo **Configuración** en sí mismo es un macro componente de SIPAC, que a su vez está compuesto por los componentes Grupos, Permisos y Nomencladores. **Configuración** se encarga de poder establecer la parte final de mecanismo de compartimentación de la información que se decida planificar, cuestión para lo cual permite generar permisos de acceso a la información a nivel de grupos y usuarios, nomencladores globales o particulares, y por supuesto, los grupos de usuarios a utilizar a la hora de indentar a los involucrados. Estos datos son la base para que los componentes del módulo **Planeación** y **Reportes** puedan acceder y mostrar los planes, actividades, objetivos y demás según la regla principal: la información debe de ser vista y gestionada en el momento adecuado, por las personas adecuadas. A continuación una imagen que muestra el modelo de componentes de SIPAC, donde se destaca el componente Configuración:

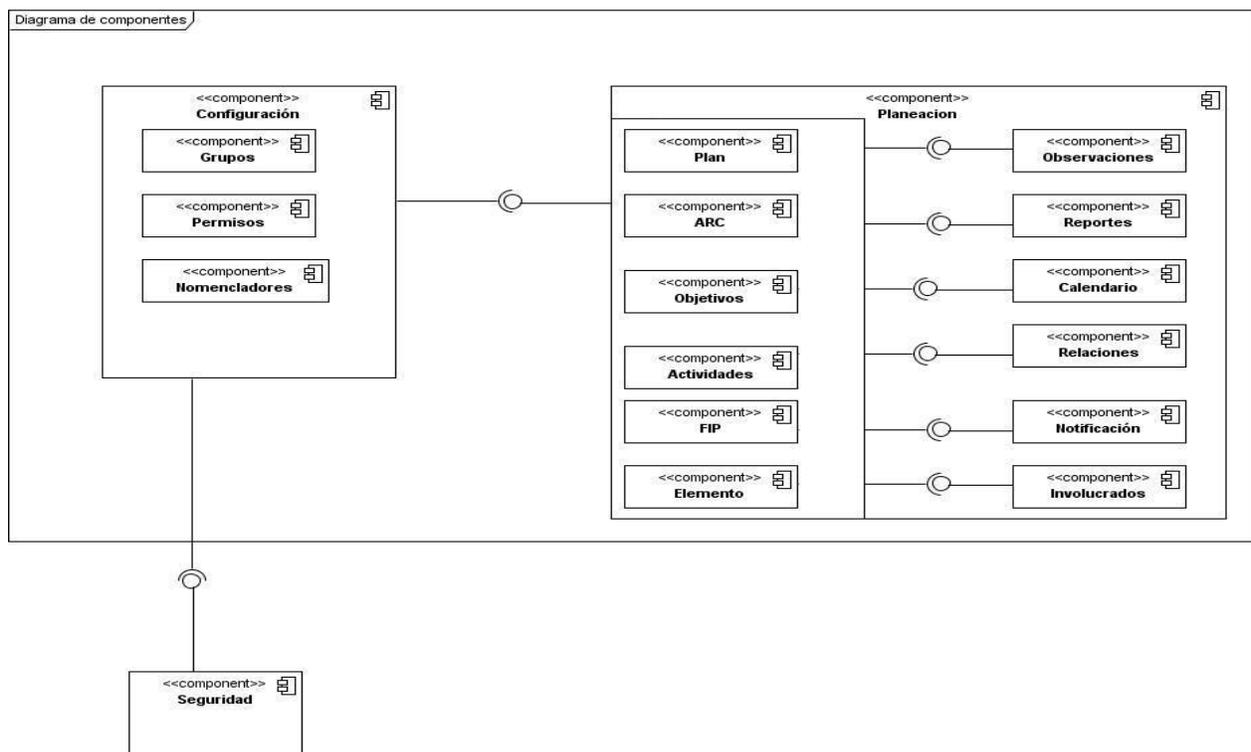


Figura 2.9: Diagrama de componentes SIPAC.

2.7.2. Servicios consumidos y brindados por el módulo Configuración.

Tabla 6: Servicios consumidos y brindados por el módulo Configuración.

Servicios que consume el módulo Configuración	
Modulo o Componente que brinda el servicio	Servicio
Seguridad	<ul style="list-style-type: none"> ✓ cargarDomainPDO: Devuelve la estructura pero de forma fragmentada, que se encuentra dentro del dominio del usuario, pero hasta el nivel de usuario. ✓ cargarDomainPDOCompleto: Devuelve todos los usuarios que se encuentran en las estructuras que están contenidas en un dominio. ✓ obtenerUsuariosCargoByArrayId: Devuelve los usuarios que están contenidos en determinado cargo. ✓ obtenerNombreUsersProfile: Devuelve el nombre de un usuario.
Servicios que brinda el módulo Configuración	
Servicio	
<ul style="list-style-type: none"> ✓ ObtenerTodosNomencladoresPorTipo: Devuelve los nomencladores según su tipo. ✓ ObtenerCategoriasHijas: Devuelve los nomencladores hijos dado el nomenclador padre, principalmente si son nomencladores de tipo Categoría de Actividad. ✓ ObtenerArbolDeGrupos: Devuelve los grupos existentes pero de forma anidada. ✓ ObtenerInvolucradosAdicionar: Devuelve los usuarios sobre los cuales se pueden adicionar planes, actividades, objetivos, etc. ✓ ObtenerGruposDondeEstaUsuario: Devuelve los grupos donde del cual forma parte determinado usuario. ✓ ObtenerUsuariosDeGrupo: Devuelve todos los usuarios contenidos en un grupo. ✓ actualizarGruposDPO: Actualiza los grupos de usuarios mediante la implementación del patrón Observador desde el modulo Seguridad. ✓ ObtenerSubordinados: Devuelve los usuarios subordinados de un determinado usuario. ✓ ObtenerPermisosLeer: Devuelve los usuarios sobre los cuales se tiene permiso para leer su planificación. ✓ BuscarInvolucrados: Devuelve los usuarios involucrados en un determinado elemento. 	

2.7.3. Diseño de clases.

El Diagrama de Clase es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de

herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones.

2.7.4. Diagramas de clases del diseño.

Seguidamente se presentan las propiedades y procedimientos relacionados con cada una de las clases de diseño para así facilitar la implementación de la solución propuesta:

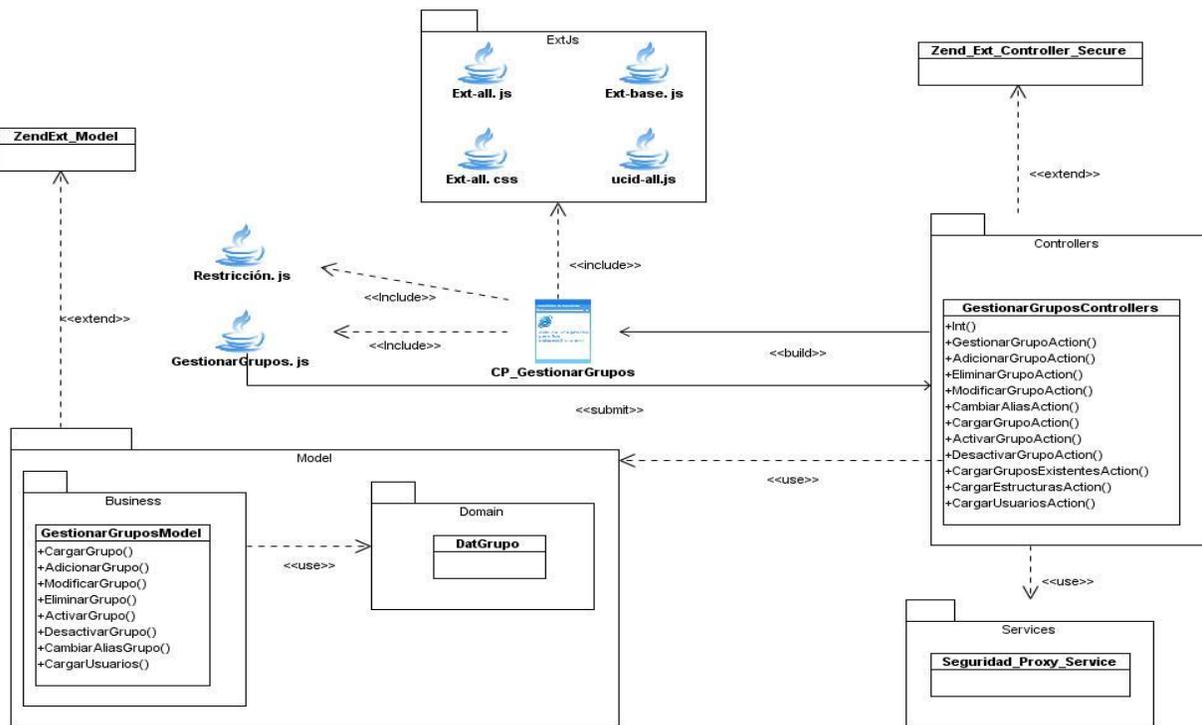


Figura 2.10: Clase del diseño perteneciente a Gestionar Grupos de usuarios.

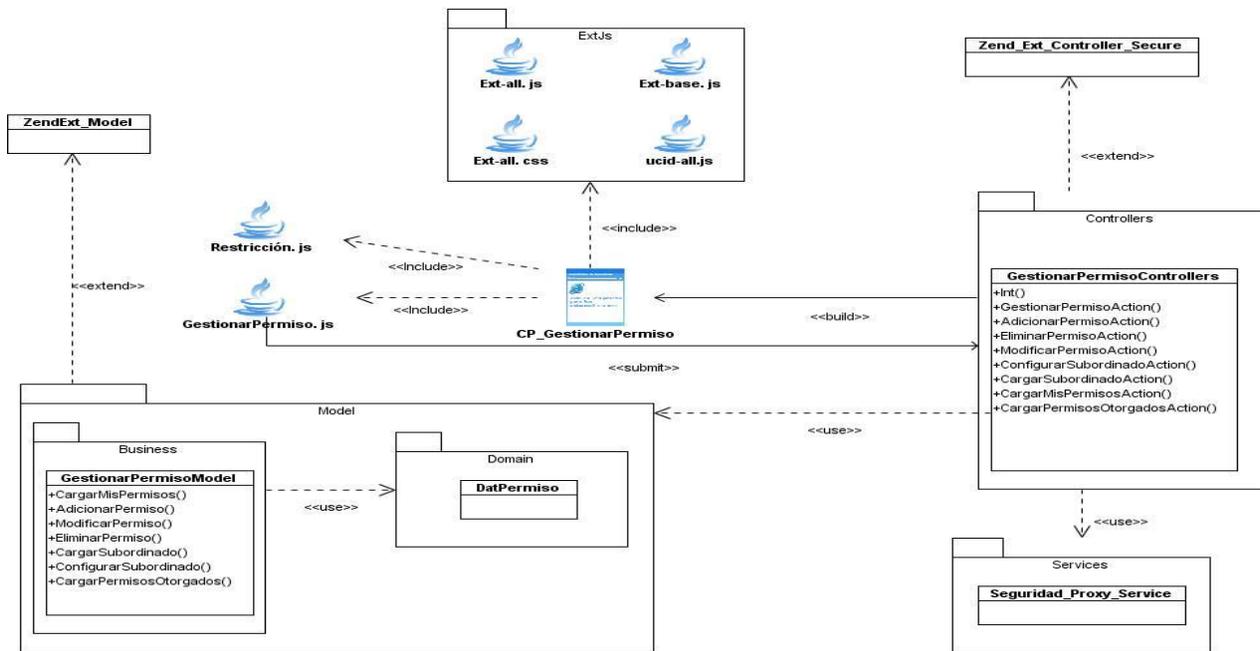


Figura 2.11: Clase del diseño perteneciente a Gestionar Permisos.

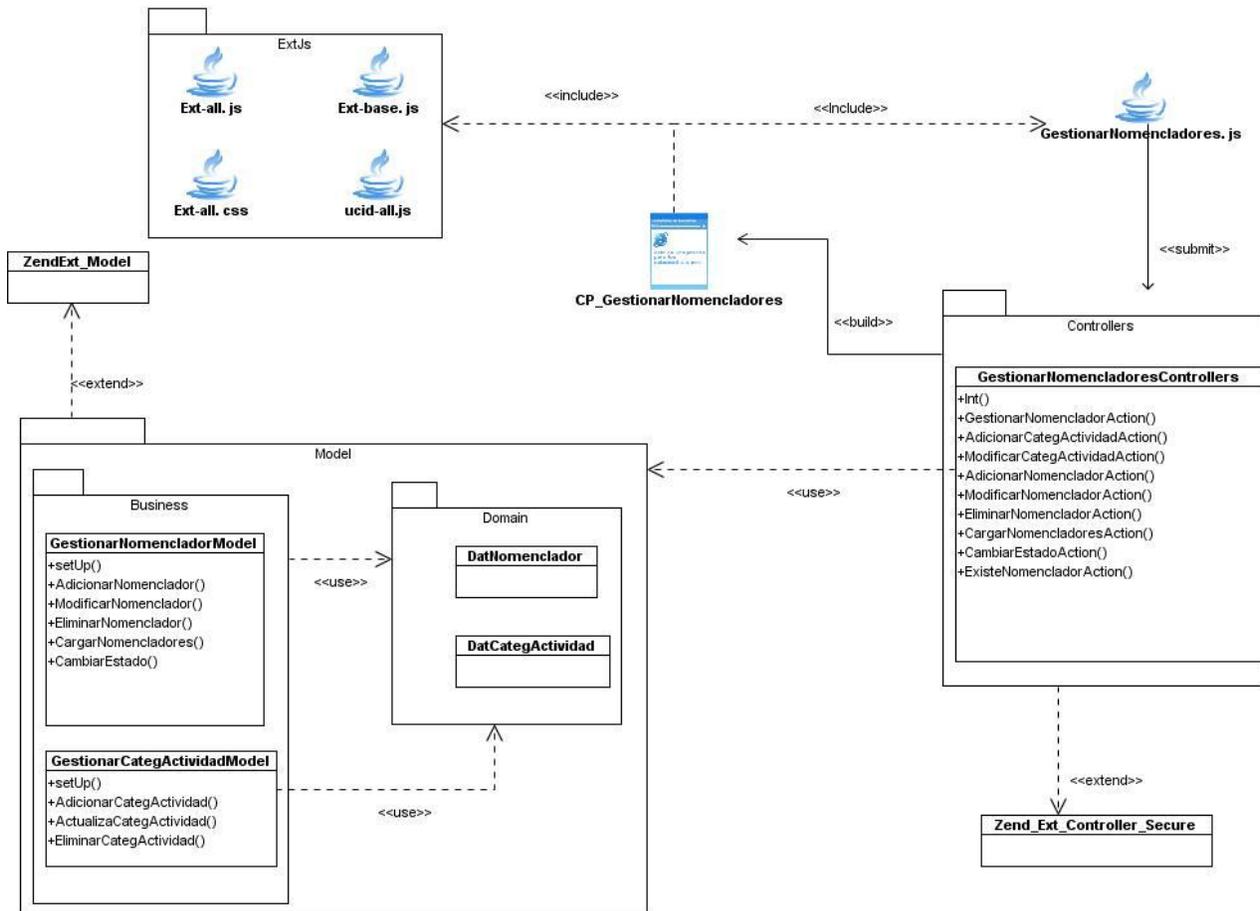


Figura 2.12: Clase del diseño perteneciente a Nomencladores.

Tabla 7: Descripción del diseño de clases asociadas al componente Nomencladores.

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_GestionarNomencladores	Página encargada de representar, a través de los archivos de extensión <i>js</i> que debe incluir, la información relacionada con la definición de los nomencladores de Tipo de Actividad y Categoría de la Actividad.
gestionarNomencladores.js	Script encargado de generar de forma dinámica a través del DOM y utilizando la librería ExtJS, los componentes de la vista necesarios para seleccionar los distintos nomencladores y realizar operaciones sobre ellos.
SP_GestionarNomencladoresController	Contiene la clase controladora encargada de ejecutar las funcionalidades según las peticiones de los usuarios.
ZendExt_Controller_Secure	Clase encargada de gestionar acciones personalizadas y está integrada a la seguridad.
NomNomencladorModel	Clase del Negocio encargada de manejar los datos persistentes dentro del componente, contiene las funcionalidades referentes a la gestión del nomenclador Tipo de Actividad.
NomCategActividadModel	Clase del Negocio encargada de manejar los datos persistentes dentro del componente, contiene las funcionalidades referentes a la gestión del nomenclador Categoría de la Actividad.
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 cada una de 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.

2.7.5. Modelos de datos.

El modelo de datos no es más que un lenguaje utilizado para la descripción de una Base de Datos. Por lo general, permite describir las estructuras de datos de la base (el tipo de los datos que incluye la base y la forma en que se relacionan), las restricciones de integridad (las

condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada) y las operaciones de manipulación de los datos (agregado, borrado, modificación y recuperación de los datos de la base). (32)

Partiendo del modelo conceptual, el modelo de datos fue diseñado bajo una visión de escalabilidad, resumiendo los conceptos y definiendo las relaciones, teniendo en cuenta la mayor visión del alcance del módulo. En este diseño se tienen en cuenta una serie de aspectos que agilizan la búsqueda de información, pues el rendimiento es uno de los temas más importantes de perfeccionar en las aplicaciones web, y más aún teniendo en cuenta la situación económica de Cuba y el llamado al uso racional de los recursos.

- ✚ **Tratamiento de la información como nodos de un árbol:** Permite hacer recorridos sobre los registros de la tabla, haciendo más eficiente las búsquedas de registros. La tabla **nom_nomenclador** almacena los registros relacionados en forma arbórea, donde un registro puede tener subregistros e ir conformando un árbol de descendencia. De esta manera, para cada registro existe el campo **ordender** y **ordenizq**, que almacenan valores numéricos generados por disparadores o triggers²⁰ que realizan cálculos a partir del recorrido entre-orden tomando a cada registro como un nodo. Se puede acceder de forma rápida a los subregistros de determinado registro, o a su registro padre en caso que tenga. De esta manera, se reduce considerablemente el recorrido sobre todas las tuplas de la tabla para buscar las relaciones de dependencias entre determinados nomencladores. Este mecanismo es utilizado también en sistemas que incluyen inteligencia artificial como los almacenes de datos.
- ✚ **Columnas indexadas:** Permite localizar los registros rápidamente mediante el acceso directo y rápido haciendo más eficiente las búsquedas en PostgreSQL. Sin índice, se debe recorrer secuencialmente toda la tabla para encontrar un registro. Indexando la columna se genera un puntero, por el cual se inician las búsquedas sobre esta tabla, eliminando así las iteraciones sobre registros innecesariamente, disminuyendo el tiempo de respuestas del servidor de datos y aumentando el rendimiento del sistema. A continuación la relación de columnas indexadas por tablas:

²⁰ **Disparador o trigger:** Procedimiento o acción definida en una tabla de la base de datos y ejecutada automáticamente por una función programada. Esta acción se activará, según se defina, cuando se realicen operaciones como INSERTAR, MODIFICAR, o ELIMINAR en dicha tabla.

Tabla 8: Relación de columnas indexadas por tablas.

Tablas	Columnas indexadas
dat_permiso	idpermiso idppara idpsobre idusuariocrea
dat_grupo	idgrupo dengrupo idusuariocrea
nom_nomenclador	idnomenclador dennomenclador idpadre idusuariocrea

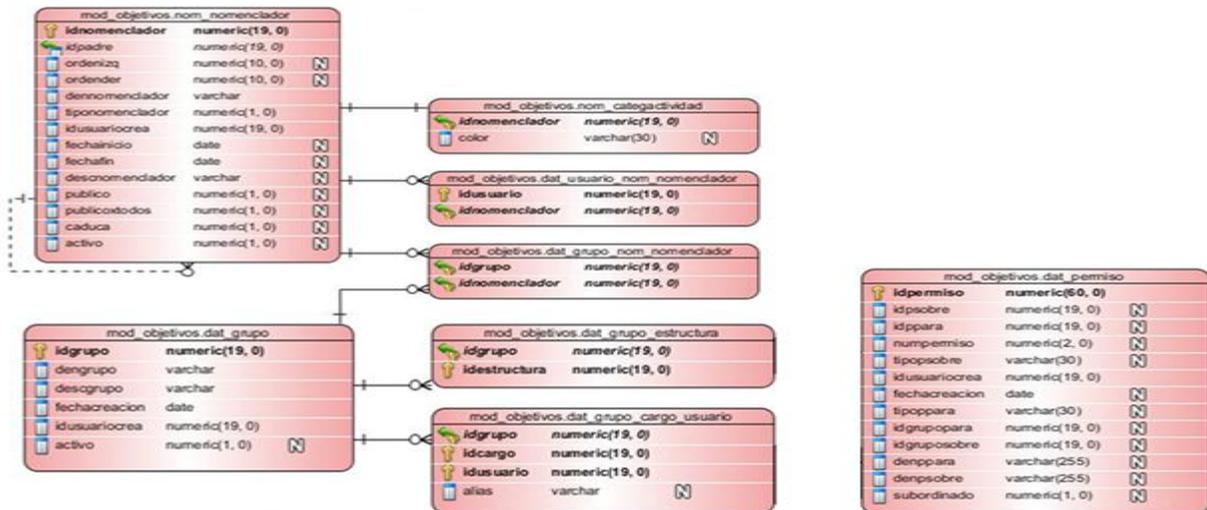


Figura 2.13: Modelo de Datos correspondiente al módulo Configuración.

2.1.1. Otros modelos aplicados.

Además de las fórmulas utilizadas para el trabajo con la información persistida en forma arbórea para acceder más rápido a la información y que pueden considerarse como parte del modelo matemático que plantea el recorrido entre-orden de los árboles. Se ha utilizado otro modelo matemático para determinar los niveles de prioridad y jerarquía a la hora de gestionar los permisos de acceso a la información planificada.

El negocio plantea la necesidad de asignar permisos de leer, adicionar, modificar y eliminar la información planificada entre usuarios y grupos de usuarios. Para los usuarios que reciben un determinado permiso se debe de comprobar si ya ese permiso le ha sido asignado mediante un grupo de usuarios donde él esté contenido, o directamente. En cualquiera de estos casos se debe de proceder a comparar los permisos existentes y hacer una unión de los mismos.

En muchos casos esta asignación-eliminación de permisos se puede efectuar por varios jefes hacia un mismo nivel inferior, lo que muchas veces genera conflicto a la hora de decidir con cual permiso quedarse. El siguiente esquema refleja lo explicado anteriormente:



Figura 2.14: Reasignación de los mismos permisos por diferentes jefes al mismo subordinado.

Para cualquiera de estos escenarios, la mejor forma de persistir estos permisos es mediante valores numéricos donde:

n Representa el numero del permiso (leer = **0**, adicionar = **1**, modificar = **2** y eliminar = **3**).

2^n Representa el valor de potencia con base 2 para cada permiso asignado.

Donde **Permiso** = $2^0 + 2^1 + \dots + 2^n$ según el permiso asignado.

De esta manera se garantiza un único valor numérico para cada permiso o combinación de permisos, en este caso:

1= leer.

2= adicionar.

4= modificar.

8= eliminar.

Ejemplo:

3 → (1+2)= leer y adicionar.

5 → (4+1)= modificar y leer.

7 → (4+2+1)= modificar y adicionar (y leer, pues para modificar primero se debe de leer).

9 → (8+1)= eliminar y leer.

11 → (8+2+1)= eliminar y adicionar (y leer, pues para eliminar primero se debe de leer).

13 → (8+4+1)= eliminar y modificar (y leer, pues para eliminar o modificar primero se lee).

15 → (8+4+2+1)= eliminar, modificar, adicionar y leer.

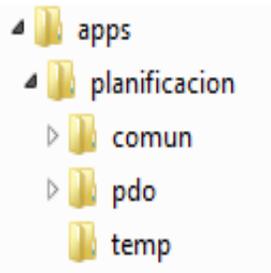
Este modelo matemático es muy parecido a la filosofía de asignación de permisos en Linux, pero con una leve diferencia ya que en Linux están definidos todos los tipos de permisos, mientras

que la implementación de este modelo matemático para el módulo Configuración permite generar dinámicamente nuevos valores según se vayan incorporando nuevos permisos o combinaciones de los mismos a medida que el negocio se pueda ir complejizando. Esta filosofía también es utilizada en el eGroupware.

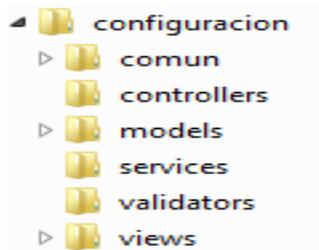
2.8. Estructura Física del Sistema.

La estructura física del sistema está definida a partir de las decisiones arquitectónicas tomadas por el Departamento de Tecnologías del proyecto ERP-Cuba y establecida por el departamento de soluciones tecnológicas de CEIGE el equipo de arquitectura, fundamentalmente diseñada para satisfacer los estilos arquitectónicos y niveles de empaquetamiento adoptados para la estructura del sistema. Dentro de esta se encuentran ubicados cada uno de los subsistemas implementados dentro de la aplicación, de manera que facilite la organización y claridad durante el desarrollo. A continuación se presenta la especificación la estructura del componente Configuración perteneciente al Subsistema Planificación.

La carpeta denominada **apps** es la que contiene la lógica del negocio. En ella se almacenan los controladores y el modelo de cada uno de los componentes correspondientes a los subsistemas. En la figura se muestra el contenido del subsistema en cuestión.



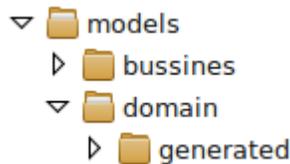
- ✚ **comun:** Comprende la configuración específica para el subsistema, contiene la carpeta **recursos** y dentro de esta una denominada **xml**. Esta última incluye al fichero **ioc**, donde se publican los servicios que brindan cada uno de los componentes del Subsistema de Planificación mediante un lenguaje de descripción de servicios web (WSDL).
- ✚ **pdo:** Contiene al Subsistema de Planificación que incluye el componente Configuración, el cual está estructurado por un conjunto de paquetes que se especifican a continuación:



- ✚ **comun:** Contiene la configuración utilizada por el componente, incluye la carpeta *recursos* y dentro de esta una denominada **xml** que contiene los ficheros: **validation** y **exception**.

El fichero **validation** chequea las precondiciones antes de ejecutar una determinada función en el servidor según el tipo de parámetros, la acción y el usuario que la realice. Mientras que en **exception** se define el tipo, idioma y la descripción del mensaje que va a ser mostrado cuando se lance una excepción en el servidor.

- ✚ **controllers:** En este se van a encontrar las clases controladoras encargadas de gestionar las funcionalidades de sistema.
- ✚ **models:** Este está estructurado de la siguiente manera:

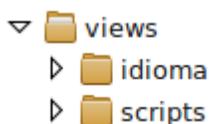


Este paquete contiene dos paquetes, los cuales a su vez agrupan clases y otros paquetes. El paquete **bussines** contendrá las clases necesarias para acceder a los datos que persisten en la base de datos.

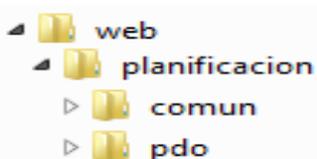
El paquete **domain** debe contener las clases generadas por el ORM Doctrine a partir de cada una de las tablas existentes en la base de datos. Cada una de estas clases heredará de clases bases que contienen los atributos y dependencia entre las tablas. Estas últimas son igualmente generada por el Doctrine las cuales se ubican dentro del paquete llamado **generated**.

- ✚ **services:** Este paquete incluye todas las clases y funcionalidades de los servicios que va a ofrecer el componente.
- ✚ **validators:** Incluye las clases y funcionalidades correspondientes a las validaciones que realiza el servidor antes de ejecutar una determinada función. Estas son llamadas en el xml **validation** perteneciente a los recursos del componente en cuestión.

- ✚ **views:** Contiene los paquetes idioma y scripts, que se encargan de contener el idioma en que se va a mostrar la aplicación y las páginas clientes respectivamente.



Al mismo nivel que la carpeta **apps** existe la carpeta **web**, esta es la que contiene las vistas de todos los subsistemas y componentes de la aplicación.

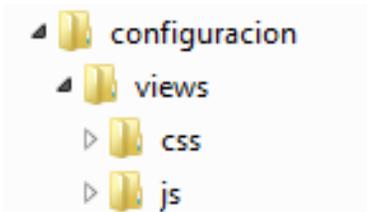


Esta a su vez contiene un fichero llamado *index.php* que incluye la dirección del archivo de configuración y a través de este inicializa la aplicación para que se carguen en la misma un conjunto de componentes necesarios para su funcionamiento. Su código permanece igual para todos los componentes.

- ✚ **pdo:** Contiene todas las vistas del Subsistema de Planificación, incluye el componente Configuración, que igualmente incorpora el fichero **index.php**. También contiene el paquete **views** que se especifica a continuación:



- ✚ **views:** Esta contiene los **css** y **js** necesarios para visualizar todo en contenido de la presentación del componente Configuración.



- ✚ **css:** Incluye las clases necesarias para estructurar gráficamente el componente, separando de esta forma el estilo del contenido.
- ✚ **js:** Comprende los ficheros JavaScript necesarios para que el usuario interactúe con el sistema y obtenga los resultados esperados.

2.9. Implementación.

En el flujo de trabajo de Implementación se comienza a trabajar a partir del resultado obtenido en el diseño y se implementa el sistema en términos de componentes, es decir, ejecutables, ficheros, códigos de fuente y similares, necesarios para la implantación y despliegue del sistema. El objetivo fundamental es desarrollar la arquitectura y el sistema como un todo. (33)

2.10. Diagrama de Despliegue.

Posteriormente se muestran las relaciones físicas entre los componentes Hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). Existen dos posibles escenarios para el despliegue de la aplicación:

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

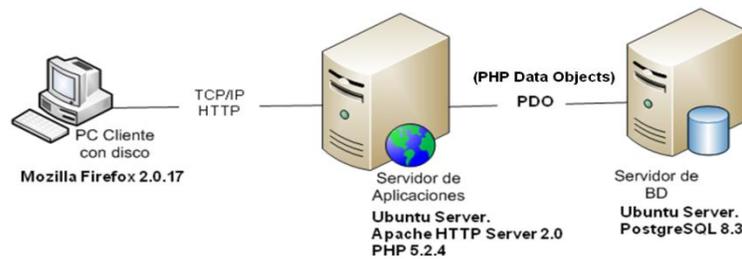


Figura 2.15: Diagrama de despliegue de escenario para PC cliente con disco.

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

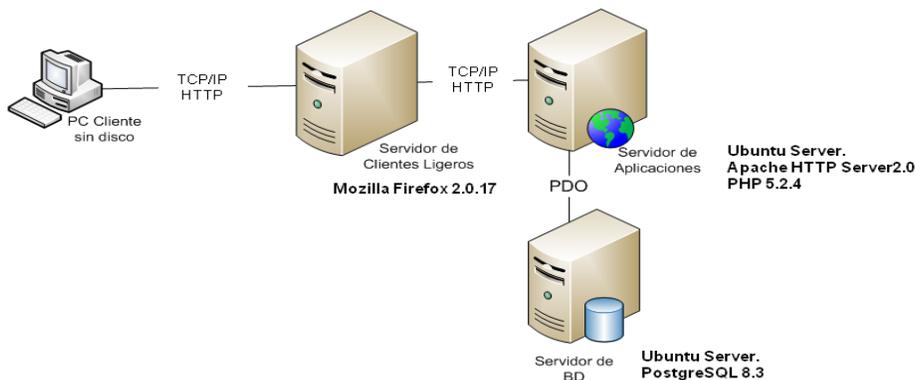


Figura 2.16: Diagrama de despliegue de escenario para PC cliente sin disco.

2.11. Conclusiones del capítulo.

El presente capítulo fue decisivo en el diseño e implementación de la solución propuesta para automatizar el proceso de organización teniendo en cuenta elementos primordiales 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 soporte que definirán la estructura física del sistema que se pretende desarrollar como propuesta de solución. Además se plasmó la arquitectura de la solución que se propone fundamentando así los patrones de diseño empleados los cuáles permitieron especificar la estructura y comportamiento de las clases, garantizando presentar una solución que atiende los niveles de reutilización y mantenimiento. De igual forma se analizaron los modelos matemáticos utilizados en la persistencia de los nomencladores y asignación de permisos. Finalmente se precisó los estándares de código para su implementación, la estructura física del sistema y los componentes necesarios para su despliegue.

Capítulo 3. Validación de la Solución Propuesta.

Introducción del Capítulo.

Este capítulo muestran las métricas aplicadas al diseño propuesto, implementando en términos de componentes las clases y subsistemas obtenidos en el capítulo anterior. De igual forma se abarcará todo lo concerniente a la realización de las pruebas de aplicación, en aras de garantizar la calidad de la misma.

3.1. Validación del Diseño Propuesto.

Las Métricas de diseño permiten medir de forma cuantitativa la calidad de los atributos internos del software. Esto posibilita al ingeniero evaluar la calidad durante el desarrollo del sistema. Son varios los puntos de vista relacionados con la calidad del software. Las métricas de diseño a nivel de componentes se concentran en las características internas de los componentes del software con medidas que pueden ayudar al desarrollador a juzgar la calidad de un diseño a nivel de componente. Estas se centran en cuantificar tanto la complejidad, como la funcionalidad y eficiencia inmersa en el desarrollo de software. Inclina sus objetivos a mejorar la comprensión de la calidad del producto, a estimar la efectividad del proceso y mejorar la calidad del trabajo. 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 empleadas están diseñadas para evaluar los siguientes atributos de calidad:

- **Responsabilidad.** Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
- **Complejidad de implementación.** Consiste en el grado de dificultad que tiene implementado un diseño de clases determinado.
- **Reutilización.** Consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.
- **Acoplamiento.** Consiste en el grado de dependencia o interconexión de una clase o estructura de clase, con otras, está muy ligada a la característica de Reutilización.

- **Complejidad del mantenimiento.** Consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costes y la planificación del proyecto.
- **Cantidad de pruebas.** Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (Unidad) del producto (componente, módulo, clase, conjunto de clases, etc.) diseñado. (34)

3.1.1. Métrica TOC: Tamaño operacional de clase.

Está dado por el número de métodos asignados a una clase y evalúa los siguientes atributos de calidad:

- **Responsabilidad:** Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
- **Complejidad de implementación:** Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
- **Reutilización:** Un aumento del TOC implica una disminución del grado de reutilización de la clase.

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

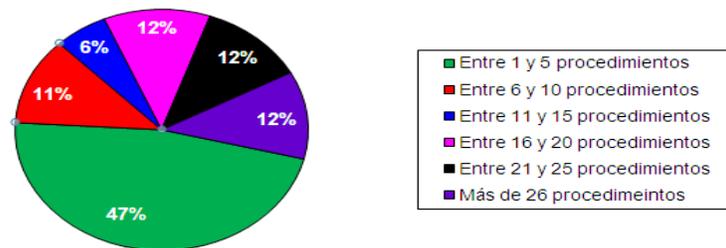


Figura 3.1: Representación en % de la cantidad de clase y el número procedimientos que contienen.

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



Figura 3.2: Representación del valor en % del atributo Responsabilidad.

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

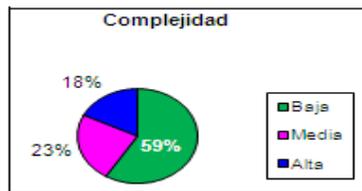


Figura 3.3: Representación del valor en % del atributo Complejidad de Implementación.

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



Figura 3.4: Representación del valor en % del atributo Reutilización.

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

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica TOC, teniendo en cuenta que el 64% de las clases posee un número de funcionalidades inferior a la media de procedimientos por clases (cuyo valor asciende a 15) con altos niveles de porcentajes en los atributos de complejidad de implementación y responsabilidad, se puede concluir que el diseño del módulo implementado tiene una calidad aceptable. Es apreciable, destacar que en el 41% de las clases que intervienen en el diseño tiene un nivel bajo de reutilización.

3.1.3. Métrica RC: Relaciones entre clases.

Está dado por el número de relaciones de uso de una clase con otra y evalúa los siguientes atributos de calidad:

Capítulo 3: Validación de la Solución Propuesta

- **Acoplamiento:** Un aumento del RC implica un aumento del Acoplamiento de la clase.
- **Complejidad de mantenimiento:** Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
- **Reutilización:** Un aumento del RC implica una disminución en el grado de reutilización de la clase.
- **Cantidad de pruebas:** Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.

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

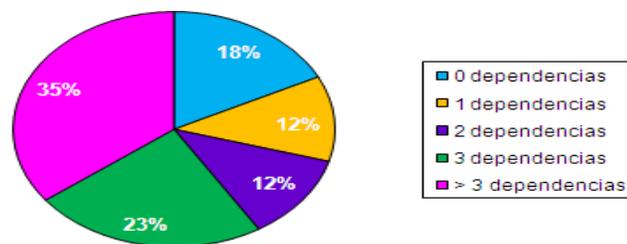


Figura 3.5: Representación en % de las asociaciones de uso por cantidad de dependencias de clases.

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

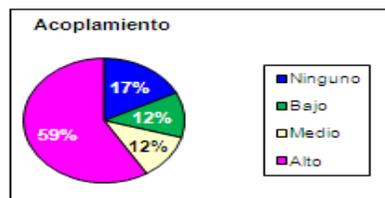


Figura 3.6: Representación del valor en % del atributo Acoplamiento.

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

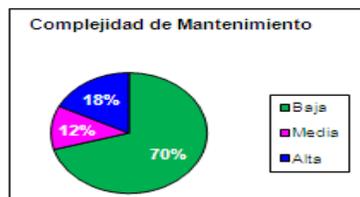


Figura 3.7: Representación del valor en % del atributo Complejidad de Mantenimiento.

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

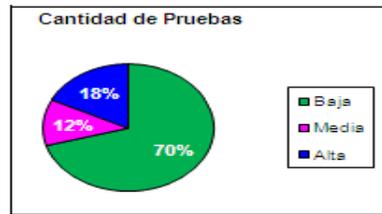


Figura 3.8: Representación del valor en % del atributo Cantidad de Pruebas.

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

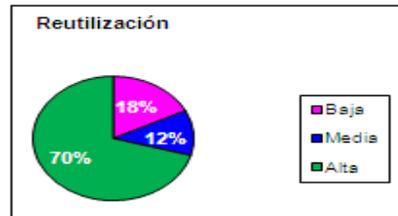


Figura 3.9: Representación del valor en % del atributo Reutilización.

3.1.4. 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 7.5 y que el 42% de dichas clases contienen menos de 3 asociaciones se puede afirmar que la solución que se propone presenta un bajo nivel de colaboraciones por clases. Por consiguiente, si el 59% de las clases presentan un alto nivel de acoplamiento, menor será la complejidad de mantenimiento y la cantidad de comprobación que se produzcan cuando se efectúen modificaciones. En consecuencia el grado de reusabilidad será mayor.

3.2. Pruebas de aplicación.

Las pruebas constituyen un instrumento adecuado para determinar el status de la calidad de un producto software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante Técnicas de prueba. Su objetivo principal es asegurar que el software cumpla con las especificaciones requeridas y eliminar los posibles defectos que este pudiera tener.

Es importante considerar que las pruebas de software no garantizan que un sistema esté libre de errores, sino que se detecten la mayor cantidad de defectos posibles para su debida corrección.

3.2.1. 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:

- ✚ **Pruebas de Unidad:** Prueba individual a las unidades separadas de un sistema de software.
- ✚ **Pruebas de Integración:** Los componentes individuales son combinados con otros componentes para asegurar que la comunicación, enlaces y los datos compartidos ocurran apropiadamente.
- ✚ **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.
- ✚ **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.

3.2.2. 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**²¹ para detectar diferentes tipos de errores.

- ✚ **Pruebas de Caja Blanca:** Consiste en realizar pruebas para verificar que líneas específicas de código funcionan tal como está definido y se comprueban los componentes internos. Al mismo tiempo prueba 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.
- ✚ **Pruebas de Caja Negra:** Se verifica que el ítem que se está probando, cuando se dan las entradas apropiadas produce los resultados esperados es decir se comprueban las funcionalidades sin tener en cuenta la estructura interna. Se refiere a las pruebas que

²¹**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 llevan a cabo sobre la interfaz del software los cuales demuestran a través de los casos de prueba que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

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. (35)

3.2.3. Pruebas realizadas a los componentes:

El módulo Configuración fue sometido a evaluación pasando por los niveles de unidad, integración, sistema y aceptación. Las pruebas de unidad y de sistema les fueron realizadas por un equipo perteneciente al Centro de Aseguramiento de la Calidad del UCID, y donde también tuvo participación una especialista de Calisoft. Se detectaron una serie de no conformidades las cuales se detallan más adelante. En cuanto a la prueba de integración le fue realizada en los laboratorios de Calisoft, sin arrojar no conformidades.

Esta evaluación fue ejecutada mediante 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. A continuación se describe uno de los casos de prueba utilizados durante la prueba de unidad y sistema.

3.2.4. Casos de pruebas de caja negra generados al aplicar dicha técnica.

Requisito a probar: **Adicionar Permiso.**

Tabla 9: Caso de Prueba Adicionar Permiso.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1: Adicionar permiso.	Se adiciona un permiso, con los atributos: Tipo de permiso Permiso para Permiso sobre	EP 1.1: Adicionar un nuevo permiso introduciendo datos válidos.	<ul style="list-style-type: none"> • Se selecciona la pestaña Permisos otorgados por mí. • Se presiona el botón Adicionar. • Se insertan todos los datos. • Se presiona el botón Aceptar. • Se presiona el botón Aceptar del mensaje de información.
		EP 1.2: Adicionar un nuevo permiso dejando campos requeridos en blanco.	<ul style="list-style-type: none"> • Se selecciona la pestaña Permisos otorgados por mí. • Se presiona el botón Adicionar. • Se introducen los datos dejando campos requeridos en blanco. • No se activa el botón de Aceptar. • Solo se puede realizar la operación de Cancelar.
		EP 1.3: Cancelar	<ul style="list-style-type: none"> • Se selecciona la pestaña Permisos otorgados por mí. • Se presiona el botón Adicionar. • Se introducen o no los datos en

Capítulo 3: Validación de la Solución Propuesta

			el formulario. • Se presiona el botón Cancelar .
--	--	--	--

Tabla 10: Descripción de las variables.

No	Nombre de campo	Tipo	Válido	Inválido
1	Tipo de permiso.	Campo de selección múltiple.	Leer, Adicionar, Modificar, Eliminar.	NA.
2	Permiso para.	Campo de selección múltiple.	Permite la selección múltiple de los elementos que muestra el sistema. No permite la entrada de datos.	NA.
3	Permiso sobre.	Campo de selección múltiple.	Permite la selección múltiple de los elementos que muestra el sistema. No permite la entrada de datos.	NA.

En estos momentos a SIPAC, con el módulo Configuración, se les está realizando pruebas de aceptación en el Ministerio de las Fuerzas Armadas Revolucionarias, pruebas que se utilizarán para hacer lo mismo en otros ministerios del país. También en el MINFAR se ejecutaron otras pruebas un poco más prácticas y específicas con el fin de validar la idea a defender planteada en la Introducción del presente trabajo.

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 desde el punto de vista interno y funcional de los componentes. 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:

- ✚ **En la primera iteración** se detectaron 20 No Conformidades (NC), de las cuales 4 fueron por error de aplicación, 6 en la documentación y 10 que no proceden.
- ✚ **En la segunda iteración** se detectaron 6 NC, entre ellas 1 por error de aplicación y 5 en la documentación.

✚ **En la tercera iteración se detectaron 2 NC en la documentación.**



Figura 3.10: No conformidades por iteración detectadas durante las pruebas funcionales.

Todas las pruebas realizadas hasta el momento y las métricas de calidad aplicadas al diseño del componente han arrojado resultados satisfactorios y se puede afirmar que el sistema cumple con los estándares de calidad de diseño requeridos y que los requisitos funcionales han sido correctamente implementados, pero, ¿realmente se puede afirmar que el resultado final valida la idea a defender planteada en la Introducción del presente trabajo? Para concretar esta interrogante se decidió simular el proceso de organización con los mecanismos anteriores y con la solución propuesta. Se usaron algunas variables para comparar los resultados obtenidos, las cuales se detallan a continuación:

Tabla 11: Variables con sus dimensiones.

Variable	Dimensiones	Unidad de medida
Compartimentación	Acceso a información no autorizada	Unidades(u)
	Planificación conjunta	Unidades(u)
	Información con daños de integridad	Unidades(u)
Entendimiento común	Información planificada no válida	Unidades(u)

El escenario seleccionado fue la configuración del proceso organizativo mediante la gestión de grupos, permisos y nomencladores, y el proceso de elaboración de la planificación. Para esto se simuló dos niveles de planificación, donde hubo un planificador representando el nivel superior (nivel uno), y dos planificadores simulando el nivel inmediato inferior (nivel dos). La

Capítulo 3: Validación de la Solución Propuesta

configuración normalmente se realiza una sola vez, y raramente es modificada a menos que por cuestiones estratégicas así se decida, mientras que el proceso de elaboración de la planificación anual tiene determinado periodo al año, en la primera quincena del mes de Mayo.

✚ Compartimentación

A partir de la configuración realizada con los mecanismos existentes antes de la obtención del modulo Configuración, se planificaron 12 objetivos y 12 actividades por cada usuario en cada nivel, sumando un total de 36 objetivos y 36 actividades. Con las funcionalidades del módulo Planeación, los planificadores del nivel inferior efectuaron un filtro para obtener las actividades y objetivos provenientes del nivel superior y las elaboradas por ellos, obteniendo como resultado una lista de elementos con 9 objetivos y 10 actividades del nivel superior a la cual no tenían derecho o acceso de visualizar. De estos elementos visualizados fueron capaces de modificar 9 objetivos y 10 actividades, y de eliminar la misma cantidad. De un total de 24 objetivos y 24 actividades, ninguna respondía a la planificación conjunta entre los dos planificadores de dicho nivel.

Utilizando las bondades del modulo Configuración, se planificaron 12 objetivos y 12 actividades por cada usuario en cada nivel, sumando un total de 36 objetivos y 36 actividades. Los planificadores del nivel inferior efectuaron un filtro para obtener las actividades y objetivos provenientes del nivel superior, obteniendo como resultado 0 actividades y 0 objetivos. El planificador del nivel superior gestionó permisos de lectura sobre todos sus elementos para ambos planificadores del nivel inferior, los cuales volvieron a efectuar el mismo procedimiento de filtrado obteniendo como resultado la visualización de 12 objetivos y 12 actividades. De 24 elementos visualizados no pudieron modificar el contenido de ninguno, hasta que se les otorgo permiso de modificar y eliminar sobre el nivel superior, pudiendo así modificar los 24 elementos y eliminar la misma cantidad. Seguidamente, el planificador del nivel superior otorgo permisos de adición para uno de los planificadores del nivel inferior sobre el restante de ese mismo nivel, pudiendo elaborar así 4 actividades y 4 objetivos de realización conjunta entre ellos.

A continuación se muestra una grafica con los datos representados:

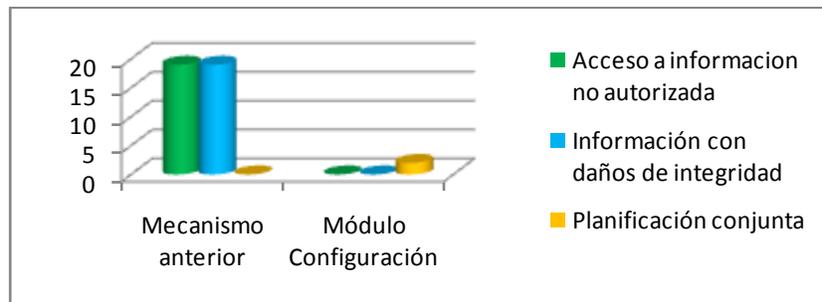


Figura 3.11: Estimación de la compartimentación y sus dimensiones.

Entendimiento Común

La medición de esta variable se apoya en los elementos planificados obtenidos en la medición de la variable **Compartimentación**. Con los mecanismos anteriores, el nivel inferior tuvo que gestionar sus propios nomencladores. De las 24 actividades planificadas en el nivel inferior, 17 fueron enviadas al nivel superior para formar parte del plan de dicho nivel, detectándose un total de 17 actividades no válidas pues no coincidía la clasificación del **tipo de actividad** y **categoría** con las que manejaba el nivel superior, teniendo que rechazar estos elementos y retornarlos al nivel inferior para su reelaboración.

Por otra parte, mediante el módulo Configuración se hicieron globales los nomencladores gestionados en el nivel superior, dando permiso de utilización de los mismos a los planificadores del nivel inferior, los cuales utilizaron estos datos en la reelaboración de las 17 actividades detectadas como rechazadas o no válidas. Una vez reelaboradas, se enviaron nuevamente al nivel superior, donde se procedió a revisarlas nuevamente y no se rechazó ninguna.

A continuación se muestra una grafica con los datos representados:

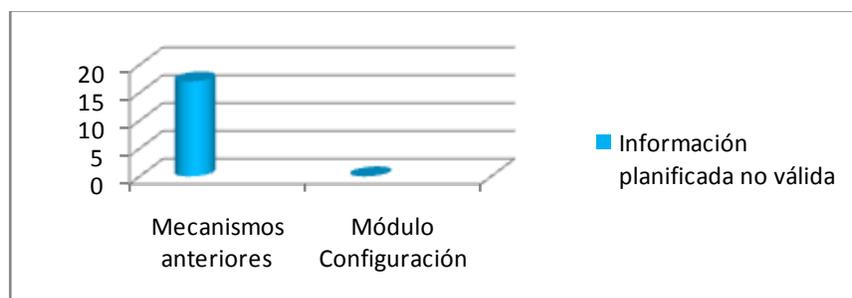


Figura 3.12: Estimación de la variable Entendimiento Común.

3.3. Conclusiones del Capitulo

En este capítulo Se realizó la evaluación del diseño utilizando las métricas Tamaño operacional de clases (TOC) y Relaciones entre clases (RC) obteniendo como resultado una calidad aceptable del diseño realizado teniendo en cuenta la valoración de la complejidad de la implementación de la solución propuesta en la presente investigación. Además se efectuaron pruebas de caja negra al componente Configuración empleando específicamente las pruebas de partición de equivalencia lo que posibilito detectar tras 3 iteraciones de realización de pruebas funcionales un total de 28 no conformidades las cuales fueron resueltas permitiendo cumplir con los requerimientos capturados. También se realizaron pruebas de aplicación para validar la idea defender del presente trabajo

Conclusiones Generales

El presente trabajo arrojó como resultado el módulo encargado de brindar funcionalidades para informatizar la personalización de los conceptos estructurales bases de la modelación de la entidad, establecer responsabilidades y atribuciones entre los niveles de la planificación y generalizar, re-implementar y adaptar las clasificaciones a utilizar en el proceso de concepción y ejecución de la planificación. Este resultado esta dado por la necesidad de las entidades cubanas de contar con una herramienta informática capaz de permitir el registro, control y seguimiento de las actividades y objetivos de acuerdo a lo plasmado en la Instrucción No1, lo que condujo a realizar un estudio sobre el modelo de planificación cubano y los procesos que incluye, su origen y evolución, las herramientas informáticas existentes utilizadas en su implementación, así como las desventajas de las mismas. Este estudio fue clave para la obtención del alcance de la solución propuesta, sus principales características tecnológicas y los escenarios a tener en cuenta, partiendo del diseño hasta la implementación y prueba de la misma, permitiendo alcanzar satisfactoriamente el cumplimiento de cada uno de los objetivos específicos, y por ende, el objetivo general. De esta manera, se pudo lograr que el Sistema para la Planificación de Actividades (SIPAC) cuente con el módulo Configuración que contribuye a que el proceso de planificación en las entidades cubanas se ejecute sobre las bases de una correcta organización, compartimentación y entendimiento común de la información entre y hacia los distintos niveles de la planificación.

Recomendaciones

Para aumentar las potencialidades en cuanto a usabilidad y robustez de SIPAC se recomienda:

- ✚ Desarrollar un componente, dentro del modulo Configuración, destinado a la configuración del perfil del planificador, donde se gestionen configuraciones que aumenten la usabilidad y flexibilidad del sistema, basadas en el manejo de las configuraciones de los calendarios por defecto o los deseados, la configuración de atributos dinámicos para la gestión de los elementos de la planificación y la gestión de relaciones dinámicas de dependencia entre los elementos de la planificación, entre otras funcionalidades.
- ✚ Agregar la funcionalidad de imprimir los grupos, permisos y nomencladores listados.
- ✚ Migrar hacia la última versión del Marco de Trabajo para explotar las nuevas y las mejoradas bondades del mismo relacionadas con los temas de compartimentación y la seguridad de los datos.

Referencia Bibliográfica

1. "El sistema de dirección y gestión empresarial en Cuba: un análisis imprescindible" en Observatorio de la Economía Latinoamericana. **León, Díaz Fernández y Echevarría**. Nº 143, 2011. <http://www.eumed.net/coursecon/ecolat/cu/2011/>.
2. Proceso de Planificación por Objetivos en las entidades de las FAR. **Bernal Vidal, Néstor y Galán Ramírez, Yuliet**. La Habana : UCI : s.n., 2008.
3. Administración. **James Arthur Finch, Stoner y otros**. 1996.
4. El Prisma. Administración de empresas. **Andrés M, Núñez Parada**. 2011.
5. The Practice of Management. **Drucker, Peter F**. 1954.
6. Modelo de Planificación por Objetivos en las FAR. **Bernal Vidal, Nestor, Galán Ramirez y Pereira, Mitzary**. La Habana:UCI : s.n., 21 de 04 de 2010.
7. Desarrollo Orientado a Objetos con UML. **Ferré, Xavier Grau**. 2011.
8. Curso Modelado de Negocio con BPMN 2.0 y UML. **Milestone, Consulting**. 2011.
9. Lenguaje HTML. **desarrolloweb.com**. 2002.
10. JavaScript. Mayo de 2010.
11. Diccionario de siglas CSS (Cascading Style Sheets). **Pérez, Iván Nieto**. 2011. <http://www.elcodigo.net/tutoriales/diccionario.html>.
12. Extensible Markup Language (XML). 2003. <http://conocimientoysistemas.wordpress.com/tag/definicion-xml/>.
13. http://php.ciberaula.com/articulo/introduccion_php.
14. CodeBox. Glosario framework. 2008. <http://www.codebox.es/glosario>.
15. Arquitectura Tecnológica para el Desarrollo de Software. **Oiner Gómez Baryolo, Yoandry Morejón Borbón, y Darien García Tejo**. Habana : UCI : s.n.
16. Learning Ext JS. **Steve 'Cutter' Blades, Colin Ramsay y Shea Frederick**. 2008.
17. ZEND FRAMEWORK. 2010. <http://framework.zend.com/manual/en>.
18. Doctrine. 2008. <http://www.doctrine-project.org>.
19. <http://www.doctrine-project.org>.
20. <http://svnbook.red-bean.com/nightly/es/svn-ch-1-sect-gacion>.
21. Visual-paradigm. 2005. <http://www.visual-paradigm.com>.

22. http://netbeans.org/community/releases/68/index_es.html..
23. Documentación del Servidor HTTP Apache 2.0. 2009. <http://httpd.apache.org/docs/2.0/es>.
24. PostgreSQL. <http://es.scribd.com/doc/36570462/postgreSQL-investigacion>..
25. <http://conceptodefinicion.de/firefox/>..
26. Ingeniería de Software. Séptima edición. **Sommerville, Ian**. Madrid : Pearson Educación S.A. : s.n., 2005.
27. Arquitectura de Software. **González, Dianelys Brito**. Habana : CEIGE : s.n., 2011.
28. Propuesta de arquitectura orientada a servicios para el módulo de inventario del ERP cubano. **Carrascoso, Enrique Gómez Chaviano y Yoan Arlet Puebla**. Habana : UCI : s.n., 2009.
29. Arquitectura Tecnológica para el Desarrollo de Software. **Gómez Baryolo, Oiner, Morejón Borbón, Yoandry y García Tejo, Darien**. Habana : UCI : s.n.
30. El mundo informático. 2006/08/17. <http://jorgesaavedra.wordpress.com>.
31. Patrones de Diseño. **Prieto, Félix**. 2009.
32. Definición. de. [En línea] 2008. <http://definicion.de/modelo-de-datos/>..
33. Proceso de Desarrollo y Gestión de Proyectos de Software. Habana: UCID : s.n., 2009.
34. Métrica de Diseño. **Ecured**. 2012.
35. Unidad de Programación. Métodos de pruebas del Software. **Ramírez, Jaime**.