



Universidad de las Ciencias Informáticas
Facultad 3

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

*“Solución informática para la caracterización de la
elaboración y ejecución del Plan anual de actividades en el Sistema
de Planificación de Actividades SIPAC.”*

Autor: Dayanís Díaz Patterson

Tutores: Ing. Mairelys Fernández González

Ing. Dionny Cardoso Carmona

Co-tutor: Ing. Roberto Bandera Gómez

La Habana, Enero del 2016.



La productividad no es un accidente. Siempre es el resultado de un compromiso con excelencia, planificación inteligente y esfuerzo concentrado.

Paul J. Meyer



DECLARACIÓN DE AUTORÍA

Declaro a Dayanís Díaz Patterson único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Autor: Dayanís Díaz Patterson

Tutor: Ing. Mairelys Fernández González

Tutor: Ing. Dionny Cardoso

Co-Tutor: Ing. Roberto Bandera Gómez



Agradecerles a mis padres por educarme e inculcarme buenos principios, por todo el apoyo que siempre he podido encontrar en ellos y por el sacrificio y esfuerzo que han tenido que hacer durante todos estos años en función de mi formación como profesional.

A mi mamá ante todo por estar cada minuto a mi lado, por darme todo su apoyo en cada paso de mi vida y por estar siempre a mi lado.

A mis tíos Luis y Lucía que desde hace dos años se convirtieron en mis segundos padres.

A mi hermano Damián que espero que siga mi ejemplo y se supere cada día más.

A mis primos Ramón, Yasel, a Yoel, Irina, Johnny, Canela, Yaima, Yanelis y Marine.

A mi abuela, a Magdalena, a mi tías y tíos Marlis, Marelis, Reina, Modesta, Mariela, Michel y a Norge.

A mis amigos Alián y Rolando por servir de apoyo en muchas ocasiones

A todas aquellas personas que siempre estuvieron dispuestas a darme su apoyo.

A las amistades de la Universidad en Manzanillo y a las nuevas que hice aquí en La Habana.

A la Revolución por permitirme la oportunidad de estudiar esta carrera.

Y por último y no menos importantes a mis tutores Mairelys, Dionnys, Rodolfo y Roberto por halarme las orejas cuando era necesario, por su apoyo y amistad y por tener la disposición y paciencia en cada explicación que me hacían, que fueron muchas.

Y a mi persona por tener el coraje y la fuerza de llegar hasta aquí.



Dedico este trabajo a mi mamá por su fe inquebrantable de que si se puede, por su amor, su cariño y su ternura. Por intentar transmitirme toda su sabiduría.

A mi papá por el amor y el cariño que me ha dado en estos 22 años.

A mi hermano que espero que le sirva de experiencia.

A toda mi familia, por ayudarme de una forma u otra cuando la necesita.

A todas aquellas personas que siempre estuvieron conmigo y supieron sacar una sonrisa en mi cuando todo parecía perdido.

A todas esas personas gracias y mil gracias.

RESUMEN

El Sistema de Planificación de Actividades (SIPAC) constituye una herramienta para la gestión de las actividades a todos los niveles organizacionales, basado en la Instrucción No.1 del Presidente de los Consejos de Estado y de Ministros para la planificación de objetivos y actividades en los Organismos de Administración Central del Estado, Órganos, Entidades nacionales y Administraciones Locales del Poder Popular. Permite interrelacionar objetivos de trabajo y actividades en tiempo real; garantizando el seguimiento del desarrollo y cumplimiento de los objetivos y tareas principales en las entidades como parte de la Planificación estratégica y operativa. Independientemente de las funcionalidades que brinda el sistema hoy, no posibilita la caracterización de la elaboración y ejecución de los planes anuales, lo que dificulta el registro y control de la Planificación estratégica tanto a nivel de entidad como a nivel de gobierno. El presente trabajo comprende el desarrollo de una solución informática para la caracterización de la elaboración y ejecución del Plan anual de actividades en SIPAC, donde el resultado está enfocado a un resumen cuantitativo. Determinado por desglose de tareas propias, externas y las tareas incorporadas al plan como parte de las puntualizaciones; que dan la medida de cómo cada organismo o entidad propone cumplir con su función o encargo estatal. El desarrollo de la funcionalidad estará guiado por la metodología de desarrollo Proceso Unificado Ágil (AUP por sus siglas en inglés) con variaciones de la Universidad de las Ciencias Informáticas (UCI), mediante la utilización de herramientas como *Visual Paradigm*, *Subversion*, *Netbeans*, *Firefox* y *Apache* las cuales permitirán dar culminación a todo el proceso de desarrollo.

Palabras claves: Planificación estratégica, Planificación Operativa, Elaboración y Ejecución.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Principales conceptos.....	5
1.2 Estudio del estado del arte.....	6
1.2.1 Dirección por objetivos (DPO).....	6
1.2.2 Dirección estratégica.....	7
1.2.3 Instrucción No.1 para la planificación.....	9
1.2.4 Valoración de los procedimientos estudiados.....	10
1.3 Metodología de desarrollo.....	11
1.4 Marco de trabajo.....	12
1.5 Lenguajes, tecnologías y herramientas para el desarrollo.....	13
1.5.1 Lenguajes de programación del lado del cliente.....	13
1.5.2 Lenguaje de Programación del lado del servidor.....	14
1.5.3 Lenguaje para el modelado.....	14
1.5.4 Tecnologías.....	15
1.5.5 Herramientas.....	15
1.6 Conclusiones parciales.....	16
CAPÍTULO 2: ANÁLISIS Y DISEÑO.....	17
2.1 Descripción de la solución.....	17
2.2 Modelo Conceptual.....	17
2.3 Reglas del negocio.....	19
2.4 Requisitos de software.....	21
2.4.1 Técnicas de captura de requisitos.....	21
2.4.2 Requisitos funcionales.....	22
2.4.3 Especificación de requisitos funcionales.....	24

2.4.4	Validación de los requisitos.....	26
2.4.5	Requisitos no funcionales.....	26
2.5	Modelado de la solución.....	28
2.5.1	Diseño de la solución en términos de componentes.....	28
2.5.2	Diagrama de clase del diseño.....	30
2.6	Patrón.....	31
2.6.1	Patrón arquitectónico.....	32
2.6.2	Patrones de diseño.....	32
2.7	Modelo de base datos.....	34
CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN.....		38
3.1	Implementación de la solución.....	38
3.1.1	Estándares de codificación.....	38
3.1.2	Nomenclatura de clases.....	38
3.1.3	Nomenclatura según el tipo de clases.....	39
3.1.4	Nomenclatura de las funcionalidades y atributos.....	39
3.1.5	Nomenclatura de los comentarios.....	39
3.2	Salidas de la funcionalidad desarrollada.....	40
3.3	Validación de la solución propuesta.....	41
3.3.1	Validación del diseño mediante métricas.....	41
3.4	Pruebas de software.....	45
3.4.1	Métodos de prueba.....	46
3.3.2	Resultados de las pruebas aplicadas.....	55
3.5	Impacto de la solución.....	55
3.6	Conclusiones parciales.....	56
CONCLUSIONES GENERALES.....		57
RECOMENDACIONES.....		58

REFERENCIAS BIBLIOGRÁFICAS.....	59
ANEXOS.....	62

ÍNDICE DE TABLA

Tabla 1: Reglas de negocio.....	20
Tabla 2: Descripción del requisito: Obtener caracterización del PAA.....	24
Tabla 3: Descripción de las clases del modelo de diseño.....	30
Tabla 4: Tamaño operacional de clase (TOC).....	40
Tabla 5: Relación entre clases (RC).....	41
Tabla 6: Caso de prueba para el camino a).....	48
Tabla 7: Caso de prueba para el camino b).....	49
Tabla 8: Descripción del caso de prueba para el requisito Caracterización del PAA.....	50
Tabla 9: Descripción de las variables.....	51
Tabla 10: Juegos de datos a probar.....	52
Tabla 11: Caso de prueba para el camino b).....	62

ÍNDICE DE FIGURA

Figura 1: Modelo Conceptual para la caracterización de la elaboración y ejecución del PAA.....	19
Figura 2: Prototipo para el requisito Obtener caracterización del PAA.....	25
Figura 3: Salida del sistema para el RF: Caracterización del PAA.....	26
Figura 4: Mapa general de componentes.....	29
Figura 5: Diagrama de componente para la funcionalidad propuesta.....	29
Figura 6: Diagrama de clase de diseño general.....	30
Figura 7: Modelo de datos para la Caracterización de la elaboración y ejecución del PAA.....	35
Figura 8: Salida para el RF Generar Resumen de cumplimiento del PTI.....	38
Figura 9: Salida para el RF Generar Resumen de cumplimiento del PM y PAA.....	38
Figura 10: Resultados de la evaluación de la métrica TOC para Responsabilidad y Complejidad de implementación.....	41
Figura 11: Resultados de la evaluación de la métrica TOC para Reutilización.....	41
Figura 12: Resultados de la evaluación de la métrica RC para Acoplamiento.....	42
Figura 13: Resultados de la evaluación de la métrica RC para Complejidad de mantenimiento y Cantidad de pruebas.....	43
Figura 14: Resultados de la evaluación de la métrica RC para Reutilización.....	43
Figura 15: Código a probar.....	46
Figura 16: Grafo de Flujo.....	47

INTRODUCCIÓN

La actualización del modelo económico cubano y con ello la implementación y desarrollo de los Lineamientos de la Política Económica y Social del Partido y la Revolución, hace necesario un seguimiento y control eficaz del proceso de planificación de objetivos y actividades en los diferentes organismos e instituciones estatales (1).

La “Planificación de actividades” es el instrumento central para el desempeño económico de un país y el desarrollo constante de las relaciones de producción, trazando pautas requeridas para el mejor funcionamiento de la economía (2).

En el 2011 se definió el modelo cubano para la “Planificación de actividades” descrito en la Instrucción No.1 del Presidente de los Consejos de Estado y de Ministros para la Planificación de los objetivos y actividades en los Órganos, Organismos de la Administración Central de Estado (OACE), Entidades nacionales y Administraciones Locales del Poder Popular (ALPP); que surge a raíz de la carencia de integralidad en los planes económicos y en la mayoría de los casos no se encuentran basados en objetivos concretos, la inadecuada vinculación entre los planes económicos y la planificación de las actividades y la falta de documentos rectores del gobierno para encauzar estos procesos.

El Sistema para la Planificación de Actividades SIPAC constituye una herramienta para la gestión de las actividades a todos los niveles organizacionales, que informatiza la Instrucción No.1, desarrollado el Centro de Informatización de Entidades (CEIGE), de la Universidad de las Ciencias Informáticas en conjunto con el Grupo de Planificación de la Secretaría del Consejo de Ministros.

SIPAC cuenta con varios módulos encargados de generar las configuraciones necesarias para el seguimiento de las tareas principales de cada entidad, gestión de los posibles involucrados, nomencladores y niveles de subordinación basados en reglas de la compartimentación de la información. Mediante el sistema se realiza el registro, seguimiento y control de los planes, las puntualizaciones de las actividades, la evaluación de los objetivos y la recuperación de la información, de acuerdo a los informes propuestos en la Instrucción No.1, específicamente: Plan de trabajo individual, Plan mensual, Plan anual de actividades y Resumen de cumplimiento. Sin embargo, en la elaboración de los planes no se incluye el total y porcentaje de las tareas propias que tributan a los objetivos estratégicos, las tareas externas y las actividades que dan aseguramiento en cada caso, ni están agrupadas por Área de Resultado Clave (ARC)

y capítulos del plan; no da la medida de cómo cada organismo o entidad va incorporando a su plan lo que hace cumplir con su función o encargo estatal de acuerdo a sus características y condiciones reales y la incidencia de factores externos.

Por otra parte en el Resumen de cumplimiento del plan no se evalúa la ejecución de las tareas propias, el impacto de las tareas externas y las variaciones del plan inicial de acuerdo a las tareas suspendidas o canceladas.

Considerando lo anteriormente expuesto se identifica el siguiente **problema a resolver**: ¿Cómo caracterizar la elaboración y ejecución del Plan anual de actividades en SIPAC de manera que se facilite el registro y control de la Planificación estratégica tanto a nivel de la entidad como a nivel de gobierno?

Con vista a dar solución al problema planteado, se define el siguiente **objeto de estudio**: Evaluación de la aplicación de la Instrucción No.1 del Presidente de los Consejos de Estado y de Ministros.

De esta forma la investigación se enmarca en el **campo de acción**: Informatización de la caracterización de la elaboración y ejecución del Plan anual de actividades en SIPAC.

Para dar solución al problema existente se traza como **objetivo general**: Desarrollar una solución informática para la caracterización de la elaboración y ejecución del Plan anual de actividades en SIPAC, de manera que facilite el registro y control de la Planificación estratégica tanto a nivel de la entidad como a nivel de gobierno.

Para cumplir con el objetivo propuesto se han definido los siguientes **objetivos específicos**:

- Elaborar el marco teórico para sustentar los conceptos, la propuesta de desarrollo de las funcionalidades, las herramientas y tecnologías a utilizar.
- Modelar el negocio asociado a la solución, para comprender los elementos significativos de la misma.
- Realizar el diseño de la solución a implementar teniendo en cuenta las necesidades del cliente.
- Implementar la solución para la caracterización de la elaboración y ejecución del Plan anual de actividades en SIPAC.
- Validar la funcionalidad propuesta mediante pruebas internas y métricas de validación para el diseño.



- La **idea a defender** con el desarrollo de una solución informática para la caracterización de la elaboración y ejecución del Plan anual de actividades en SIPAC, se facilitarán el registro y control de la Planificación estratégica tanto a nivel de la entidad como a nivel de gobierno.

Para dar cumplimiento a los objetivos específicos se definen las siguientes **tareas de la investigación**:

- Análisis de la arquitectura de SIPAC para conocer las características fundamentales, frameworks, herramientas y tecnologías definidas para el desarrollo.
- Estudio de los patrones de Diseño.
- Elaboración del Modelo conceptual y Reglas del negocio.
- Identificación y especificación de los requisitos de software.
- Diseño de los prototipos de interfaz de usuario.
- Validación de los requisitos funcionales.
- Análisis del Modelo de Datos de SIPAC.
- Elaboración del diseño de las clases asociado a la solución.
- Implementación de los requisitos identificados.
- Validación del diseño mediante métricas.
- Validación mediante los métodos de pruebas de caja negra y caja blanca.

Métodos teóricos y empíricos

Para el desarrollo de la solución informática se emplearon métodos teóricos y empíricos de la investigación científica. Los **métodos teóricos** utilizados para cumplir con las tareas a desarrollar son:

Analítico-Sintético: se aplicó durante la investigación para analizar la bibliografía existente relacionada con la planificación de actividades tanto nacional como internacionalmente para identificar características y mejores prácticas. Además, se empleó en el análisis de las herramientas y tecnologías utilizadas para el desarrollo de la solución.

Entre los **Métodos Empíricos** se emplearon:

La Entrevista: se realizó al cliente para identificar cuáles son las principales funcionalidades a cumplir por el sistema, así como para hacer un estudio detallado del seguimiento y control de la planificación a corto, mediano y largo plazo tanto a nivel de la entidad como a nivel de gobierno.



La Observación: se aplicó durante la investigación para diagnosticar el estado actual del seguimiento y control de la planificación a corto, mediano y largo plazo tanto a nivel de la entidad como a nivel de gobierno.

Estructura del documento

El presente documento se estructura en 3 capítulos desglosados de la siguiente forma:

Capítulo 1. Fundamentación teórica: Se describen los principales conceptos acerca del estado del arte referente a la caracterización de la elaboración y ejecución del Plan anual de actividades. Se describen las herramientas y tecnologías que permitirán el diseño e implementación de las funcionalidades que darán respuesta al problema planteado. Se hace una descripción de la metodología que guiará el desarrollo de la solución.

Capítulo 2. Análisis y diseño: Se identifican los requisitos funcionales y no funcionales asociados a la propuesta de solución del problema planteado. Se realiza el diseño del sistema y se analizan los patrones de diseño a emplear, se modela la solución de acuerdo con la metodología empleada para su desarrollo.

Capítulo 3. Implementación y validación: Se valida la solución propuesta a través de las pruebas internas, el diseño por métricas de validación y se analizan los resultados obtenidos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se analizan los fundamentos teóricos asociados a la solución donde se realiza un estudio de modelos de dirección y la Instrucción No.1, para hacer una valoración del estado del arte referente a la elaboración y ejecución del Plan anual de actividades. Se presentan las tecnologías, metodología, los lenguajes y herramientas que fueron usadas en la solución propuesta.

1.1 Principales conceptos

Plan: Es un modelo sistemático que se desarrolla antes de concretar una cierta acción con la intención de dirigirla. También constituye un instrumento de planificación y gestión que permite llevar a cabo los fines de la organización, mediante una adecuada definición de los objetivos y metas que se pretenden alcanzar de manera que se utilicen con eficiencia, eficacia y economicidad. Contiene al menos: información referente al programa, objetivos (generales y específicos), actividades y metas, indicadores del cumplimiento de las metas, fechas de su realización, recursos económicos y materiales que sustenten su ejecución y personal responsable (3).

Planificación: Es la primera función administrativa porque sirve de base para las demás funciones. Esta función determina por anticipado cuáles son los objetivos que deben cumplirse y que debe hacerse para alcanzarlos; por tanto, es un modelo teórico para actuar en el futuro. La planificación comienza por establecer los objetivos y detallar los planes necesarios para alcanzarlos de la mejor manera posible. La planificación determina donde se pretende llegar, que debe hacerse, como, cuando y en qué orden debe hacerse (4).

Tareas externas: Son aquellas tareas que están contenidas en los planes del Partido y de la Asamblea Nacional del Poder Popular que tienen incidencias en los diferentes niveles de dirección, así como, las diferentes organizaciones de masas de carácter político e ideológico. Son tareas propuestas por otras organizaciones que cumplen con los objetivos de una organización específica y que pertenecen a su plan (5).

Tareas internas: Son aquellas tareas que están contenidas en los planes de una organización que son propuestas por ella misma. Su función principal es dar respuestas a todos los objetivos definidos por la misma y donde los subordinados estarán directamente encauzándola (5).

Actividades de aseguramiento: Son aquellas actividades que en su conjunto contribuirán a que se cumplan todas las tareas propias o externas contenidas en los planes de anuales, además le otorga importancia a las tareas (5).

Tareas puntualizadas: Son aquellas tareas que se precisan o se ajustan una vez que fueron aprobadas o que se introducen debido a la implementación de documentos jurídicos, acuerdos u otras decisiones de los jefes facultados para modificar los planes aprobados, así como otras causas que incluyan errores en la planificación (5).

Tareas extraplanes: Son las tareas que se agregan al plan luego de haberse aprobado debido a la implementación de documentos jurídicos, acuerdos u otras decisiones de los jefes facultados para modificar los planes aprobados, así como otras causas que incluyan errores en la planificación (5).

1.2 Estudio del estado del arte

A continuación se presenta un estudio de modelos de dirección, para identificar sus principales características y mejores prácticas, de manera que puedan ser aplicadas a la propuesta de solución.

1.2.1 Dirección por objetivos (DPO)

Es una herramienta para evaluar el desempeño, que integra los objetivos individuales y organizacionales, formando también parte de la Planificación estratégica de la organización. En la DPO la relación entre los objetivos del individuo y las metas comunes de la organización es algo de vital importancia (6).

Los tres elementos esenciales de la DPO son:

- El establecimiento de objetivos claros, concisos y comunicados.
- La participación en el proceso de establecimiento de los objetivos.
- La evaluación de las actuaciones basada en los resultados.

Elaboración: La definición de los objetivos establece la base de relación entre la organización y su entorno. El directivo o la persona que definen los objetivos debe tener en cuenta todos los factores que influyen en la organización de la que forma parte: el capital, el trabajo, los factores económicos, los factores políticos, las condiciones del mercado, la normativa, las posibilidades de suministro y el precio de las materias primas, etc. No obstante, la elección de un determinado objetivo también está condicionada por los intereses profesionales, las preferencias, las motivaciones y la formación de la persona que

efectúa dicha elección. Los objetivos de todo directivo deben reflejar su contribución al logro de las metas de la compañía en todos los aspectos del negocio (7).

Es importante tener en cuenta que antes de iniciar cualquier acción, los objetivos deben ser claramente identificados, comprendidos y definidos, debido a que sin una clara y precisa definición en términos accesibles y comprensibles para todos los interesados, no se puede pretender que los objetivos sean interpretados correctamente y sean aceptados. Sólo si los objetivos son comprendidos y aceptados pueden ser considerados como propios por aquellos que deben actuar para alcanzarlos.

Evaluación: En las sesiones de revisión, los directivos y sus subordinados examinan los objetivos que se establecieron en su día, comprueban los que se han logrado, aquellos en los que aún no se ha comenzado a trabajar y los que implican una actividad continua. El objetivo de estas sesiones es que no pase nada desapercibido. Estas reuniones proporcionan la oportunidad para resumir cómo va la marcha de lo que se planificó y revisarlo.

Un aspecto relevante de la DPO es la comunicación que debe existir entre los superiores y subordinados, tanto en el establecimiento de objetivos como en la evaluación de los mismos, donde todos sus miembros deben conocer y comprender los objetivos de la empresa y lo que se espera.

Observaciones sobre la DPO: Para la definición de los elementos de planificación propios, en este caso los objetivos, se debe tener en cuenta todos los factores que influyen en la organización de la que forman parte, los elementos propios que la distinguen y sus condiciones reales.

Y la evaluación debe estar sustentada por una propuesta de evaluación discutida por superiores y subordinados, donde además se definan niveles de consecución, para ponderar los objetivos determinantes en el cumplimiento de lo planificado.

1.2.2 Dirección estratégica

Modo en el que la empresa despliega sus recursos y capacidades en el entorno con el fin de alcanzar sus objetivos. Se caracteriza por:

- La incertidumbre del entorno, el comportamiento de competidores y las preferencias de clientes.
- La complejidad de distintas formas de percibir el entorno y de interrelacionarse con empresa.
- Los conflictos organizativos entre los que toman decisiones y los afectados por ellas.

La denominación de dirección estratégica, que podemos definirla como el arte y la ciencia de poner en práctica y desarrollar todos los potenciales de una empresa, que le aseguren una supervivencia a largo plazo y a ser posible beneficiosa (8).

La dirección estratégica puede ser dividida en tres fases:

- **Definición de objetivos estratégicos:**
 - Definir la filosofía y misión de la empresa o unidad de negocio.
 - Establecer objetivos a corto y largo plazo para lograr la misión de la empresa, que define las actividades de negocios presentes y futuras de una organización.

- **Planificación estratégica:**
 - Formular diversas estrategias posibles y elegir la que será más adecuada para conseguir los objetivos establecidos en la misión de la empresa.
 - Desarrollar una estructura organizativa para conseguir la estrategia.

- **Implementación estratégica:**
 - Asegurar las actividades necesarias para lograr que la estrategia se cumpla con efectividad.
 - Controlar la eficacia de la estrategia para conseguir los objetivos de la organización.

El proceso de dirección estratégica requiere una planificación, un proceso continuo de toma de decisiones, decidiendo por adelantado qué hacer, cómo hacerlo, cuándo hacerlo y quién lo va a hacer (8).

Elaboración: La dirección estratégica, como parte de la dirección empresarial, se ocupa de formular e implantar la estrategia, movilizandolos recursos de la empresa y abordando la creciente complejidad de las empresas y del entorno. El proceso de dirección estratégica requiere una planificación, un proceso continuo de toma de decisiones, decidiendo por adelantado qué hacer, cómo hacerlo, cuándo hacerlo y quién lo va a hacer. Esta toma de decisiones estratégicas es función y responsabilidad de directivos de todos los niveles de la organización, pero la responsabilidad final corresponde a la alta dirección. Es esta quien establecerá la visión, la misión y la filosofía de la empresa (9).

El análisis interno, además de identificar las fortalezas y debilidades, trata de dar una visión de conjunto sobre los recursos, medios principales y habilidades para hacer frente al entorno.

La denominación de la matriz DAFO (ver Anexo 1) hace referencia al objeto de su estudio:

- Debilidades de la empresa

- Amenazas del entorno
- Fortalezas internas de la empresa
- Oportunidades del entorno

Constituye una valiosa metodología para formular la estrategia a nivel de negocio, siendo especialmente útil en el análisis estratégico.

Evaluación: Para la realización de la evaluación se debe tener en cuenta los siguientes factores.

- Examinar las bases subyacentes a la estrategia de una empresa.
- Comparar los resultados esperados con los reales.
- Tomar acciones correctivas para asegurarse que el desempeño va de acuerdo con los planes.

Observaciones sobre la Dirección estratégica: Para la elaboración de los elementos de planificación en este caso, los planes, la dirección estratégica propone que sea función y responsabilidad de directivos de todos los niveles de la organización, pero la responsabilidad final corresponde a la alta dirección, donde se consolida y aprueba el plan. Destaca las acciones correctivas o puntualizaciones para asegurarse del desempeño de los planes. En cuanto a la evaluación es preciso comparar los resultados esperados con los reales.

1.2.3 Instrucción No.1 para la planificación

Instrucción No.1: establece un modelo que guía el proceso de planificación de actividades y objetivos tanto en organizaciones políticas como administrativa en el país, que permite dar cumplimiento a los acuerdos y resoluciones aprobadas en el VI Congreso del Partido Comunista de Cuba, las decisiones de la Asamblea Nacional del Poder Popular, el Consejo de Ministros y la actualización de los planes de la economía. Este modelo define las directrices de la Planificación estratégica y operativa de actividades, posibilitando que las actividades individuales y organizacionales se ajusten a las necesidades propias de las instituciones cubanas (10).

Elaboración: Para desarrollar el proceso de planificación de objetivos y actividades del Gobierno, tendrán en cuenta los propósitos siguientes:

- Definir los objetivos de trabajo que aseguran el continuo desarrollo del país, con el empleo racional y eficiente de los recursos disponibles a corto, mediano y largo plazos, así como la medición y evaluación sistemática de estos.

- Lograr una adecuada organización y coordinación de todos los factores implicados, que intervienen en el proceso, en interés del cumplimiento de los objetivos de trabajo.
- Garantizar eficiencia y eficacia en la implementación de lo planificado y en la comprobación de su ejecución mediante las diferentes formas de control.

Evaluación: Para valorar los resultados alcanzados hasta la fecha en que se realiza el análisis los jefes y el personal designado elaboran un informe resumen de cumplimiento del Plan de trabajo, tanto del individual, como del mensual de la organización, los cuales se archivan por un período de un año, dirigiendo su contenido a evaluar los aspectos siguientes:

- **Cuantitativo:** Reflejar el total de tareas planificadas y cumplidas. De ellas: las externas, las propias y las nuevas que se incorporen al plan. Para cada uno de estos aspectos se le asigna un número y un porcentaje que luego depende de la cantidad de tareas propuesta, cumplidas e incorporadas.
- **Cualitativo:** Realizar un análisis de los resultados más significativos, principales dificultades afrontadas y sus causas, contribución de las tareas valoradas a los objetivos previstos, así como propuestas de decisiones a tomar para asegurar el cumplimiento de lo planificado.

Observaciones sobre la Instrucción No.1: La Instrucción No.1 destaca que en el proceso de elaboración de objetivos y actividades, se tenga en cuenta la posibilidad de medir y evaluar de manera sistemática los objetivos y actividades mediante las diferentes formas de control. Que se identifique lo que hace la organización que asegura el continuo desarrollo del país como tareas principales.

Y para la evaluación reflejar de forma cuantitativa los resultados; en función del total de tareas planificadas, cumplidas, externas, propias y las nuevas incorporadas al plan, así como los resultados más significativos, dificultades afrontadas y causas en una evaluación cualitativa .

1.2.4 Valoración de los procedimientos estudiados

Con el estudio de los modelos de dirección antes presentados así como, de la Instrucción No.1, se destacan los conceptos de las tareas propias, tareas externas, y puntualización como elementos fundamentales tanto para la elaboración de los planes como para la evaluación de los mismos. Teniendo en cuenta que no existe un informe previamente definido que incluya estos elementos, que además estén adaptados a lo establecido para la planificación de objetivos y actividades en Cuba, se evidencia la

necesidad de incorporar en SIPAC los conceptos e informes que complementen la caracterización y evaluación de los planes.

1.3 Metodología de desarrollo

Para el desarrollo de nuevos sistemas informáticos en la universidad de las ciencias informáticas se basa en la metodología AUP con variaciones propias de la UCI en unión con el modelo CMMI-DEV¹ v1.3. Esta contribuye a que estos sistemas se desarrollen con calidad y además que exista una homogeneidad en ellos ya que en la universidad se desarrollan un gran número de proyectos (11).

AUP en su variación cuenta con 3 fases:

Inicio: en esta etapa se llevan a cabo las actividades relacionadas con la planeación del proyecto. Se realiza un estudio acerca de la organización cliente para obtener la información necesaria acerca de lo que se pretende desarrollar, se realizan las estimaciones de tiempo, esfuerzo y costo y es aquí también donde se decide si el proyecto se ejecuta o no.

Ejecución: aquí se desarrollan las actividades que van a permitir darle respuesta a lo que se quiere, es decir, se implementan las funcionalidades identificadas teniendo en cuenta la arquitectura, los planes del proyecto y los requisitos. Además, se lleva a cabo el modelado del negocio y por último se libera.

Cierre: en esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Se decide para el ciclo de vida de los proyectos de la UCI tener 7 disciplinas, pero a un nivel más atómico. Seguidamente se explican las disciplinas por las que transitará la funcionalidad propuesta:

Requisitos: esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales.

Análisis y diseño: aquí se realizará un análisis para modelar la solución, incluyendo la arquitectura, de manera tal que soporte los requisitos funcionales y no funcionales.

Implementación: se materializará la solución a partir de los resultados obtenidos en la disciplina anterior.

Pruebas internas: después de implementadas todas las funcionalidades, se verifica el resultado para su aprobación; se generan ciertos artefactos de pruebas entre ellos, los diseños de casos de prueba, entre otros.

¹ **Integración de modelos de madurez de capacidades** (Capability Maturity Model Integration) para Desarrollo (CMMI-DEV): proporciona un conjunto completo e integrado de guías para desarrollar productos y servicios.

Pruebas de aceptación: Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

1.4 Marco de trabajo

Un marco de trabajo es una estructura de soportes de programas, librerías y lenguajes de script. Es considerado una arquitectura de software que modela las relaciones generales de los componentes del proyecto que lo implementa; provee una estructura y manera de trabajo la cual utilizan las aplicaciones del proyecto. La finalidad de los marco de trabajo es facilitar el desarrollo de software, permitiéndoles a diseñadores y programadores concentrarse en los requerimientos del proyecto, reduciendo los posibles problemas con las tecnologías utilizadas, así como facilitando ciertas funcionalidades básicas y comunes (12).

Es una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un marco de trabajo se puede considerar como una aplicación genérica incompleta y configurable que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web (12).

Sauxe v2.0: El marco de trabajo *Sauxe* es una base tecnológica que facilita el desarrollo de aplicaciones de gestión para la web. Uno de los objetivos de *Sauxe* es permitir la reutilización de los componentes que se desarrollan sobre él, estos no son extensibles o personalizables más allá de los sistemas para los que fueron concebidos. La ausencia de un estándar que especifique cómo definir, documentar, empaquetar o integrar componentes, es una de las causas que inciden en el bajo índice de reutilización (13).

Zend Framework v1.9.7: es un marco de trabajo de código abierto para el manejo de la lógica de negocio en el desarrollo de aplicaciones web y servicios web con PHP (*Hypertext Preprocessor*). Su código es orientado a objetos y posee un bajo acoplamiento entre componentes, lo que permite a los desarrolladores utilizar los componentes por separado. Implementa el patrón Modelo Vista Controlador (MVC). Consta de mecanismos de filtrado y validación de entradas de datos. Permite convertir estructuras de datos PHP a JSON (*JavaScript Object Notation*) y viceversa, para su utilización en aplicaciones AJAX (*JavaScript y XML² asíncronos*) y provee capacidades de búsqueda sobre documentos y contenidos (14).

² **XML** siglas en inglés de *eXtensible Markup Language* ("lenguaje de marcas Extensible"), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.

Doctrine v1.2.2: es empleado para la capa de acceso a datos. Es un sistema ORM (en inglés *Object Relational Mapper*) para PHP 5.2 o superior que incorpora una DBL³ (capa de abstracción a base de datos). Uno de sus rasgos importantes es la habilidad de escribir opcionalmente las preguntas de la base de datos orientada a objeto. Esto les proporciona una alternativa poderosa a diseñadores de SQL⁴, manteniendo un máximo de flexibilidad sin requerir la duplicación del código innecesario. Además, exporta una base de datos existente a sus clases correspondientes y convierte clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos (15).

Ext Js v2.2: es un marco de trabajo de JavaScript para el desarrollo de aplicaciones, su función es crear páginas o aplicaciones web con contenido dinámico, aplicaciones cliente-servidor y animaciones de componentes de las páginas. Trabaja con el patrón de diseño Modelo Vista Controlador que permite la creación de aplicaciones para aprovechar funciones como desplazamiento infinito de una cuadrícula y para crear un nuevo nivel de interactividad a las aplicaciones web (16).

1.5 Lenguajes, tecnologías y herramientas para el desarrollo

Los lenguajes, tecnologías y herramientas a usar para desarrollar la funcionalidad propuesta, son las definidas por el equipo de desarrollo del proyecto SIPAC, éstas se describen a continuación:

1.5.1 Lenguajes de programación del lado del cliente

JavaScript 1.6: es un lenguaje de programación de scripts (secuencia de comandos) orientado a objetos. Actualmente es principalmente utilizado en internet, junto con las páginas web. Javascript está directamente incluido en la página web (o en un archivo externo) y mejora una página HTML (*HyperText Markup Language*), añadiendo interacción del usuario, animación y ayudas a la navegación. Se dice que es un lenguaje del lado del cliente, es decir que los scripts son ejecutados por el navegador del usuario (cliente). Esto difiere de los llamados lenguajes de script del lado del servidor que son ejecutadas por el servidor web (17).

Se usa JavaScript como lenguaje de programación interpretado utilizado principalmente en su forma del lado del cliente para la captura de evento dentro de la interfaz. Además, pone a disposición del

³ **DBL** siglas en inglés de Domain Block List. es una base de datos en tiempo real de los dominios (típicamente dominios de sitios web) que se encuentran en los mensajes de spam.

⁴ **SQL** (por sus siglas en inglés *Structured Query Language*) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

programador todos los elementos que forman la página web para acceder a ellos y modificarlos dinámicamente.

HTML en español (Lenguaje de Marcado de Hipertexto): es el lenguaje que se utiliza para crear páginas web. Una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo. Es un lenguaje que se emplea en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica (18).

Hoja de estilo en cascada o **CSS** (siglas en inglés de *cascading style sheets*): constituye un estándar que permite establecer el estilo (tamaños, iconos, imágenes, colores, espacios y bordes) de documentos estructurados, dígame la capa de presentación de una página web. Este lenguaje permite definir el aspecto visual del documento, mientras que separa la parte semántica (HTML) de la presentacional (*style sheets*) (19).

1.5.2 Lenguaje de Programación del lado del servidor

PHP v5.4: es el lenguaje que se empleará para programar del lado del servidor. Es interpretado y completamente orientado al desarrollo de aplicaciones web dinámicas. Es gratuito, fácil de usar y aprender, portable, de código abierto y multiplataforma. Presenta interfaces para una gran cantidad de sistemas de base de datos diferentes, así como bibliotecas incorporadas para muchas tareas web habituales (20).

1.5.3 Lenguaje para el modelado

UML v2.0 (Lenguaje Unificado de Modelado): UML permite una modelación de los componentes estáticos de una aplicación software (diagramas de casos de uso, diagramas de clases), así como del comportamiento dinámico de sus principales elementos durante su funcionamiento (entre ellos diagramas de estados y diagramas de secuencias) (21).

El modelado UML ayuda a una rápida construcción de aplicaciones de calidad y mejoras a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

1.5.4 Tecnologías

JSON: se basada en texto ligero, independiente del lenguaje de formato de intercambio de datos. Fue derivado de la programación *ECMAScript* que es un lenguaje estándar. *JSON* define un pequeño conjunto de reglas de formato para la representación portátil de los datos estructurados (22).

Tiene muchas ventajas como la simplicidad, velocidad y facilidad de lectura. Permite una integración sencilla con otras herramientas mientras que proporciona un aumento en el rendimiento, además posibilita construir aplicaciones más rápidas e interactivas.

AJAX: es una técnica empleada en el desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios, mientras se mantiene la comunicación asíncrona con el servidor en segundo plano (23).

Permite realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad y velocidad en las aplicaciones.

1.5.5 Herramientas

Visual Paradigm v8.0: Es una herramienta CASE⁵ que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software. Permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores en la codificación del software (24).

Netbeans 8.0 como Entorno Integrado de Desarrollo: La presente funcionalidad se desarrolla sobre el IDE de programación multiplataforma *NetBeans* 8.0. Es un producto de código abierto, con todos los beneficios del programa disponible en forma gratuita. Hace uso de *plugins* para ampliar sus funcionalidades, lo que le da una gran facilidad de uso (25).

Subversion 1.6.16 como herramienta para el control de versiones: es la herramienta de entorno colaborativo que se utilizará para el control de versiones. Se encuentra preparado para funcionar en red y se distribuye bajo licencia libre. Además de los cambios en el contenido de los documentos, se mantiene la historia de todas las operaciones de cada elemento, incluyendo la copia, cambio de directorio o de nombre (26).

⁵ **CASE** (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas o programas informáticos destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

PostgreSQL v9.1 como sistema gestor de Base de Datos: *PostgreSQL* es un sistema de gestión de base de datos relacional orientada a objeto. Es de código abierto y es compatible con una gran parte del estándar SQL. Debido a la licencia liberal, *PostgreSQL* puede ser utilizado, modificado y distribuido por cualquier persona de forma gratuita (27).

Apache v2.2 como servidor de aplicaciones: es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP⁶. Es una tecnología gratuita de código abierto. Se ejecuta en varios sistemas operativos, característica que lo hace prácticamente universal. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es altamente configurable en la creación y gestión de *logs* (28).

Mozilla Firefox 29 o superior como navegador web: es un navegador libre y de código abierto. Es usado para visualizar páginas web. Incluye corrector ortográfico, búsqueda progresiva y marcadores dinámicos. Además, se pueden añadir funciones a través de complementos desarrollados por terceros. Contiene el plugin *Firebug* que se utiliza para ver los errores del código. Es multiplataforma, realiza la navegación por pestañas, presenta compatibilidad para múltiples extensiones.

1.6 Conclusiones parciales

El estudio de modelos de dirección así como de la Instrucción No.1, evidenció la necesidad de incorporar a SIPAC modelos para la caracterización, resúmenes de cumplimiento y desglose por ARC de los planes, donde se representen las tareas propias, externas y puntualizadas que permitan evaluar de manera cuantitativa y cualitativa la elaboración y ejecución de lo planificado. Se presentaron conceptos fundamentales asociados al dominio del problema, el modelo desarrollo AUP con variaciones UCI que debe guiar el desarrollo de la solución, así como las tecnologías y herramientas, conjuntamente con sus versiones, definidas en la Arquitectura base del sistema.

⁶ **Hypertext Transfer Protocol** o **HTTP** (en español *protocolo de transferencia de hipertexto*) es el protocolo de comunicación que permite las transferencias de información en la World Wide Web.

CAPÍTULO 2: ANÁLISIS Y DISEÑO

En el presente capítulo se realiza la descripción de la solución propuesta, para proporcionar un mayor entendimiento. Se modela la solución mediante la definición del modelo conceptual y reglas del negocio. Se identifican y describen de los requisitos funcionales y no funcionales de acuerdo a las necesidades del cliente y como parte del diseño se construyen, empleando la notación UML, los diagramas que representan tanto la comunicación entre las clases del diseño, como el modelo de datos. Se especifican además los patrones de diseño aplicados.

2.1 Descripción de la solución

La propuesta de solución permite la especificación de las tareas propias y externas, las tareas nuevas incorporadas como parte de la puntualización y las suspendidas o canceladas. Se agrega un informe para la caracterización del Plan anual de actividades que incluye el total de las tareas propias, externas y las actividades que dan aseguramiento en cada caso, que además estarán agrupadas por capítulos del plan, facilitando el registro de la Planificación estratégica. Se propone un modelo para el resumen de cumplimiento del Plan mensual y Plan anual de actividades donde se incluyen las tareas externas, las propias y las nuevas incorporadas como parte de la puntualización. Este resumen permitirá realizar el análisis de lo que se planificó, lo que se cumplió y no se cumplió, calculando también las tareas suspendidas o canceladas y las extraplanes. Se actualizará el resumen de cumplimiento del Plan de trabajo individual que permitirá controlar y evaluar la ejecución de las tareas planificadas y las variaciones del plan inicial de acuerdo a las tareas suspendidas o canceladas y la introducción de otras nuevas. El desglose de actividades por ARC para el año y cada trimestre de este mostrará un análisis cuantitativo de las actividades propuestas en el plan del Gobierno.

2.2 Modelo Conceptual

El mapa conceptual es una representación gráfica de un conjunto de conceptos y sus relaciones sobre un dominio específico de conocimiento (29).

A continuación se presenta el modelo conceptual asociado a la solución para la caracterización de la elaboración y ejecución del Plan anual de actividades, que describe las clases dentro del contexto que se está analizando.

Actividad: Conjunto de operaciones o tareas, propias de una persona o entidad, destinadas para cumplir determinado(s) objetivo(s).

Plan: Es un modelo sistemático que se elabora antes de realizar una acción, con el propósito de dirigirla y encauzarla. En este sentido, un plan también es un documento que precisa los detalles necesarios para realizar una misión.

Objetivo: Un objetivo es una meta o finalidad a cumplir para la que se disponen medios determinados. Los objetivos se pueden clasificar en estratégicos (generales y a largo plazo), por área (funcionales) e individuales.

Elemento de la planificación: Son los elementos que se toman en consideración dentro de la planificación, estos se refieren a los planes, los objetivos y las actividades.

Tipo de actividad: Son los diferentes tipos de actividades que pueden existir dentro de la planificación.

Categoría de la actividad: Son las clasificaciones que pueden tener o tomar las actividades dentro de la planificación.

ARC: es el término empleado para definir un resultado cuyo logro implica la participación de dos o más elementos (departamento, dirección, grupo de trabajo, etc.) de la estructura funcional de una entidad. Tal logro es esencial para que la organización cumpla su misión.

Reporte: efectúa el análisis de la planificación mostrando al usuario la información organizada en un formato desde el cual se puede imprimir para ser distribuido.

Externa: Son todas aquellas tareas que sean propuestas por organismos externos a la organización o entidad.

Propia: Son todas aquellas tareas que sean propuestas por el mismo organismo.

Aseguramiento: Son tareas que contribuye a que una actividad sea ejecutada correctamente.

Extraplan: Se refiere a todas las tareas que se propongan luego de que se haya aprobado el plan mensual o anual. Estas actividades son incluidas en el plan y elaboradas por los mismos organismos y entidades que realicen el plan inicial.

Puntualizada: Son las tareas que son modificadas o cambiadas en el plan.

PDF y EXCEL: Formato del documento que se genera del resumen seleccionado.

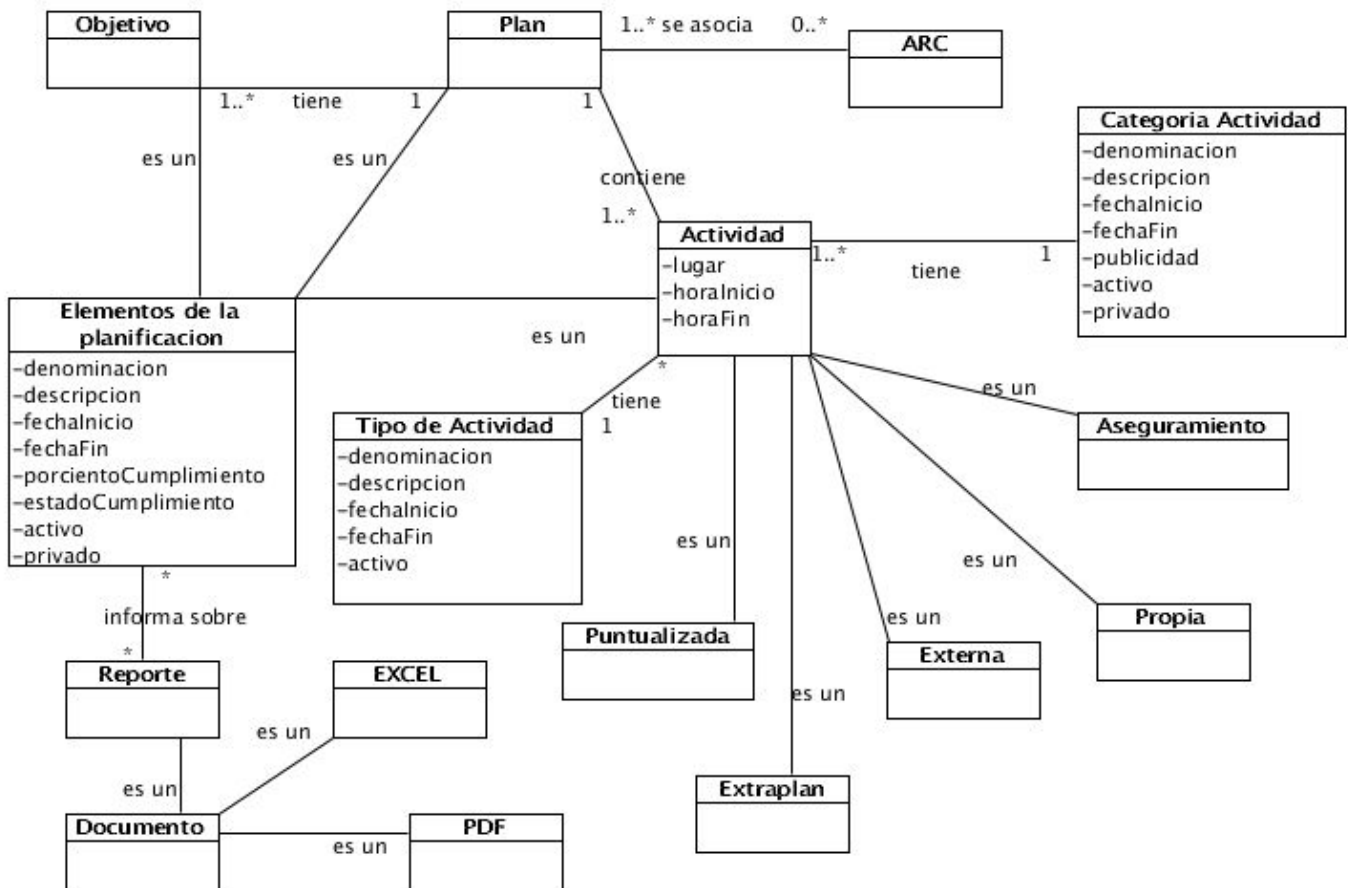


Figura 1: Modelo Conceptual para la caracterización de la elaboración y ejecución del PAA.

2.3 Reglas de negocio

Las reglas de negocio permiten definir las condiciones para establecer un negocio o precisar de qué forma se controlará el comportamiento de los eventos dentro de este. Una regla de negocio es una declaración que define, establece u obliga un cierto aspecto del negocio (30).

La caracterización de la elaboración y ejecución del PAA se enmarca en los procesos de negocio: Establecer Plan, en este caso para el informe de caracterización del PAA y el desglose por ARC del PAA y Ejecución del sistema de control, para los informes de resúmenes del cumplimiento de lo planificado. Para la funcionalidad, se hizo uso de las reglas de negocio definidas por el proyecto SIPAC. A continuación se presentan las descripciones de las reglas para la funcionalidad. Las demás reglas se encuentran descritas en el expediente de proyecto en la siguiente dirección Exp_Proyecto/Ingenieria/Requisitos/Modelado_del_negocio/Reglas_Negocio.



Tabla 1: Reglas de negocio

No	Tipo	Nombre	Descripción
1	Textual	Recuperaciones	Para obtener cualquier tipo de reporte se debe seleccionar el plan del cual se quiere obtener el reporte y el rango de fechas.
2	Textual	Recuperaciones	Las actividades que se muestran en el Plan Anual de Actividades, son aquellas que han sido relacionadas al Plan que se seleccione y a ARC(s) directamente.
3	Textual	Recuperaciones	Los datos del Encabezado y Pie de firma se pueden configurar en el sistema.
4	Textual	Recuperaciones	Para obtener en el Reporte "Resumen del cumplimiento del plan" el Breve Resumen cualitativo se debe llenar el campo Descripción en el plan seleccionado.
5	Textual	Recuperaciones	Para obtener el informe de Caracterización del PAA y Desglose por ARC se debe señalar la opción Caracterización del plan o Desglose por ARC respectivamente además de una fecha y un plan.
6	Textual	Recuperaciones	El Resumen de Cumplimiento para los planes es diferente según su categoría: para el plan individual se genera el Resumen de Cumplimiento

Individual y para el
anual y mensual es el
Resumen de
Cumplimiento Mensual.

2.4 Requisitos de software

La ingeniería de requisitos es uno de los procesos fundamentales en el desarrollo de software. Su principal objetivo se enmarca en el análisis, mediante el cual se documentan las necesidades funcionales que deberán ser soportadas por el subsistema a desarrollar. Para el diseño e implementación es necesario satisfacer un conjunto de requerimientos; para la identificación de los mismos se hace uso de la entrevista como técnica de recopilación de información. Esta consiste en una conversación dirigida con un propósito bien definido, en la que se usa un formato de preguntas y respuestas. Puede estar o no estructurada (31).

2.4.1 Técnicas de captura de requisitos

La obtención de requisitos es una de las etapas más importantes en el proceso de desarrollo del software. Una buena comprensión de los requisitos puede conducir a mejores productos de software que satisfagan las necesidades de los interesados. Sin embargo, el proceso de captura de requisitos se torna difícil para el analista debido, en gran parte, al carácter presencial que tienen las reuniones que se realizan con tal fin y a la dificultad que presentan algunas personas para expresar sus ideas de forma clara (32).

Las técnicas empleadas para la obtención de los requisitos que debe cumplir la funcionalidad se describen a continuación:

- **Entrevista:** Es de gran utilidad para obtener información cualitativa como opiniones, o descripciones subjetivas de actividades. Es una técnica muy utilizada, y requiere una mayor preparación y experiencia por parte del analista. Es muy importante la forma en que se plantea la conversación y la relación que se establece en la entrevista (33).

Se utilizó de manera frecuente en los diferentes encuentros con el cliente mediante preguntas, con el objetivo de obtener toda la información posible sobre la visión que el entrevistado tiene de los requisitos y comprender los propósitos de la solución buscada.

- **Talleres:** La técnica de taller se utiliza en una reunión estructurada con el interés de obtener de esta manera una serie de requisitos bien definidos. Promueve la discusión entre los diversos mediadores. Una técnica que se utiliza en los talleres es una lluvia de ideas (34).

La captura de requisito utilizando los talleres consistió en realizar reuniones con los especialistas de SIPAC principalmente con la jefa del proyecto y el cliente funcional. Su objetivo fundamental es detallar cada requisito y sirve como base para la posterior especificación de los mismos.

2.4.2 Requisitos funcionales

Los requisitos funcionales especifican las funciones que un sistema o algún componente de software debe ser capaz de ejecutar (35). Permiten satisfacer las necesidades y exigencias del cliente. Para el desarrollo del producto se definieron los siguientes requisitos funcionales:

RF 1: Generar Resumen de cumplimiento del PTI (por sus siglas: Plan de trabajo individual)

RF 1.1: Obtener total de tareas planificadas en el PTI.

RF 1.1.1: Obtener total de tareas cumplidas en el PTI.

RF 1.1.2: Obtener total de tareas incumplidas en el PTI.

RF 1.2: Obtener total de tareas suspendidas o canceladas en el PTI.

RF 1.2.1: Obtener total de tareas nuevas (extraplanes) en el PTI.

RF 1.3: Listar las tareas incumplidas, quién las originó y causas en el PTI.

RF 1.4: Listar las tareas suspendidas o canceladas, quién las originó y causas en el PTI.

RF 1.5: Listar las nuevas tareas, quién las originó y causas en el PTI.

RF 2: Generar Resumen de cumplimiento del Plan mensual.

RF 2.1: Obtener actividades planificadas en el PM (por sus siglas: Plan mensual)

RF 2.1.1: Obtener subtotal de actividades del PAA (por sus siglas: Plan de anual de actividades) para el mes.

RF 2.1.2: Obtener total de actividades externas del PAA para el mes.

RF 2.1.3: Obtener % de actividades externas del PAA para el mes.

RF 2.1.4: Obtener total de actividades propias del PAA para el mes.

RF 2.1.5: Obtener % de actividades propias del PAA para el mes.

RF 2.1.6: Obtener subtotal de nuevas tareas incorporadas por puntualización para el mes.

RF 2.1.7: Obtener total de actividades externas por puntualización para el mes.

RF 2.1.8: Obtener % de actividades externas por puntualización para el mes.

RF 2.1.9: Obtener total de actividades propias por puntualización para el mes.

RF 2.1.10: Obtener % de actividades propias por puntualización para el mes.

RF 2.1.11: Obtener % de tareas incorporadas vs PAA para el mes.



RF 2.2: Obtener actividades cumplidas en el Plan mensual para el mes.

RF 2.2.1: Obtener total de actividades cumplidas del PAA para el mes.

RF 2.2.2: Obtener % de actividades cumplidas del PAA para el mes.

RF 2.2.3: Obtener total de actividades incumplidas del PAA para el mes.

RF 2.2.4: Obtener total de actividades pospuestas o suspendidas del PAA para el mes.

RF 2.2.5: Obtener total de actividades extraplanes del PAA para el mes.

RF 2.2.6: Obtener total de actividades externas por extraplanes para el mes.

RF 2.2.7: Obtener % de actividades externas por extraplanes para el mes.

RF 2.2.8: Obtener total de actividades propias por extraplanes para el mes.

RF 2.2.9: Obtener % de actividades propias por extraplanes para el mes.

RF 2.2.10: Obtener % de extraplanes vs tareas planificadas para el mes.

RF 3: Generar Resumen de cumplimiento del PAA

RF 3.1: Obtener actividades planificadas en el PAA para el año.

RF 3.1.1: Obtener subtotal de actividades del PAA para el año.

RF 3.1.2: Obtener total de actividades externas del PAA para el año.

RF 3.1.3: Obtener % de actividades externas del PAA para el año.

RF 3.1.4: Obtener total de actividades propias del PAA para el año.

RF 3.1.5: Obtener % de actividades propias del PAA para el año.

RF 3.1.6: Obtener subtotal de nuevas tareas incorporadas por puntualización para el año.

RF 3.1.7: Obtener total de actividades externas por puntualización para el año.

RF 3.1.8: Obtener % de actividades externas por puntualización para el año.

RF 3.1.9: Obtener total de actividades propias por puntualización para el año.

RF 3.1.10: Obtener % de actividades propias por puntualización para el año.

RF 3.1.11: Obtener % de tareas incorporadas vs PAA para el año.

RF 3.2: Obtener actividades cumplidas en el PAA para el año.

RF 3.2.1: Obtener total de actividades cumplidas del PAA para el año.

RF 3.2.2: Obtener % de actividades cumplidas del PAA para el año.

RF 3.2.3: Obtener total de actividades incumplidas del PAA para el año.

RF 3.2.4: Obtener total de actividades pospuestas o suspendidas del PAA para el año.

RF 3.2.5: Obtener total de actividades extraplanes del PAA para el año.

RF 3.2.6: Obtener total de actividades externas por extraplanes para el año.

RF 3.2.7: Obtener % de actividades externas por extraplanes para el año.

RF 3.2.8: Obtener total de actividades propias por extraplanes para el año.

RF 3.2.9: Obtener % de actividades propias por extraplanes para el año.

RF 3.2.10: Obtener % de extraplanes vs tareas planificadas para el año.

RF 4: Generar Desglose por ARC del PAA por trimestre.

RF 4.1: Obtener total de actividades por ARC del PAA para el año.

RF 4.2: Obtener resumen de actividades por ARC del PAA para cada trimestre en el año

RF 5: Obtener Caracterización del PAA.

RF 5.1: Obtener total de actividades del PAA.

RF 5.1.1: Obtener total de actividades principales del PAA.

RF 5.1.2: Obtener total de actividades de aseguramiento del PAA.

RF 5.2: Obtener actividades externas del PAA.

RF 5.1.1: Obtener total actividades externas del PAA.

RF 5.1.2: Obtener actividades de aseguramiento externas del PAA.

RF 5.3: Obtener actividades propias del PAA.

RF 5.1.1: Obtener total actividades propias del PAA.

RF 5.1.2: Obtener actividades propias de aseguramiento del PAA.

2.4.3 Especificación de requisitos funcionales

La especificación de los requisitos del software es una descripción completa del comportamiento del sistema a desarrollar. Incluye la descripción de todas las interacciones que se prevén que los usuarios tendrán con el software (36).

La tabla que se muestra a continuación, responde a la especificación del requisito funcional: Obtener Caracterización del PAA; las restantes especificaciones de los requisitos se encuentran explícitamente descritos en el expediente de proyecto en la dirección Exp_Proyecto/Ingeniería/Requisitos/Requisitos.

Tabla 2: Descripción del requisito: Obtener caracterización del PAA.

Precondiciones	El usuario encargado de Caracterización del PAA debe estar autenticado y tener elaborado un PAA.
Flujo de eventos	
Flujo básico Adicionar usuario	
	1. Selecciona en el menú principal la opción "Recuperaciones".
	2. De las opciones mostradas, seleccionar la opción "Reportes".



	3. Muestra una ventana en el panel izquierdo y se debe seleccionar “Plan Anual de actividades”.
	4. Selecciona en el menú central la opción “Caracterización del PAA”.
	5. Selecciona un Plan Anual de los que aparecen en la ventana derecha.
	6. Selecciona en la barra superior la opción Obtener Reporte.
	7. Escoge la opción exportar a PDF y guarda el documento.
	8. Concluye la ejecución del requisito.
Pos-condiciones	
	<ul style="list-style-type: none"> • Obtener o guardar el reporte de caracterización
Flujos alternativos	
Flujo alternativo: Datos Incompletos	
	Marcar el campo obligatorio que no ha sido llenado.
	Vuelve al paso 3 del flujo básico.
Pos-condiciones	
	No aplica.
Validaciones	
Conceptos	No aplica.
Requisitos especiales	No aplica.
Asuntos pendientes	No aplica.

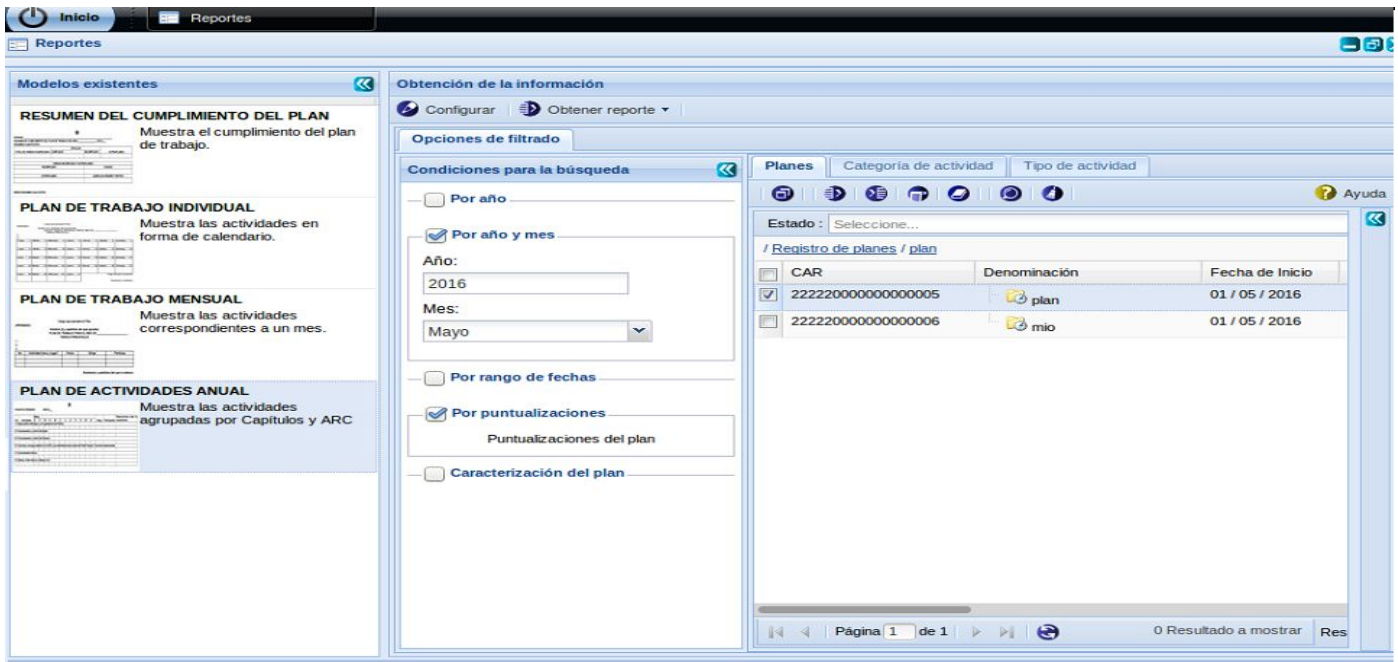


Figura 2: Prototipo para el requisito Obtener caracterización del PAA.

RESUMEN DEL PLAN ANUAL DE ACTIVIDADES PARA EL AÑO: 2016

ENTIDAD: MINISTRO SIPAC

Clasificación:
Ejemplar No.:

ACTIVIDADES Y TAREAS DE ASEGURAMIENTO PLANIFICADAS						
CAPITULOS	TOTAL		EXTERNAS		PROPIAS	
	ACTIVIDADES	TAREAS DE ASEGURAMIENTO	ACTIVIDADES	TAREAS DE ASEGURAMIENTO	ACTIVIDADES	TAREAS DE ASEGURAMIENTO
II	2				2	
TOTAL	2				2	

Figura 3: Salida del sistema para el RF: Caracterización del PAA.

2.4.4 Validación de los requisitos

La validación de requerimientos es importante debido a que los errores en el documento de requerimientos pueden conducir a importantes costos al repetir el trabajo cuando son descubiertos durante el desarrollo o después de que el sistema esté en uso (37).

Para la validación de los requisitos funcionales definidos, se utilizó la técnica construcción de prototipo de interfaz de usuario, con el uso de la herramienta *Visual Paradigm*.

Construcción de prototipos de interfaz de usuario: Es una técnica que consiste en construir herramientas basadas en la especificación de requerimientos. Los prototipos por lo general se construyen

a partir de un conjunto parcial de los requerimientos funcionales del sistema, pero permiten al usuario tener una idea global de la estructura de la interfaz de usuario del sistema. Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final. Ayuda al ingeniero de sistemas y al cliente a entender de mejor manera cuál será el resultado de la construcción cuando los requisitos estén satisfechos (38).

Mediante dicha técnica de validación se mostró al cliente un modelo ejecutable de las funcionalidades en cuestión, permitiéndole tener una visión preliminar y comprobando la satisfacción de sus necesidades iniciales. Seguidamente se presenta el prototipo de interfaz utilizado para el requisito funcional caracterización del PAA.

2.4.5 Requisitos no funcionales

Los requisitos no funcionales se refieren a aspectos técnicos que debe incluir el sistema y se relacionan con características de restricciones y de calidad que debe cumplir el software. Suelen ser más críticos que los requerimientos funcionales, dado que la ausencia de uno de los mismos puede inutilizar al sistema (35).

Para la funcionalidad no se definieron nuevos requisitos, se hizo uso de los existentes en el expediente de proyecto en la dirección Exp_Proyecto/Ingeniería/Requisitos/Modelado_del_negocio/CIG-SPA-N-i3514-RNF. A continuación se mencionan algunos:

Funcionalidad

Interoperabilidad

Breve descripción: Capacidad del producto de software para interactuar recíprocamente con uno o más sistemas especificados.

- La solución deberá exportar información relacionada con los elementos primarios de planificación en formato de hoja de cálculo (xlsx) y en un sistema ilegible para el usuario.

Usabilidad

Comprensibilidad

Breve descripción: Capacidad del producto de software para permitirle al usuario entender si el software es idóneo, y cómo puede usarse para las tareas y condiciones de uso particulares.

- Las etiquetas y los campos de cada interfaz tendrán títulos asociados a su función de negocio.

Idoneidad

Breve descripción: Capacidad del software para mantener un conjunto apropiado de funciones para las tareas y los objetivos del usuario especificados.

- La solución permitirá generar reportes estándares en los formatos de tablas que define la Instrucción No.1 Del Presidente de Los Consejos de Estado y de Ministros 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, en archivos de extensión PDF, DOC, XSL, HTML.

Operabilidad

Breve descripción: Capacidad del producto de software para permitirle al usuario operarlo y controlarlo.

- El orden de desplazamiento por campos en cada formulario del sistema siempre será de izquierda a derecha.

Portabilidad

Coexistencia

Breve descripción: Capacidad del producto de software de coexistir con otro software independiente en un ambiente común y compartir los recursos comunes.

- El sistema interactuará con visualizador de ficheros PDF para la presentación de los reportes que genere y los ficheros que importe.

2.5 Modelado de la solución

En este epígrafe se procede a modelar los artefactos y conceptos necesarios para obtener una mejor comprensión de la funcionalidad a desarrollar.

2.5.1 Diseño de la solución en términos de componentes

SIPAC es el sistema que se encarga de interrelacionar objetivos de trabajos con actividades en tiempo real, que facilita el registro y control de la planificación tanto a nivel de la entidad como a nivel de gobierno. La arquitectura del sistema presenta una estructura que responde a diferentes niveles de empaquetamiento, como: subsistema, componente, funcionalidad y requisitos (40).

Subsistema Planificación: permite el registro, seguimiento y control de la Planeación Estratégica y Operativa para todos los niveles organizacionales. Dicho subsistema está constituido por los siguientes componentes:

- **Configuración:** Permite configurar y gestionar usuarios y grupos de usuarios, permisos y nomencladores del sistema.
- **Planeación:** Contiene las operaciones a realizar con los documentos de la planificación, entiéndase: planes, objetivos, criterios de medidas, actividades, áreas de resultados clave (ARC) y factores que intervienen en la planificación (FIP).
- **Notificaciones:** mediante el cual es posible generar además de las notificaciones, a partir de determinadas acciones que se realizan sobre los diferentes elementos, los reportes de acciones realizadas sobre elementos a los cuales el usuario tiene acceso.

Subsistema Configuración: brinda una serie de funcionalidades que deben ser ejecutadas antes de comenzar a utilizar el resto de los subsistemas, específicamente para el funcionamiento del sistema SIPAC es necesario definir el flujo de información de los elementos de la planificación a partir de estados de los documentos y las transiciones, el cual es posible a través del componente: WorkFlow.

A continuación se muestra el diagrama de componente general para SIPAC.

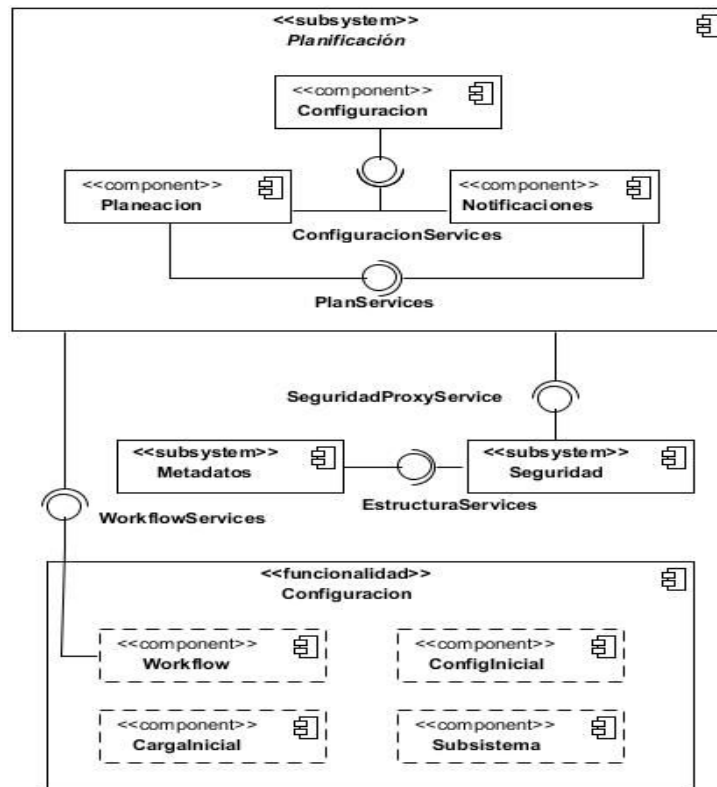


Figura 5: Mapa general de componentes

El modelo de componentes de la solución para la caracterización de la elaboración y ejecución del PAA se va a encontrar en el subsistema de Planificación y a su vez en el componente Planeación, el cual contiene un conjunto de funcionalidades verticales que describen la interrelación existente entre los elementos de la planificación y la funcionalidad propuesta, encontrándose específicamente en el módulo Recuperaciones/Reportes.

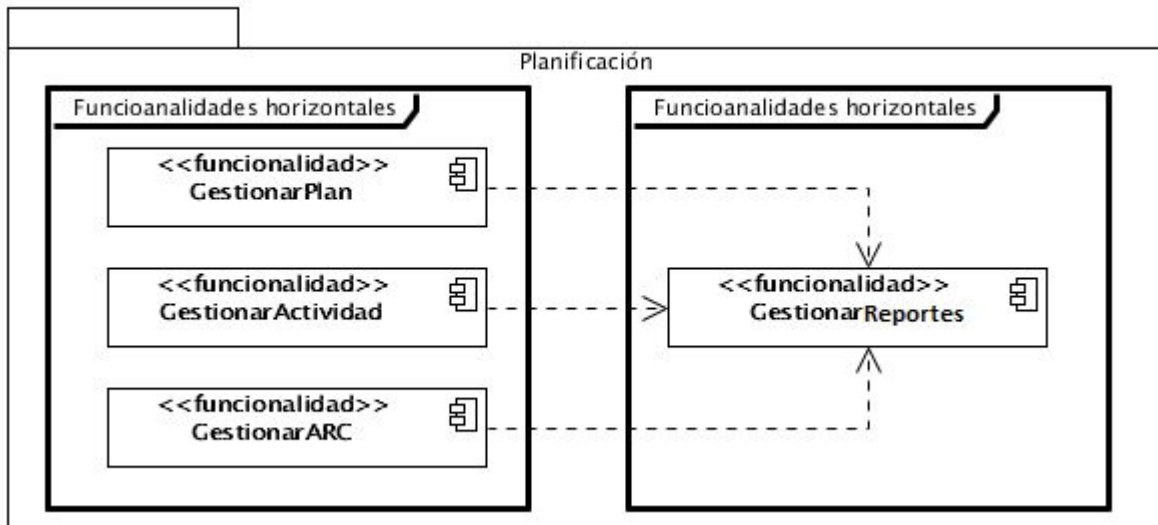


Figura 6: Diagrama de componente para la funcionalidad propuesta.

2.5.2 Diagrama de clase del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. En la figura que se muestra a continuación se presenta el diagrama de clases del diseño para la solución propuesta, el cual está basado en estereotipos web y en el patrón arquitectónico MVC.

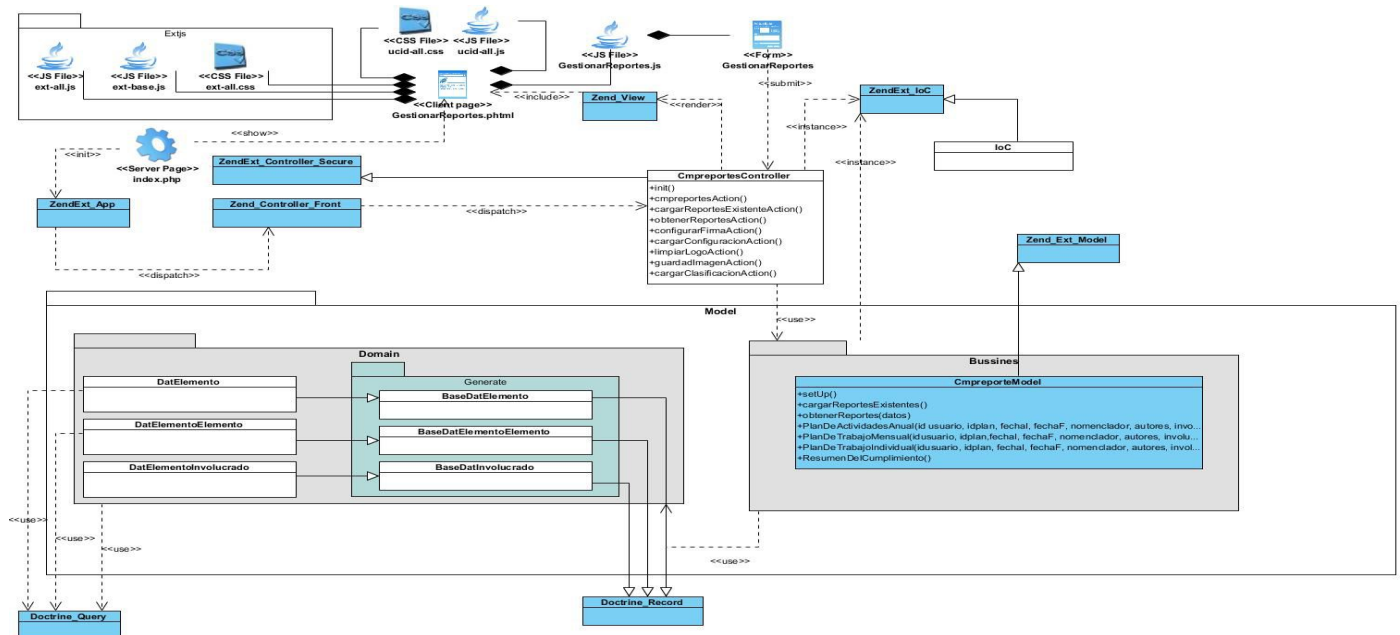


Figura 7: Diagrama de clase para el RF obtener caracterización del PAA.

Tabla 3: Descripción de las clases del modelo de diseño.

Clases	Descripción
cmpreportes.phtml	Página cliente de la funcionalidad, encargada de incluir las clases js y css a visualizar.
cmpreportes.css	Contiene las clases para darle los estilos gráficos a la solución, separando de esta forma los estilos del contenido.
cmpreporstes.js	Contiene las funciones y variables globales que se ejecutan llamando a sus funciones desde cualquier clase sin tener que incluirlas en cada una de ellas.
CmpreportesController	Clase encargada de controlar la comunicación entre la Vista y el Modelo.
ZendExt_Controller_Secure	Encargada de gestionar acciones personalizadas y está integrada a la seguridad.
Zend_Ext_Model	Modelo gestor de negocio que permite entre otras funcionalidades iniciar la conexión a la base de datos.
Extjs	Contiene los componentes generados a través de la biblioteca JavaScript Extjs.

Paquete Model	Encargado de manejar los datos persistentes dentro del componente. Contiene el Bussines y el Domain.
Paquete Bussines	Encargado de manejar toda la lógica de negocio.
Paquete Domain	Contiene los métodos para manejar el negocio.

2.6 Patrón

Un patrón es una solución a un problema en un contexto particular, recurrente (lo que hace la solución relevante a otra situaciones) que enseña (permite entender cómo adaptarlo a la variante particular del problema donde se quiere aplicar) y tiene un nombre para referirse al patrón. Los patrones facilitan la reutilización de diseños y arquitecturas software que han tenido éxito (39).

Describe una estructura de diseño que resuelve un problema en particular dentro de un contexto específico.

2.6.1 Patrón arquitectónico

Un patrón de arquitectura encapsula los elementos y las relaciones que existen entre ellos permitiendo abstraer su comportamiento para que sea posible tener una configuración de componentes que satisfaga ciertas necesidades. Existen una gran variedad de patrones arquitectónicos y cada uno de ellos está pensado para un determinado dominio. Existen patrones para sistemas distribuidos, para sistemas en capas, para sistemas basados en componentes, etc. y cada uno de ellos garantiza resolver un determinado problema utilizando una determinada configuración de los componentes (40).

Patrón Modelo-Vista-Controlador

El patrón MVC (Model-View-Controller) es un patrón de arquitectura de las aplicaciones software. Separa la lógica de negocio de la interfaz de usuario. Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores, o lo que es lo mismo. (41)

Es un patrón de diseño de arquitectura de software usado principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas donde se requiere una mejor separación de conceptos para que el desarrollo esté estructurado de una mejor manera, facilitando la programación en diferentes capas de manera paralela e independiente. MVC sugiere la separación del software en 3 estratos: Modelo, Vista y Controlador, los cuales serán explicados en breve (42):

Modelo: Es la representación de la información que maneja la aplicación. El modelo en sí son los datos puros que puestos en contexto del sistema proveen de información al usuario o a la aplicación misma.

Vista: Es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación Web, la “Vista” es una página HTML con contenido dinámico sobre el cuál el usuario puede realizar operaciones.

Controlador: Es la capa encargada de manejar y responder las solicitudes del usuario, procesando la información en caso de ser

2.6.2 Patrones de diseño

Los patrones brindan una solución generalmente ya probada y documentada a problemas que se dan durante el proceso de desarrollo de software. Facilitan el trabajo al emplear un conjunto de buenas prácticas, además de presentar la ventaja de ser genéricos, al no depender de ningún lenguaje. Los patrones se dividen fundamentalmente en patrones arquitectónicos y de diseño (43).

Los patrones de diseño especifican la experiencia de diseñadores expertos, ayudando a los desarrolladores de software en su documentación y transferencia de conocimientos, proporcionándoles un vocabulario común (44).

Patrón GRASP

Los patrones GRASP (*General Responsibility Assignment Software Patterns*) son parejas de problema solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (43).

Con el objetivo de obtener un sistema flexible y reusable, se pusieron en práctica los siguientes patrones GRASP en el diseño de la funcionalidad:

- **Experto (en Información):** Experto en información nos dice que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo o ejecutarlo, de este modo obtendremos un diseño con mayor cohesión y cuya información se mantiene encapsulada, es decir, disminuye el acoplamiento. El uso de este patrón se evidencia en la clase: DatElemento.
- **Creador:** nos ayuda a identificar quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases. Este patrón como su nombre lo indica es el que crea, guía la asignación



de responsabilidades relacionadas con la creación de objetos, se evidencia este patrón en la clase CmpReporteController la cual se encarga de crear las instancias de la clase CmpReporteModel, para usar las funcionalidades de ésta.

- **Alta Cohesión:** El uso de este patrón indica que la información almacenada en las clases debe ser coherente y relacionada a lo que se maneja en dicha clase. Su uso se evidencia en todas las clases.
- **Bajo Acoplamiento:** Es la idea de tener las clases lo menos ligadas entre sí que se pueda, de tal forma que, en caso de producirse una modificación en alguna de ellas, tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización y disminuyendo la dependencia entre las clases. Este patrón es un principio que asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.).
- **Controlador:** El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es el controlador quien recibe los datos del usuario y quien los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocio debe estar separada de la capa de presentación, lo que aumenta la reutilización de código y permite a la vez tener un mayor control (45). Es un evento generado por actores externos. Se asocian con operaciones del sistema, operaciones del sistema como respuestas a los eventos del sistema, tal como se relacionan los mensajes y los métodos. El patrón Controlador se ve evidenciado en la funcionalidad en la clase CmpReporteController.

Patrón GOF

Los patrones GOF (patrones de la pandilla de los cuatros): describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos y otros ámbitos referentes al diseño de interacción o interfaces (46).

En la funcionalidad desarrollada fueron usados los siguientes patrones GOF:

- **Fachada (Estructural):** Este patrón proporciona una interfaz unificada de alto nivel para un subsistema, que oculta las interfaces de bajo nivel de las clases que lo implementan simplificando así la interacción con el subsistema. Se utiliza para proporcionar un fácil acceso a subsistemas

complejos. En el diseño de la solución que se propone, la clase que se utiliza como fachada es *ZendExt_IoC*, para acceder a los servicios de otros componentes.

- **Cadena de Responsabilidad (Comportamiento):** Este patrón permite establecer la línea que deben llevar los mensajes para que los objetos realicen la tarea indicada. Se evidencia al distribuir las responsabilidades, ya que ante la ocurrencia de un error al realizarse una determinada consulta a la base de datos el mismo es 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.

2.7 Modelo de base datos

Un modelo de datos es básicamente una "descripción" de un contenedor de datos (donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos son abstracciones que permiten la implementación de un sistema eficiente de base de datos (47).

Con el análisis del modelo de dato de SIPAC se demostró que los datos referentes a la caracterización de la elaboración y ejecución del PAA pueden ser almacenados en las tablas ya existentes, por tanto, no es necesaria la creación de otras nuevas. El modelo de base de datos para SIPAC cuenta con 46 tablas de estas se utilizarán 14 tablas.

A continuación, se muestra el modelo de base de datos correspondiente a la caracterización de la elaboración y ejecución del PAA:

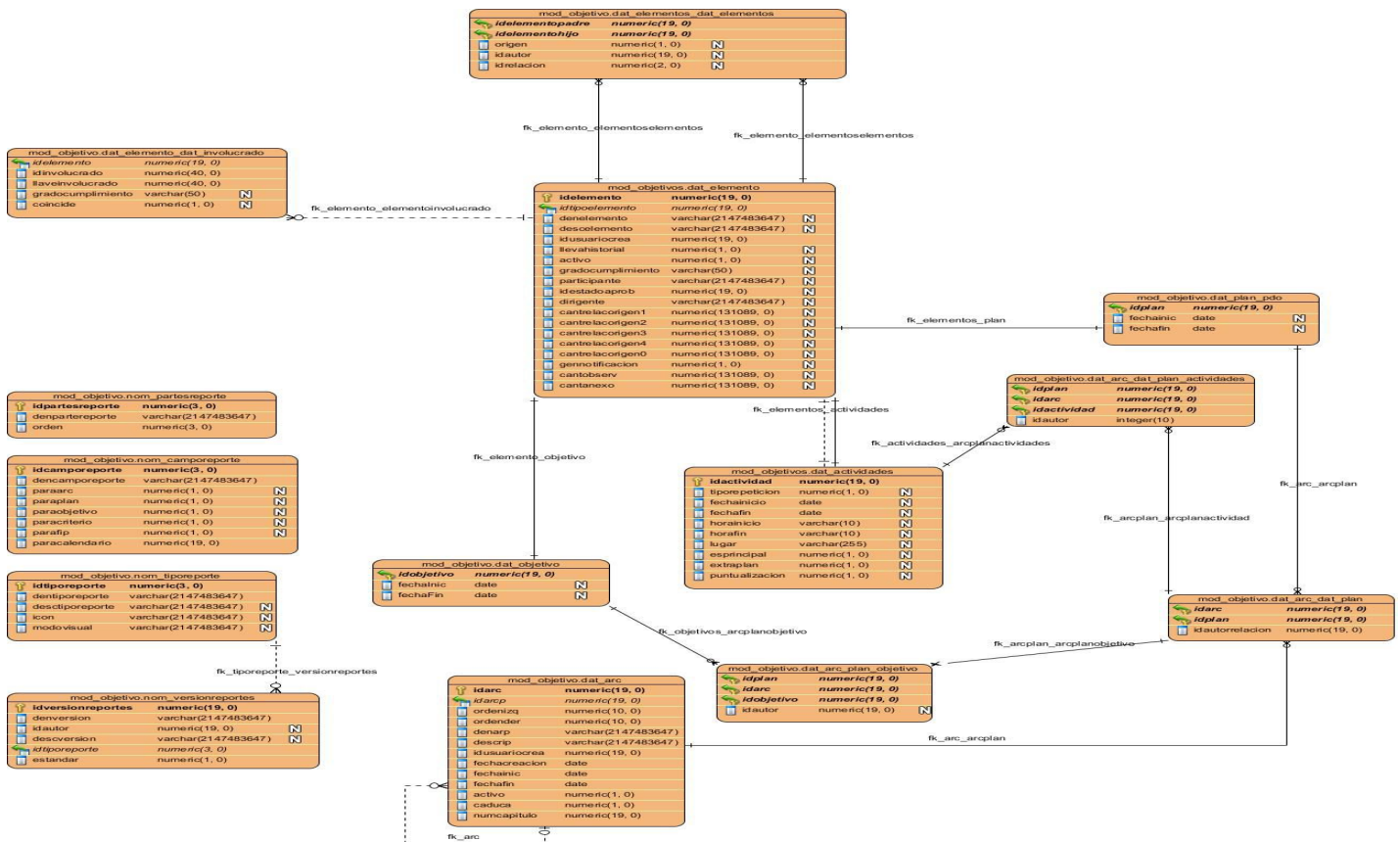


Figura 8: Modelo de datos para la Caracterización de la elaboración y ejecución del PAA.

El modelo de datos de la solución va estar caracterizado por la entidad DatElementos, la cual se encarga de almacenar toda la información relacionada con los elementos primarios de la planificación, representados por las entidades DatPlanPdo y DatActividades; estos estarán relacionados directamente a través de una relación de 1 a muchos (*). Las Áreas de Resultados Claves (ARC) estarán representadas por la entidad DatArc y esta a su vez tendrá una relación con los planes, objetivos y actividades; pudiendo cada uno de estos elementos estar relacionados a varias ARC. Los elementos de la planificación pueden tener relacionados varios involucrados asociados a la vez, estos son representados por la entidad DatInvolucrados. La entidad DatElementosDatElementos también estará asociada con DatElemnto donde se observará la relación que existe entre cada elemento de la planificación. Los reportes están representados por la entidad NomTipoReportes que a su vez está relacionada con NomVersionReporte.

2.8 Conclusiones Parciales

Con el desarrollo del presente capítulo se definieron los principales conceptos y nueve reglas del negocio asociados a la caracterización de la elaboración y ejecución del PAA. Se llevó a cabo la captura, descripción y validación de 70 requisitos funcionales y diez no funcionales, lo que posibilitó un mayor conocimiento sobre los principales requerimientos que debe cumplir la solución. El diseño de las clases permitió una mejor comprensión de cómo están estructuradas las clases y la relación entre ellas. Se definieron los patrones de diseño necesarios para el desarrollo posibilitó una entrada apropiada como punto de partida a las actividades de implementación.

CAPÍTULO 3: IMPLMNTACIÓN Y VALIDACIÓN

En el presente capítulo se explican los estándares de codificación por los que se rige el código fuente. Se muestran las salidas del sistema, es decir, los reportes que se obtienen para la caracterización del PAA, resúmenes de cumplimiento para el plan individual, mensual y anual y para el desglose por ARC. Se realiza la validación del diseño propuesto mediante métricas. Además, se valida la funcionalidad a través de métodos de prueba de caja blanca y caja negra, con el propósito de obtener un software con la calidad que se requiere. Finalmente se realiza el análisis de los resultados obtenidos y el impacto de la funcionalidad.

3.1 Implementación de la solución

La implementación de la funcionalidad para la caracterización de la elaboración y ejecución del PAA proporcionará como resultado un producto que cumpla las exigencias y necesidades existentes en SIPAC.

3.1.1 Estándares de codificación

La calidad del código en la programación se mide por varios aspectos: la estabilidad, la tolerancia a los fallos y la reusabilidad son algunos de ellos. Pero otros aspectos importantes a la hora de medir su calidad son la legibilidad y la posibilidad de incorporarle nuevas sentencias. Para poder conseguir un resultado con estas dos últimas características es necesario establecer un conjunto de reglas a la hora de escribir el código del programa, es decir: utilizar un estándar. Los estándares de codificación permiten establecer reglas sobre la escritura del código fuente, permitiendo seguir un estilo de programación homogéneo de forma tal que todos los participantes de un proyecto puedan entender el mismo y el código, en consecuencia, sea mantenible (48).

A continuación, se definen las tres partes fundamentales dentro de un estándar de programación:

- **Convención de nomenclatura:** Define cómo nombrar variables, funciones y clases.
- **Convenciones de legibilidad de código:** Es la forma de organizar el código y lograr que independientemente de quien desarrolle se entienda como un todo.
- **Convenciones de documentación:** Define cómo establecer comentarios, archivos de ayuda, entre otros.



3.1.2 Nomenclatura de clases

Los nombres de las clases deben comenzar con mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación *PascalCasing* el cual define que la nomenclatura está compuesta por tantas palabras como sean necesarias pero que la primera letra de cada una de las palabras irá siempre en mayúsculas obviando el uso de artículos.

Ejemplo: DatActividades

3.1.3 Nomenclatura según el tipo de clases

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

Ejemplo: CmpReportesController.

Clases de los modelos:

- Business (Negocio): Las clases que se encuentran dentro de Business después del nombre llevan la palabra: “Model”.

Ejemplo: CmpReportesModell.

- 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: DatArcDatPlanActividades.
- Generated (Dominio base): Las clases que se encuentran dentro de *Generated* el nombre comienza con la palabra: “Base” y seguido el nombre de la tabla en la base de datos. Ejemplo: BaseDatActividades.

3.1.4 Nomenclatura de las funcionalidades y atributos

El nombre a emplear para las funciones y los atributos se escriben con la inicial del identificador en minúscula, en caso de que sea un nombre compuesto se empleará la notación *PascalCasing* similar a la antes mencionada, con la excepción de la primera letra. Las funcionalidades de las clases controladoras se les asignará un nombre y seguidamente la palabra *Action*.

Ejemplo:

Ejemplo: obtenerReporteAction().

Ejemplo de atributo: \$datos.

3.1.5 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.

3.2 Salidas de la funcionalidad desarrollada

A continuación se presentan las salidas del sistema que dan respuesta al problema planteado. Las demás salidas se encuentran en el expediente de proyecto en la dirección Exp_Proyecto/Ingeniería/Requisitos/Requisitos/Salidas del sistema.

RESUMEN DEL CUMPLIMIENTO DEL PLAN INDIVIDUAL DEL MES ENERO

CARGO Y NOMBRE Y APELLIDOS: -

TOTAL DE PLANIFICADAS	DE ELLAS		POSPUESTAS (OTRO MES) SUSPENDIDAS	NUEVAS TAREAS (EXTRA PLANES)
	CUMPLIDAS	INCUMPLIDAS		
0	0	0	0	0

OBSERVACIONES DEL CUMPLIMIENTO	QUIEN LAS ORIGINO	CAUSAS
TAREAS INCUMPLIDAS		
SUSPENDIDAS O POSPUESTAS		
NUEVAS TAREAS		

ANALISIS CUALITATIVO

Figura 9: Salida para el RF Generar Resumen de cumplimiento del PTI



MINISTRO SIPAC

RESUMEN DEL CUMPLIMIENTO DEL PLAN DE TRABAJO DEL MES ENERO DE 2016

ENTIDAD:

TAREAS PLANIFICADAS												
TOTAL TAREAS DEL PLAN ANUAL		DEL PLAN ANUAL PARA EL MES					NUEVAS INCORPORADAS EN LA PUNTUALIZACION MENSUAL					%INCORPORADAS vs PLAN ANUAL
		SUB TOTAL	EXTERNAS (Nivel superior o igual)	%	PROPIAS	%	SUB TOTAL	EXTERNAS (Nivel superior o igual)	%	PROPIAS	%	
MES DE	0	0	0	0	0	0	0	0	0	0	0	0

TAREAS CUMPLIDAS												
TOTAL TAREAS CUMPLIDAS EN EL MES		PLANIFICADAS EN EL PLAN MENSUAL (incluye las del Plan Anual y las puntualizadas)					EXTRAPLANES					
		PLANIFICADAS	CUMPLIDAS	%	INCUMPLIDAS	POSPUESTAS O SUSPENDIDAS	TOTAL	EXTERNAS (Nivel superior o igual)	%	PROPIAS	%	%
MES DE		0	0	0	0	0	0	0	0	0	0	0

Figura 10: Salida para el RF Generar Resumen de cumplimiento del PM y PAA

3.3 Validación de la solución propuesta

Para detectar y corregir los posibles errores existentes en la funcionalidad implementada se detalla a continuación la validación de la misma y se describen los métodos utilizados, así como las métricas para la validación del diseño propuesto.

3.3.1 Validación del diseño mediante métricas

Las métricas de diseño son las que permiten medir de forma cuantitativa la calidad de los atributos internos del software. Esto permite al ingeniero evaluar la calidad durante el desarrollo del sistema (49).

Para validar las clases del diseño se utilizaron las métricas: Tamaño operacional de clases (TOC) y Relación entre clases (RC). Estas métricas fueron diseñadas para validar 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:** grado de dificultad que tiene implementar un diseño de clases determinado.



- **Reutilización:** grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.
- **Acoplamiento:** 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:** grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software.
- **Cantidad de pruebas:** número o grado de esfuerzo para realizar las pruebas de calidad (unidad) del producto (componente, modulo, clase, conjunto de clases, etc.) diseñado.

Tamaño operacional de clases (TOC)

Está dado por el número de métodos asignados a una clase. Los atributos de calidad que utilizan para evaluar el diseño son los tres primeros mencionados anteriormente. En la aplicación de esta métrica se tuvieron en cuenta las clases del Modelo y del *Domain*, así como la clase Controladora de acuerdo al diseño de la funcionalidad desarrollada. Se tuvo en cuenta, además, la cantidad de procedimientos que contienen cada una de las clases, obteniendo los siguientes resultados:

Tabla 4: Tamaño operacional de clase (TOC).

Atributo	Categorías	Criterios
Responsabilidad	Baja	\leq Prom
	Media	Entre Prom. y 2^* Prom
	Alta	$> 2^*$ Prom
Complejidad de implementación	Baja	\leq Prom
	Media	Entre Prom. y 2^* Prom
	Alta	$> 2^*$ Prom
Reutilización	Baja	$> 2^*$ Prom
	Media	Entre Prom. y 2^* Prom
	Alta	\leq Prom

El resultado de la métrica TOC en cuanto a los parámetros de calidad: Responsabilidad, Complejidad de Implementación y Reutilización se encuentran mostradas en el **Anexo 2**.

La figura 10 muestra la representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad y Complejidad de implementación. Este gráfico muestra un resultado satisfactorio para los dos primeros atributos, pues el 43% de las clases poseen una baja responsabilidad y complejidad de implementación, lo que permite que en caso de fallos, ninguna clase es demasiado crítica como para dejar al sistema fuera de servicio y como la complejidad también es de 43% se hace más fácil mejorar el mantenimiento y el soporte de las clases respectivamente.

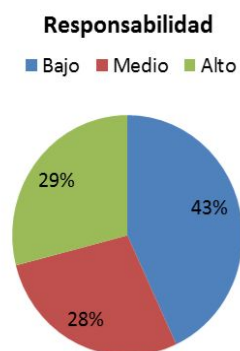


Figura 11: Resultados de la evaluación de la métrica TOC para Responsabilidad y Complejidad de implementación.

La figura 11 muestra la representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización. Demostrado que el diseño de la funcionalidad es eficiente pues solamente el 28% del total de las clases poseen una baja reutilización.

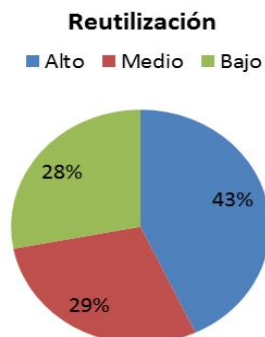


Figura 12: Resultados de la evaluación de la métrica TOC para Reutilización

Relación entre clases (RC)

Está dado por el número de relaciones de una clase con otra. Dicha métrica está determinada por los atributos Acoplamiento, Complejidad de mantenimiento, Reutilización y Cantidad de pruebas, existiendo una relación directa con los tres primeros e inversa con el último antes mencionado. Al aplicarse la métrica Relación entre clases se obtuvo los siguientes resultados:

Tabla 5: Relación entre clases (RC).

Atributo	Categorías	Criterios
Acoplamiento	Baja	RC=1
	Media	RC=2
	Alta	RC>2
Complejidad de mantenimiento	Baja	\leq Prom
	Media	Entre Prom. y 2*Prom
	Alta	$> 2*Prom$
Cantidad de pruebas	Baja	\leq Prom
	Media	Entre Prom. y 2*Prom
	Alta	$> 2*Prom$
Reutilización	Baja	$> 2*Prom$
	Media	Entre Prom. y 2*Prom
	Alta	\leq Prom

El resultado de la métrica RC en cuanto a los parámetros de calidad: Responsabilidad, Complejidad de Implementación y Reutilización se encuentran mostradas en el **Anexo 2**.

La figura 12 muestra la representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento. Se evidencia un acoplamiento favorable entre las clases, pues el 58% de las clases presentan dependencia. Este resultado es muy favorable para el diseño de la funcionalidad pues al existir poca dependencia entre las clases aumenta el grado de reutilización de la misma.

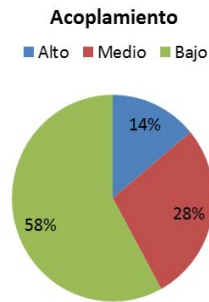


Figura 13: Resultados de la evaluación de la métrica RC para Acoplamiento.

La figura 13 muestra la representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de mantenimiento y Cantidad de pruebas. El gráfico refleja resultados aceptables de los atributo, pues el 85% de las clases presentan una baja complejidad de mantenimiento.

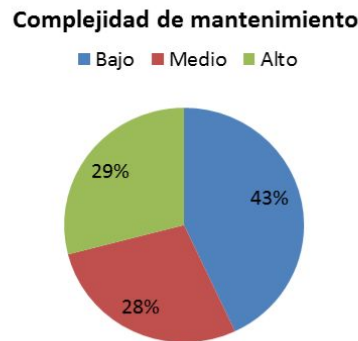


Figura 14: Resultados de la evaluación de la métrica RC para Complejidad de mantenimiento y Cantidad de pruebas.

La figura 14 muestra la representación de la incidencia de los resultados de la evaluación del atributo reutilización. Esto evidencia que el 85% de las clases poseen una alta reutilización lo que es un factor fundamental que debe ser tenido en cuenta en el desarrollo de software.

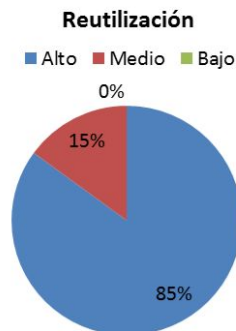


Figura 15: Resultados de la evaluación de la métrica RC para Reutilización.

3.4 Pruebas de software

Las pruebas de software son los procesos que permiten verificar y revelar la calidad de un producto antes de su puesta en marcha. Básicamente es una fase del proceso de desarrollo de software que consiste en probar las aplicaciones construidas. Son usadas con el fin de detectar posibles errores de implementación, calidad o usabilidad de un programa (50).

Los niveles de pruebas que se pueden aplicar a un software son descritos a continuación:

- **Pruebas de unitarias:** son usadas para comprobar el correcto funcionamiento de un módulo de código. Con esta prueba se comprueba que cada módulo funcione correctamente por separado.
- **Pruebas de integración:** en estas pruebas se combinan módulos individuales de forma tal que seas probados como un grupo. Se usan para detectar errores de interfaces y relaciones entre componentes.
- **Pruebas de sistema:** con estas pruebas se trata de asegurar la correcta navegación dentro del sistema, ingreso, procesamiento y recuperación de datos. Son usadas para detectar fallas en el cubrimiento de los requisitos.
- **Pruebas de aceptación:** son desarrolladas y ejecutadas con el cliente para determinar si el producto satisface sus criterios o requisitos. Estas son las únicas pruebas que se realizan directamente con el cliente, a diferencia de las restantes.



Para la funcionalidad las pruebas que se realizarán son las de aceptación con el objetivo de verificar que el software está listo y que puede ser usado por usuarios finales. Las pruebas unitarias para validar la unidad del código (una clase, función o método), las pruebas del sistema y las de aceptación.

3.4.1 Métodos de prueba

Un método de prueba nos es más que un procedimiento definitivo que produce un resultado de una prueba. Al aplicarle pruebas a un determinado producto se pueden detectar diversos tipos de errores al usar conjuntos de casos de pruebas.

Las pruebas usadas para asegurar la calidad de la funcionalidad y determinar cómo y en qué sentido el producto final cumple con los intereses del cliente, son las pruebas de caja blanca y caja negra, para ello se transan como objetivos los siguientes:

- Verificar la implementación de la solución.
- Verificar la integración adecuada de la solución.
- Verificar que todos los requisitos se han implementado correctamente.
- Identificar los errores y asegurar que estos sean corregidos.

Descripción y aplicación de la Prueba de Caja Blanca o Estructural

Se denomina pruebas de caja blanca a las pruebas de software que se realizan para verificar la correcta implementación de las unidades internas, las estructuras y sus relaciones, haciendo énfasis en la reducción de errores internos (51).

Las técnicas usadas para ejecutar dichas pruebas son las descritas a continuación:

- **Camino básico:** permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico. Los casos de prueba derivados del conjunto básico garantizan que durante la prueba se ejecute por lo menos una vez cada sentencia del programa.
- **Prueba de condición:** ejercita las condiciones lógicas contenidas en el módulo de un programa. Garantiza la ejecución por lo menos una vez de todos los caminos independientes de cada módulo, programa o método.



- **Prueba de Flujo de Datos:** se seleccionan caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa. Garantiza que se ejerciten las estructuras internas de datos para asegurar su validez.
- **Prueba de Bucles:** se centra exclusivamente en la validez de las construcciones de bucles. Garantiza la ejecución todos los bucles en sus límites operacionales.

A partir de la necesidad de crear un producto de alta calidad que cumpla con los requisitos deseados por el cliente es preciso valorar qué tan certera ha sido la implementación de la funcionalidad para la caracterización de la elaboración y ejecución del PAA y para ello es necesario aplicar una de las técnicas para la prueba de caja blanca. La cual comprende, en este caso la del **camino básico**.

Para ello, es necesario conocer el número de caminos independientes de un determinado algoritmo mediante el cálculo de la complejidad ciclomática, comenzando por un análisis del código, posteriormente son enumeradas cada una de las instrucciones, se construye el grafo de flujo asociado y mediante las fórmulas pertinentes se calcula dicha complejidad.

De esta manera se analizan y enumeran las sentencias de código de uno de los procedimientos contenidos en la clase DatElemento.

La función a probar fue seleccionada por la importancia que tiene en el desarrollo de la funcionalidad. Esta se encarga de clasificar dentro del conjunto de tareas puntualizadas cuales son externas y cuales propias. Para dar cumplimiento a los anteriores pasos básicos se enumeran cada una de las sentencias de código del procedimiento seleccionado:



```

public function Tareas_Externas_Propias($arrayTareas, $reupEntidad) {
    $tabla = array(); // 1

    if ($arrayTareas) { // 2

        $i = 0; // 3
        $j = 0; // 3
        foreach ($arrayTareas as $key => $tarea) { // 4
            $idElemento = $tarea[idelemento]; // 5
            $connect = Doctrine_Manager::connection(); // 5
            $cadena = "SELECT r.codigo_reup "; // 5
            $cadena .= "FROM mod_objetivos.dat_elementos de
                LEFT JOIN mod_objetivos.dat_elementos_nom_nomenclador denm
                ON de.idelemento = denm.idelemento
                INNER JOIN mod_objetivos.nom_reup r using(idnomenclador)"; // 5
            $cadena .= " WHERE de.idelemento = '$idElemento'"; // 5
            $res = $connect->execute($cadena); // 5
            $result = $res->fetchAll(PDO::FETCH_ASSOC); // 5

            $reupActividad = $result[0][codigo_reup]; // 5

            if ($reupEntidad == $reupActividad) { // 6
                $tabla['tareas_propias'][$i++] = $tarea; // 7
            } else {
                $tabla['tareas_externas'][$j++] = $tarea; // 8
            }
        }
    } else {
        return null; //9
    }
    return $tabla; //10
}

```

Figura 16: Código a probar

Luego del paso anterior, se representa el grafo de flujo asociado al código antes presentado a través de nodos, aristas y regiones donde:



Nodo: Representa una o más secuencias del procedimiento, un nodo en sí puede representar un proceso, una secuencia de procesos o una sentencia de decisión. Los nodos que no están asociados se utilizan al inicio y final del grafo.



Arista: Saeta a través de la cual se unen los Nodos y constituye el flujo de control del procedimiento.

RGC

Las regiones son las áreas delimitadas por las aristas y nodos.

Regiones:

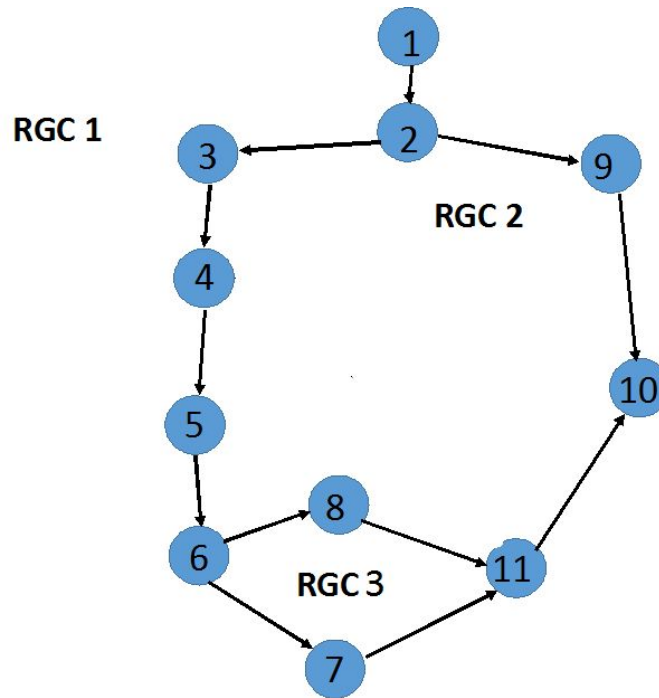


Figura 17: Grafo de Flujo

Una vez construido el grafo de flujo asociado al procedimiento anterior se determina la complejidad ciclomática ($V(G)$), el cálculo es necesario efectuarlo mediante tres vías o fórmulas de manera tal que quede justificado el resultado, siendo el mismo en cada caso:

Complejidad ciclomática

- Según (A) Aristas y (N) los Nodos.

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

$$V(G) = 12 - 11 + 2$$

$$V(G) = 3$$

- Según (P) Nodos Predicados (nodos de donde parten más de 2 aristas).

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

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

$$V(G) = 3$$

- Según áreas cerradas (R), se incluye el área externa del grafo

$$V(G) = R$$

$$V(G) = 3$$

El cálculo efectuado para cada caso es $V(G) = 3$, demostrando la existencia de 3 casos de pruebas a elaborar, lo que coincide con la cantidad de caminos básicos.

Caminos básicos

- a) 1 – 2 – 3 – 4 - 5 – 6 – 7 – 11 - 10
- b) 1 – 2 – 3 – 4 - 5 – 6 – 8 – 11 - 10
- c) 1 – 2 – 9 - 10

Se procede a realizar los casos de pruebas para cada uno de los caminos básicos determinados en el grafo de flujo.

Casos de pruebas

Se expondrán a continuación los casos de pruebas correspondientes a los dos primeros caminos, el caso de prueba del tercero se podrá ver en el **Anexo 3**

Tabla 6: Caso de prueba para el camino a).

Camino básico a): 1-2-3-4-5-6-7-11-10	
Descripción	Se envía cuando se quiere clasificar en tareas propias o externas un listado de taras puntualizadas.
Condición de ejecución	Tener un listado de las tareas planificadas y mediante una consulta de base dato obtener el código reup. Si este código esta dentro de su idactividad la adiciona en la lista de tareas propias
Entrada	Tener una lista de tareas puntualizadas.
Resultado esperado	Obtener un listado con las tareas propias dentro del listado de puntualizadas.

Tabla 7: Caso de prueba para el camino b).

Camino básico a): 1-2-3-4-5-6-8-11-10	
Descripción	Se envía cuando se quiere clasificar en tareas propias o externas un listado de taras puntualizadas.

Condición de ejecución	Tener un listado de las tareas planificadas y mediante una consulta de base dato obtener el código reup. Si este código esta dentro de su idactividad la adiciona en la lista de tareas externas.
Entrada	Tener una lista de tareas puntualizadas.
Resultado esperado	Obtener un listado con las tareas externas dentro del listado de puntualizadas.

Una vez realizados todos los casos de pruebas correspondientes a cada camino básico identificado en el grafo de flujo anterior, se verificó el correcto funcionamiento del método Tareas_Externas_Propias. Donde no se mostró ningún error para el procedimiento seleccionado, evidenciándose una ejecución satisfactoria.

Descripción y aplicación de la Prueba de Caja Negra o Funcional

Estas pruebas permiten verificar el correcto manejo de funciones externas soportadas por el software, así como que el comportamiento observado se apegue a las especificaciones del producto y a las expectativas del cliente (52).

En esencia permite encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a la base de datos externa.
- Errores de rendimiento.
- Errores de inicialización y fin.

Para desarrollar la prueba de caja negra existen técnicas las cuales se describen a continuación:

- **Partición de Equivalencia:** esta técnica 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.

A continuación se aplica la prueba de partición de equivalencia como parte de la prueba de caja negra realizada al RF: Caracterización del PAA. Para ello se definió un conjunto de estados válidos y no válidos para las condiciones de entrada del sistema.

La **partición de equivalencia** 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 (53).

Tabla 8: Descripción del caso de prueba para el requisito Caracterización del PAA.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Caracterización de PAA.	El sistema debe permitir visualizar al usuario el informe de Caracterización del Plan.	EP 1.1: Seleccionar un tipo de reporte, fecha, la opción de caracterización, un plan y el botón Obtener reporte.	<ul style="list-style-type: none"> El usuario selecciona el reporte Plan anual de actividades. El sistema activa un menú central para seleccionar las condiciones de búsquedas. El usuario selecciona la condición de búsqueda que sería una fecha y la caracterización. El usuario selecciona un plan y la opción Obtener reporte. El sistema despliega un menú para el formato, se selecciona Formato PDF y se muestra el informe.
		EP 1.2: Seleccionar un reporte, la opción de caracterización, un plan y el botón Obtener reporte.	<ul style="list-style-type: none"> El usuario selecciona el reporte Plan anual de actividades. El sistema activa un menú central para seleccionar las condiciones de búsquedas. El usuario selecciona la condición de búsqueda caracterización del PAA. El usuario selecciona un plan y la opción Obtener reporte. El sistema despliega un menú para el formato, se selecciona Formato PDF y se muestra el informe.
		EP 1.3: No seleccionar plan.	<ul style="list-style-type: none"> El usuario selecciona el reporte Plan anual de actividades.

- El sistema activa un menú central para seleccionar las condiciones de búsquedas.
- El usuario selecciona la condición de búsqueda caracterización del PAA.
- El usuario selecciona la opción Obtener reporte.
- El sistema muestra en mensaje “Debes seleccionar un plan”.

EP 1.4: No seleccionar condiciones de búsqueda.

1. El usuario selecciona el reporte Plan anual de actividades.
2. El sistema activa un menú central para seleccionar las condiciones de búsquedas.
3. El usuario selecciona un plan y la opción Obtener reporte.
4. El sistema muestra el informe para el Plan anual de actividades.

Tabla 9: Descripción de las variables

No	Nombre de campo	Tipo	Válido	Inválido
1	Fechas	Campo de selección, Date	Hacer clic en cualquiera de las opciones de fechas y seleccionar una	N/A
2	Caracterización del PAA	Campo de selección	Hacer clic en la opción Caracterización	N/A
3	Plan	Campo de selección	Hacer clic sobre en un Plan	N/A

Tabla 10: Juegos de datos a probar

Id del escenario	Escenario	Fechas	Caracterización del PAA	Plan	Respuesta del sistema	Resultado de la prueba

EP 1.1	Mostrar el reporte de caracterización	V(28/05/2016)	V(Vacío)	V(Vacío)	El sistema muestra un mensaje de error diciendo que hay que señalar un plan. Debe Mostrar el Plan Anual de actividades	
EP 1.1	Mostrar el reporte de caracterización	V(28/05/2016)	V(1)	V(Vacío)	El sistema muestra un mensaje de error diciendo que hay que señalar un plan.	
EP 1.1	Mostrar el reporte de caracterización	V(28/05/2016)	V(1)	V(1)	El sistema muestra el reporte correspondiente a la caracterización para ese plan	
EP 1.1	Mostrar el reporte de caracterización	V(Vacío)	V(1)	V(1)	El sistema muestra un mensaje de error diciendo que hay que señalar una fecha.	

Los demás casos de pruebas están descritos en el expediente de proyecto en la dirección Exp_Proyecto/Ingenieria/Pruebas.

3.3.2 Resultados de las pruebas aplicadas

Los métodos de pruebas aplicados a la funcionalidad demostraron resultados satisfactorios desde el punto de vista funcional. Donde cada no conformidad detectada fueron analizadas y corregidas debidamente, logrando un correcto comportamiento de la funcionalidad antes diferentes situaciones (entradas válidas y no válidas). A continuación se muestra un gráfico con el número de no conformidades detectadas tras cada iteración de pruebas realizadas.

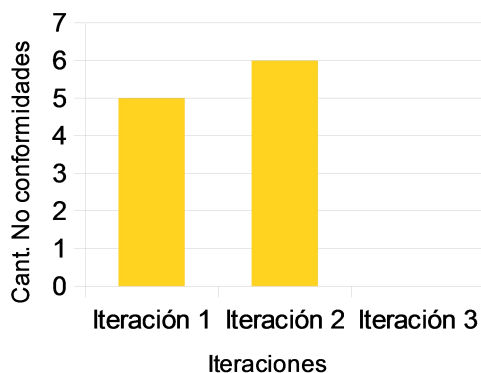


Figura 14: No conformidades por iteración

3.5 Impacto de la solución

De manera general la incorporación en SIPAC de la caracterización de la elaboración y ejecución del PAA facilita el registro y control de la Planificación estratégica según establece la Instrucción No.1 del Presidente de los Consejos de Estado y de Ministros; teniendo en cuenta que orienta el proceso de elaboración de los planes de los órganos, organismos y entidades nacionales en función de la especificación de las de tareas propias, externas, de aseguramiento en cada caso, y el porcentaje que representan; que dan la medida de cómo cada organismo o entidad dan cumplimiento a sus objetivos o al funcionamiento interno, predominando las del nivel superior. Los informes elaborados e implementados al sistema garantizan que las actividades de los planes anuales, mensuales o individuales cumplan con los acuerdos de los órganos colegiados, mandatos, indicaciones o decisiones de los jefes logrando integrar y coordinar en el plan las acciones entre todos los factores. También permite la revisión de los PTI de los subordinados donde se exponen los señalamientos correspondientes y un análisis de su comportamiento de manera formal.

3.6 Conclusiones parciales

En el transcurso del presente capítulo se analizó:

- Se compró que el estándar de codificación se utilizó correctamente en la implementación de la funcionalidad.
- Se aplicó las métricas TOC y RC para la validación del diseño, las cuales arrojaron valores satisfactorios para cada uno de los indicadores correspondientes, demostrándose que la solución desarrollada cuenta con alto por ciento de reutilización para las dos métricas, obteniendo también un bajo por ciento en reutilización, complejidad de implementación, acoplamiento complejidad de mantenimiento y cantidad de pruebas.



- Se aplicó las pruebas de caja negra y de caja blanca para comprobar la correcta ejecución de la funcionalidad, donde se detectaron un total de 6 no conformidades en su totalidad, las cuales fueron detectadas y debidamente resuelta.
- Se realizó un análisis del impacto de la funcionalidad llegando a la conclusión de que con el desarrollo de la misma, se realizará una mejor planificación tanto en entidades nacionales como a nivel de Gobierno en función de cumplir los objetivos de cada organización.

CONCLUSIONES GENERALES

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

- El estudio de modelos de dirección así como de la Instrucción No.1 permitieron identificar características y mejores prácticas, donde se evidenció la necesidad de incorporar al sistema informes para la caracterización, resúmenes de cumplimiento y desglose de los planes.
- El modelado de negocio de la funcionalidad desarrollada permitió conocer las condiciones necesarias mediante la identificación de las reglas de negocio así como, los conceptos fundamentales.
- El diseño asociado a la solución posibilitó que la implementación de la misma cumpliera con todos los requisitos capturados y la satisfacción de las necesidades del cliente.
- La implementación de la caracterización de la elaboración y ejecución del PAA permitió obtener un producto capaz de permitir a los usuarios que operen con ella la visualización de la caracterización, el estado de cumplimiento de los planes y el desglose por ARC.
- La validación del diseño y de la solución permitieron encontrar no conformidades en la funcionalidad, que fueron corregidas para garantizar la calidad requerida.

RECOMENDACIONES

Al concluir el presente trabajo de diploma y considerando cumplidos todo los objetivos trazados, se recomienda lo siguiente:

- Comprobar la factibilidad de la funcionalidad desarrollada tomando en cuenta diferentes entornos organizacionales.
- Concebir y desarrollar una funcionalidad que permita graficar los datos que se obtienen en la generación de los resúmenes de cumplimiento y la Caracterización del PAA.

REFERENCIAS BIBLIOGRÁFICAS

1. **Paz, MSC. Norma Sánchez.** *Fundamentos y métodos generales de planificación.* UNIVERSIDAD DE HOLGUÍN.
2. **Villalón-Madrado, Lic. Kenilia M.** *La planificación y el modelo económico cubano.* Santiago de Cuba: s.n., 2012.
3. **Herrera, Luz Marina Carvajal.** *Guía para la elaboración del plan de trabajo.* Universidad de Colombia: s.n., 2014.
4. **Bernal, Dra. María.** *La Planificación: Conceptos básicos, principios, componentes, características y desarrollo del proceso.* Los Teques: s.n., 2012.
5. **Montano, Norberto Gibert. A:** *Los jefes de órganos, OACE, entidades nacionales y consejos de administración de organismos locales del Poder Popular.* La Habana: s.n., 2014.
6. **Isotools.** [En línea] [Citado el: 3 de 05 de 2016.]
<https://www.isotools.org/soluciones/personas/direccion-por-objetivos/>.
7. **Eloisa Diaz Garrido, José Pinillos Costa.** *Dirección por Objetivo. La dirección participativa.* 2013.
8. **Pública, Dpto. Salud.** *Planificación.* Facultad Manuel Fajardo : s.n., 2014.
9. **Albavera, Fernando Sánchez.** *Planificación estratégica y gestión pública por objetivos.* Santiago de Chile: s.n., 2013.
10. **CASTRO RAÚL.** *Instrucción No.1 del Presidente de los Consejos de Estado y Ministro.* La Habana: s.n., 2011.
11. **Sánchez, Tamara Rodríguez.** *Metodología de desarrollo para la actividad productiva de la UCI.* Universidad de las Ciencias Informáticas: s.n., 2015.
12. **Gutiérrez, Javier J.** *¿Qué es un framework Web?* Available in: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf Accessed May, 2014, vol. 12.
13. **Torres, Abraham Calás.** *Modelo de componentes para el marco de trabajo Sauxe.* 2014
14. **Correa-Isaza, Daniel Guillermo, et al.** *Buenas prácticas para el desarrollo de aplicaciones web a través de patrones de diseño usando el marco de trabajo Zend Framework.* 2014. Tesis Doctoral. Escuela de Ingeniería de Antioquia.
15. **Doctrine 1.2 ORM Manual.** [Citado el: 15 de enero 2016]; Available from: <http://docs.doctrine-project.org/projects/doctrine1/en/latest/en/manual/introduction.html#aboutthis-version>.
16. **Fabián Wilfrido Chimbo Fernández, Edison Paul Villa Mendoza.** *Análisis comparativo entre los frameworks Primeface y Ext JS para el desarrollo de la aplicación web de gestión y evaluación del desempeño de puestos de trabajo de la ESPOCH.* 2014.

17. **Asensio, Rafael Menéndez-Barzanallana.** *JAVASCRIPT*. 2012.
18. **ACERCA DE HTML.** [En línea] [Citado el: 27 de 03 de 2016.] <http://www.acercadehtml.com/manual-html/que-es-html.html>.
19. **Manual de CSS, hojas de estilo.** [Citado el: 12 de enero de 2016]; Available from: <http://www.desarrolloweb.com/manuales/manual-css-hojas-de-estilo.html>.
20. **PHP.** [Citado el: 03 de 03 de 2016]; Available from: <http://www.php.net/>.
21. **Vidal, Cristian L., et al.** *Extensión del Diagrama de Secuencias UML (Lenguaje de Modelado Unificado) para el Modelado Orientado a Aspectos. Información tecnológica*, 2012, vol. 23, no 6, p. 51-62.
22. **Bray, Tim.** *The javascript object notation (json) data interchange format*. 2014.
23. **Introducción a AJAX.** [En línea] [Citado el: 27 de 03 de 2016.] http://librosweb.es/libro/ajax/capitulo_1.html
24. **Guión Visual Paradigm for UML.** [En línea] 2013-2014. [Citado el: 20 de 02 de 2016.] <http://www.ie.inf.uc3m.es/grupo/docencia/reglada/1s1y2/PracticaVP.pdf>.
25. **NetBeans IDE.** [Citado el: 11 de enero 2016]; Available from: <https://netbeans.org/features/index.html>.
26. **Subversion.** [Citado el: 21 enero de 2016]. <http://subversion.apache.org>
27. **Group, The PostgreSQL Global Development.** *PostgreSQL 9.3.11 Documentation*. 2011.
28. **Mestras, Juan Pavón.** *Material bajo licencia Creative Commons*. Universidad Complutense Madrid: s.n., 2012.
29. **CAÑAS, Alberto J.** *Herramientas para construir y compartir modelos de conocimiento basados en mapas conceptuales*. Revista de informática educativa, 20138.
30. **González, Juan Carlos García; AMPUERO, Margarita André.** *Implementación del enfoque de reglas de negocio utilizando motores de reglas en el desarrollo de aplicaciones Java. Ciencias de la Información*, 2015.
31. **Pytel, Pablo.** *Ingeniería de requisitos basada en técnicas de ingeniería del conocimiento*. En XIII Workshop de Investigadores en Ciencias de la Computación. 2012.
32. **Devmedia. Engenharia de Software 2 - Técnicas para levantamento de Requisitos.** [En línea] [Citado el: 22 de 03 de 2016.] <http://www.devmedia.com.br/engenharia-de-software-2-tecnicas-para-levantamento-de-requisitos/9151>.
33. **SG Buzz.** [En línea] [Citado el: 22 de 03 de 2016.] <http://sg.com.mx/revista/17/obtencion-requerimientos-tecnicas-y-estrategia>.

34. **Zapata-Jaramillo, Carlos Mario.** *UNC-analista: hacia la captura de un corpus de requisitos a partir de la aplicación del experimento Mago de Oz.* 2014.
35. **Juan Carlos Moreno, Marcelo Martín Marciszack.** *La Usabilidad Desde La Perspectiva De La Validación.* Córdoba : s.n., 2013.
36. **Rojo, Silvana del Valle.** *Especificación de requerimientos no funcionales en aplicaciones web.* 2014. Tesis Doctoral. Facultad de Informática.
37. **Marcelo Marciszack, Ramiro Pérez, Claudia Castro.** Validación de Requerimientos a través de Modelos. SEDICI. [En línea] 2013. [Citado el: 14 de 03 de 2016.]
<http://sedici.unlp.edu.ar/handle/10915/27231>.
38. **Moderp.** *Modelo Prescriptivos de proceso.* 2013.
39. **Trenilli, Lic. Ariel.** *Arquitectura y diseño de sistemas.* 2015.
40. **Mora, Juan Tahuiton.** *Arquitectura de software para aplicaciones Web.* Mexico: s.n., 2012.
41. **González, Yanette Díaz; ROMERO, Yenisleidy Fernández.** Patrón Modelo-Vista-Controlador. *Revista Telem@tica*, 2012, vol. 11, no 1, p. 47-57.
42. **Gómez, José Jorge Márquez.** *Arquitectura MVC. Visión general.* 2014.
43. **Hernández, Pedro Veloso.** *Uso de patrones de arquitectura.* 2014
44. **Cortez, Alberto Alejandro.** Validación de Patrones de Diseño de Comportamiento a través de Perfiles UML: Observer, un caso de estudio. *Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação*, 2015.
45. **Carmona, Juan García.** *Buenas prácticas hacia el éxito en el desarrollo de software.* 2012.
46. **Patrones del "Gang of Four".** Madrid. 2013.
47. **CIG-SPA-N-i4302-**Manual de usuario SIPAC. La Habana : s.n., 2015.
48. **CENTRO DE INFORMATIZACIÓN DE ENTIDADES.** *Estándares de codificación Proyectos con el marco de trabajo Sauxe del CEIGE.*
49. **Aparicio, Ing. Alexandra.** *Ingeniería de software.* 2012.
50. **Dpto. Profesionales, Carreras.** *Pruebas de software.* 2012.
51. **Santana, Alicia Bárbara Expósito.** *Pruebas de caja blanca. Prueba y Mantenimiento del Software.* 2012.
52. **Govea, Blanca A. Vargas.** *Aseguramiento de la calidad y pruebas d software.* 2013.
53. **LÓPEZ, DR. MARIO ROSSAINZ.** *Pruebas de caja negra.* 2016.