

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 3



**Sistema informático para la evaluación de los profesores del Departamento de
Ingeniería de Software de la Facultad 3**

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

Autor:

Yaima Cepero de Loyola

Tutor:

Ing. Yoan Antonio López Rodríguez.

La Habana

Junio, 2015

Año 56 de la Revolución

Pensamiento



El futuro tiene muchos nombres, para los débiles es lo inalcanzable, para los temerosos, lo desconocido y para los valientes es la oportunidad.

Víctor Hugo.

Declaración de autoría

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Autor:

Yaima Cepero de Loyola

Tutor:

Yoan Antonio López Rodríguez

Datos de contacto

Tutor:

Ing. Yoan Antonio López Rodríguez

- Especialista del Centro CEIGE, profesor asistente.
- Ha tutorado 8 tesis relacionadas con la gestión bancaria y software educativo.

Correo electrónico: yalopez@uci.cu

Agradecimientos

En especial quiero agradecer ser hoy la profesional que soy a mis padres, por haber sido capaces de traerme hoy hasta aquí, con todos los esfuerzos, preocupaciones y dolores de cabeza que eso ha implicado. A mi mamá por ser una súper madre, por ser tan luchadora, incansable, por darme fuerzas cuando más lo necesito, por sus consejos, por hacer de mí una mujer de bien...Te quiero mucho mamá... A mi papá tan luchador, preocupado siempre por mí y lograr ser hoy lo que soy...Te quiero papá... A mi querido hermanito del alma por apoyarme...Te adoro... A mi adorado esposo por darme la felicidad que hoy siento, por amarme tanto, por ser de mí otra persona, por ser todo lo que yo quería para mí...Te Amo... A la familia de mi esposo, mi suegrí, mi cuñí por acogerme con tanto cariño...los quiero...

A mi tutor por su dedicación y apoyo durante el desarrollo de la tesis. A mis compañeros de aula por compartir estos años de estudio con ellos a los viejos y a los nuevos. A Rosy gracias por apoyarme siempre y estar conmigo, por acogerme en tu cuarto. A las niñas del apartamento gracias por ayudarme en estos meses. A la profe Mailén por preocuparse tanto de mí, a la profe Marieta por estar al pendiente de mi tesis, a la profe Olga, al tribunal por ser tan paciente conmigo, a todos los profesores que me impartieron clases en el transcurso de estos años y que contribuyeron en mi formación. A la UCI, porque me ha permitido crecer, como profesional y como persona, gracias por todas las oportunidades que me ha brindado. A todos los que permitieron que hoy estuviera escribiendo estas letras muchas gracias.

De Yaíma Cepero de Loyola

Dedicatoria

Quisiera dedicarle esta tesis que evidencia el esfuerzo y resultado que he obtenido durante estos años de la carrera y que si no hubiera sido por ellos no habría tenido la fuerza de seguir adelante con mis sueños, a mis padres.

A mi madre querida que para mí hay pocas en este mundo por ser tan fuerte, por darme todo su apoyo siempre que lo necesito y por ser mi mejor amiga en los buenos y malos momentos de la vida, por ser un ejemplo para mí de mujer y madre luchadora en todo momento.

A mi papá porque es la persona más noble que conozco, luchador y por apoyarme siempre.

A mi esposo, por apoyarme durante todos estos años, por aguantarme mis malacrianzas y mis pesadeces.

A mi hermano, gracias por apoyarme en todo y por estar ahí siempre.

De Yaíma Cepero de Loyola

Resumen

En el Departamento de Ingeniería de Software de la Facultad 3 de la Universidad de las Ciencias Informáticas, se lleva a cabo la gestión de los planes de trabajo tanto de los profesores como del departamento. El plan de trabajo de un profesor abarca un conjunto de actividades que se derivan de los resultados que se planifica el profesor para un año. Estos resultados se elaboran en correspondencia con los objetivos del departamento en cada una de las áreas de resultados claves. El plan de trabajo del departamento contiene un conjunto de actividades, que van a estar respondiendo a los objetivos que trace el departamento para el año. En la actualidad la gestión de los planes de trabajo se realiza de forma manual y por ello, a pesar de la estrecha relación que existe entre las actividades y los resultados de los profesores con los objetivos del departamento, determinar el cumplimiento de los resultados a partir del cumplimiento de las actividades de los profesores se torna engorroso. El presente trabajo describe el desarrollo de un sistema informático que permita lograr mayor correspondencia entre las evaluaciones de los profesores y el cumplimiento de los objetivos del Departamento de Ingeniería de Software de la Facultad 3. Al concluir este trabajo se lograron satisfacer las necesidades identificadas en el proceso de planificación a partir del diseño e implementación de las funcionalidades previstas, las cuales fueron validadas a través del desarrollo de diferentes tipos de pruebas.

Palabras claves: actividades, objetivos, plan de trabajo, planificación y resultados.

Índice de contenido

| | |
|------------------------------------------------------------------------------------|----|
| Introducción | 2 |
| Capítulo 1: Fundamentación teórica | 6 |
| 1.1 Introducción | 6 |
| 1.2 Conceptos básicos relacionados con el dominio del problema | 6 |
| 1.2.1 Software | 6 |
| 1.2.2 Planificación | 6 |
| 1.2.3 Plan de trabajo | 6 |
| 1.3 Proceso de desarrollo de software | 7 |
| 1.4 Evolución y desarrollo del software | 8 |
| 1.5 Clasificación del software | 9 |
| 1.5.1 Software de sistema | 9 |
| 1.5.2 Software de programación | 9 |
| 1.5.3 Software de aplicación | 9 |
| 1.6 Software de gestión | 10 |
| 1.6.1 Tipos de software de gestión | 10 |
| 1.7 Proceso de planificación en el departamento de Ingeniería de Software | 14 |
| 1.8 Análisis de soluciones existentes | 14 |
| 1.9 Valoración del estudio realizado | 16 |
| 1.10 Metodologías para el desarrollo del software | 17 |
| 1.10.1 Proceso Unificado Ágil (AUP-UCI) | 17 |
| 1.11 Herramientas y tecnologías | 18 |
| 1.11.1 Lenguaje de Modelado (UML 2.0) | 18 |

| | | |
|-----------------------------------------------|----------------------------------------------------------------------------------|----|
| 1.11.2 | Herramienta CASE (Visual Paradigm 8.0) | 18 |
| 1.11.3 | Lenguajes de programación..... | 19 |
| 1.11.4 | Sistema Gestor de Bases de Datos | 21 |
| 1.11.5 | Entorno de Desarrollo Integrado (PhpStorm 8.0) | 22 |
| 1.11.6 | Marco de Trabajo..... | 22 |
| 1.11.7 | Servidor de Aplicaciones Web | 22 |
| 1.12 | Conclusiones del capítulo | 23 |
| Capítulo 2: Características del sistema | | 24 |
| 2.1 | Introducción..... | 24 |
| 2.2 | Modelo conceptual..... | 24 |
| 2.3 | Técnicas de captura de requisitos..... | 25 |
| 2.4 | Requisitos funcionales del sistema | 26 |
| 2.5 | Historias de usuario | 29 |
| 2.6 | Requisitos no funcionales del sistema | 30 |
| 2.7 | Técnicas de validación de requisitos..... | 32 |
| 2.8 | Modelo del diseño..... | 32 |
| 2.8.1 | Diagrama de clases | 32 |
| 2.8.2 | Diagrama de secuencia | 33 |
| 2.8.3 | Modelo de datos | 34 |
| 2.9 | Patrón arquitectónico | 35 |
| 2.10 | Patrones de diseño | 36 |
| 2.10.1 | Patrones generales de software para asignación de responsabilidades (GRASP)..... | 37 |
| 2.11 | Diagrama de componentes | 39 |
| 2.12 | Validación del diseño mediante métricas | 40 |

| | | |
|--------------------------------------------------------------------------------------------|-------------------------------------------|----|
| 2.12.1 | Métrica Tamaño operacional de clase | 40 |
| 2.12.2 | Métrica Relaciones entre Clases | 44 |
| 2.13 | Conclusiones del capítulo | 48 |
| Capítulo 3: Implementación y pruebas | | 49 |
| 3.1 | Introducción | 49 |
| 3.2 | Implementación | 49 |
| 3.2.1 | Estándares de codificación | 49 |
| 3.3 | Interfaces del sistema | 51 |
| 3.4 | Diagrama de despliegue | 53 |
| 3.5 | Pruebas de software | 54 |
| 3.5.1 | Pruebas de Caja negra | 54 |
| 3.5.2 | Prueba de Caja blanca | 55 |
| 3.5.3 | Resultados de las pruebas | 56 |
| 3.6 | Validación de la investigación | 61 |
| 3.7 | Conclusiones del capítulo | 61 |
| Conclusiones Generales | | 62 |
| Recomendaciones | | 63 |
| Bibliografía | | 64 |
| Anexos | | 70 |
| Anexo 1. Diagrama de clases de la agrupación Gestionar actividad | | 70 |
| Anexo 2. Diagrama de clases de la agrupación Gestionar objetivos | | 71 |
| Anexo 3. Diagrama de clases de la agrupación Gestionar plan mensual del departamento | | 72 |
| Anexo 4. Diagrama de clases de la agrupación Gestionar plan mensual del profesor | | 73 |
| Anexo 5. Diagrama de clases de la agrupación Gestionar plan de objetivos | | 74 |

| | |
|--------------------------------------------------------------------------------------------------------------------------------|-----|
| Anexo 6. Diagrama de clases de la agrupación Gestionar profesor..... | 75 |
| Anexo 7. Diagrama de clases de la agrupación Gestionar resultados | 76 |
| Anexo 8. Diagrama de clases de la agrupación Gestionar departamento..... | 77 |
| Anexo 9. Diagrama de clases de la agrupación Gestionar disciplina..... | 78 |
| Anexo 10. Diagrama de clases de la agrupación Gestionar colectivo..... | 79 |
| Anexo 11. Diagrama de clases de la agrupación Gestionar Usuario | 80 |
| Anexo 12. Diagrama de clases de la agrupación Gestionar Usuario | 81 |
| Anexo 13. Historias de usuario perteneciente a la agrupación por requisitos Gestionar departamento .. | 81 |
| Anexo 14. Historias de usuario perteneciente a la agrupación por requisitos Gestionar disciplina | 84 |
| Anexo 15. Historias de usuario perteneciente a la agrupación por requisitos Gestionar colectivo | 87 |
| Anexo 16. Historias de usuario perteneciente a la agrupación por requisitos Gestionar profesor | 90 |
| Anexo 17. Historias de usuario perteneciente a la agrupación por requisitos Gestionar plan de objetivos del departamento | 93 |
| Anexo 18. Historias de usuario perteneciente a la agrupación por requisitos Gestionar plan de resultados | 96 |
| Anexo 19. Historias de usuario perteneciente a la agrupación por requisitos Gestionar plan mensual del departamento | 99 |
| Anexo 20. Historias de usuario perteneciente a la agrupación por requisitos Gestionar plan mensual del profesor..... | 102 |
| Anexo 21. Diagrama de clases de la agrupación por requisitos Gestionar objetivos | 105 |
| Anexo 22. Historias de usuario perteneciente a la agrupación por requisitos Gestionar resultado | 108 |
| Anexo 23. Historias de usuario perteneciente a la agrupación por requisitos Gestionar actividad | 111 |
| Anexo 24. Prueba de aceptación..... | 114 |
| Anexo 25. Diagrama de componentes para el sistema..... | 115 |

Índice de Figuras

| | |
|----------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 1: Modelo conceptual del departamento | 25 |
| Figura 2: Diagrama de clases de la agrupación por requisitos del Gestionar plan de resultados | 33 |
| Figura 3: Diagrama de secuencia del requisito Adicionar plan de resultados | 34 |
| Figura 4: Modelo de datos | 35 |
| Figura 5: Diseño arquitectónico de Symfony | 36 |
| Figura 6: Método adicionar de la clase TPlanResultadosRepository | 38 |
| Figura 7: Patrón bajo acoplamiento en la clase TPlanResultadosRepository | 39 |
| Figura 8: Diagrama de componentes | 39 |
| Figura 9: Representación de la evaluación de la métrica TOC | 42 |
| Figura 10: Representación en por ciento de los resultados obtenidos en la evaluación de la métrica TOC | 42 |
| Figura 11: Resultados de la evaluación de la métrica TOC para el atributo responsabilidad | 43 |
| Figura 12: Resultados de la evaluación de la métrica TOC para el atributo complejidad | 43 |
| Figura 13: Resultados de la evaluación de la métrica TOC para el atributo reutilización | 44 |
| Figura 14: Representación en por ciento de los resultados obtenidos en los intervalos definidos según la métrica RC | 46 |
| Figura 15: Resultados de la evaluación de la métrica RC para el atributo acoplamiento | 46 |
| Figura 16: Resultados de la evaluación de la métrica RC para el atributo complejidad de mantenimiento | 47 |
| Figura 17: Resultados de la evaluación de la métrica RC para el atributo reutilización | 47 |
| Figura 18: Resultados de la evaluación de la métrica RC para el atributo cantidad de pruebas | 48 |
| Figura 19: Pantalla de inicio asociada al rol jefe de departamento | 52 |
| Figura 20: Pantalla de inicio asociada al rol profesor | 52 |
| Figura 21: Diagrama de despliegue | 53 |
| Figura 22: Código fuente de la funcionalidad adicionar (\$objeto, \$resultado) | 57 |
| Figura 23: Grafo de flujo asociado a la funcionalidad adicionar (\$objeto, \$resultado) | 57 |
| Figura 24: Resultados de la prueba de Caja negra | 60 |

| | |
|--------------------------------------------------------------------------------------|----|
| Tabla 1: Listado de requisitos funcionales | 26 |
| Tabla 2: Historia de usuario del requisito Adicionar plan de resultados | 29 |
| Tabla 3: Métrica Tamaño operacional de clase | 41 |
| Tabla 4: Rango de valores para la métrica TOC..... | 41 |
| Tabla 5: Atributos de calidad evaluados por la métrica RC..... | 44 |
| Tabla 6: Rango de valores para la métrica RC | 45 |
| Tabla 7: Caso de prueba para el camino básico #1 | 59 |
| Tabla 8: Caso de prueba para el camino básico #2..... | 59 |
| Tabla 9: Caso de prueba para el camino básico #3..... | 59 |

Introducción

Esta es la Era de la Información y de las Tecnologías Informáticas, poseer información, generarla, transformarla, almacenarla y transmitirla es sinónimo de progreso asegurado. Una sociedad que aplique la informatización en todas sus esferas y procesos será más eficaz, eficiente y competitiva. Es evidente que para los países subdesarrollados resulta un reto el logro de este propósito, debido al costo tan alto de la tecnología y del conocimiento que se necesita para trabajar con ella. Cuba ha identificado desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social las Tecnologías de la Información y las Comunicaciones (TICs) y lograr una cultura digital como una de las características imprescindibles del hombre nuevo, lo que facilitaría a la sociedad acercarse más hacia el objetivo de un desarrollo sostenible. Marcada por el bloqueo económico, comercial y financiero de Estados Unidos contra la isla, la informatización de la sociedad cubana es un proceso en avance, y los principales logros en esa área se muestran en sectores prioritarios como la Salud Pública y la Educación. (1)

Hoy día existe un gran número de empresas e instituciones que se encuentran envueltas en un gran flujo de datos procesados de forma manual y de cierta forma desorganizada. Por ello, se realizan herramientas informáticas para la gestión de todos estos datos, pues las mismas permiten agilizar el trabajo, además de poder gestionar gran cantidad de información logrando que la toma de decisiones se realice de forma rápida y concisa. En el país las máximas autoridades de los órganos, organismos, organizaciones y demás entidades deben integrar a su sistema de control interno el proceso de planificación, los objetivos y planes de trabajo, para relacionarlo con los procesos, actividades y operaciones en el interés de asegurar el cumplimiento de su misión y de las disposiciones legales que le competen. (2)

El control y seguimiento del plan de trabajo individual permite a una organización obtener información sobre actividades pasadas y en curso, que se pueden emplear como base para ajustar la programación de las mismas y reorientarlas para la planificación futura. El análisis de dicho documento permite realizar comparaciones de lo realizado durante un período determinado por un trabajador y asignar la evaluación de éste.

La Universidad de las Ciencias Informáticas (UCI) como casa de altos estudios, encargada de formar ingenieros en la rama de la informática, cuenta con fortalezas en cuanto a tecnología y capital humano. Actualmente se desarrollan una serie de proyectos para la informatización de algunas empresas tanto nacionales como internacionales, pero también emplea su intelecto en la informatización de los procesos

que rigen la vida en la Universidad. La universidad está integrada por varias facultades, donde cada una de ellas cuenta con una estructura administrativa formada por: un decano, 4 vicedecanos (vicedecano de formación, vicedecano de investigación y postgrado, vicedecano de extensión universitaria y vicedecanato de administración), varios departamentos y varios centros productivos en dependencia de su perfil productivo.

En el Departamento de Ingeniería de Software de la Facultad 3 se lleva a cabo el proceso de evaluación de sus profesores. Inicialmente el departamento define sus objetivos para el año a partir de los cuales se conforma el plan de resultados anual del profesor, el cual se desglosa por áreas de resultados claves. A partir del plan de resultados se conforma el plan de actividades mensual del profesor, donde el cumplimiento de las actividades va a tributar al cumplimiento de los resultados. A nivel de facultad se ha planteado que la evaluación de los profesores deberá estar en correspondencia con el cumplimiento de los objetivos del área a la cual pertenecen. En estos momentos a la dirección del departamento se le dificulta obtener información de los objetivos que se estarán cumpliendo en un período determinado, ya que la gestión manual de los mismos no permite establecer trazabilidad entre los objetivos del departamento y los planes de trabajo de los profesores, dificultándose además la gestión de las evidencias del cumplimiento de las actividades de los profesores.

Lo anteriormente planteado conduce a la formulación del siguiente **problema a resolver**: ¿Cómo lograr mayor correspondencia entre las evaluaciones de los profesores y el cumplimiento de los objetivos del Departamento de Ingeniería de Software de la Facultad 3?

Esta problemática se enmarca en el **objeto de estudio**: Procesos de desarrollo de software.

Para dar una solución efectiva al problema se propone como **objetivo general**: Desarrollar un sistema informático que permita lograr mayor correspondencia entre las evaluaciones de los profesores y el cumplimiento de los objetivos del Departamento de Ingeniería de Software de la Facultad 3.

El objetivo planteado anteriormente delimita **el campo de acción**: Procesos de desarrollo de software de gestión.

Como **objetivos específicos** se definen los siguientes:

- Elaborar el marco teórico de la investigación en función del objeto de estudio.
- Desarrollar el sistema para la evaluación de los profesores del Departamento de Ingeniería de Software de la Facultad 3.
- Validar el sistema propuesto.

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

1. Elaboración del marco conceptual de la investigación.
2. Estudio de Sistemas de planificación.
3. Selección de la metodología de desarrollo de software a emplear, sus etapas y principales artefactos.
4. Definición de la tecnología a emplear en el desarrollo del sistema.
5. Identificación de los requisitos funcionales.
6. Especificación de la arquitectura del sistema.
7. Diseño de los diagramas de clases del diseño.
8. Diseño del modelo de datos mediante un diagrama entidad relación.
9. Diseño del diagrama de componentes del sistema.
10. Validación del diseño de las clases mediante métricas.
11. Implementación del sistema.
12. Realización de pruebas de caja negra sobre las funcionalidades del sistema.
13. Realización de pruebas de caja blanca sobre el código desarrollado.
14. Realización de pruebas de aceptación con el cliente.

Idea a defender:

El desarrollo de un sistema informático permitirá lograr una mayor correspondencia entre las evaluaciones de los profesores y el cumplimiento de los objetivos del Departamento de Ingeniería de Software de la

Facultad 3.

En el trabajo se hace necesario el empleo de varios métodos de investigación, teóricos y empíricos.

Como **métodos teóricos** se utilizan; el **histórico-lógico**: con el objetivo de conocer el desarrollo y evolución del software desde sus inicios hasta la actualidad, el **analítico-sintético**: a través de este método se analiza la bibliografía disponible para lograr un mejor entendimiento y arribar a conclusiones sobre el proceso de confección, gestión y control de los planes de trabajo de los profesores. Por último, la **modelación**: se utiliza en la confección de los artefactos propuestos por la metodología escogida, para lograr un mejor entendimiento en la implementación del sistema. Como **método empírico** se utiliza la **observación**: que permite obtener conocimiento acerca del comportamiento del objeto de investigación tal y como este se desarrolla en la realidad.

El contenido del presente trabajo se estructura en 3 capítulos:

Capítulo 1. Fundamentación Teórica: en este capítulo se aborda el desarrollo y evolución del software, así como las metodologías que son de gran utilidad, como guías para lograr un proceso de desarrollo organizado. Se especifican las herramientas y los lenguajes para el desarrollo de la investigación.

Capítulo 2. Características del sistema: en este capítulo se comprende cómo funciona el negocio que se desea informatizar, para tener garantías de que el software desarrollado va a cumplir su propósito. Se especifican los requisitos funcionales y no funcionales del sistema. Se definen las clases del diseño, el modelo de datos del sistema y se explican los patrones utilizados en el diseño y el patrón arquitectónico aplicado en la construcción del software.

Capítulo 3. Implementación y Pruebas: en este capítulo se lleva a cabo la implementación del sistema y se realizan las pruebas necesarias para la verificación del software y la validación de la investigación.

Capítulo 1: Fundamentación teórica

1.1 Introducción

En el presente capítulo se abordan los conceptos fundamentales relacionados con el problema, se realiza un estudio sobre el desarrollo y evolución del software, se explica la clasificación de los tipos de software que existen y las características de algunos de los sistemas de gestión que se utilizan tanto nacional como internacionalmente. Se realiza un análisis de sistemas homólogos para la planificación de los planes de trabajo y un estudio del proceso de desarrollo de software como vía para desarrollar un sistema de forma organizada y con la calidad requerida, haciendo uso de los modelos y la metodología seleccionada. Además se especifican las herramientas y tecnologías seleccionadas para la construcción del software.

1.2 Conceptos básicos relacionados con el dominio del problema

Para el desarrollo de la aplicación fue necesaria la comprensión y estudio de un grupo de términos tales como:

1.2.1 Software

Según la Real Academia Española, el software es un conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora. (3)

1.2.2 Planificación

Existen varios tipos de planificación pero todos ellos responden a una misma cláusula, formas, métodos y acciones que son ideados con el fin de alcanzar ciertas metas de forma ordenada y eficiente. (4)

1.2.3 Plan de trabajo

Un plan de trabajo es una herramienta que permite ordenar y sistematizar información relevante para realizar un trabajo. Esta especie de guía propone una forma de interrelacionar los recursos humanos, financieros, materiales y tecnológicos disponibles. Como instrumento de planificación, el Plan de Trabajo establece un cronograma, designa a los responsables, marca metas y objetivos. El Plan de Trabajo suele ser válido para

un determinado período de tiempo. De esta manera, las acciones que propone deben desarrollarse en un cierto plazo y los objetivos tienen que ser cumplidos antes de una fecha límite. (5)

1.3 Proceso de desarrollo de software

Proceso de software: Pressman define proceso de software como un marco de trabajo de las tareas que se requieren para construir un software de alta calidad. (6)

El software requiere un tiempo y esfuerzo considerable para su desarrollo. Durante este tiempo de desarrollo, desde que se detecta la necesidad de construir un software hasta que este es retirado, se identifican varias etapas que en conjunto se denominan el ciclo de vida del software y en cada caso, en función de cuales sean las características del proyecto, se configurará el ciclo de vida de forma diferente ajustándolo a las necesidades y particularidades de cada proyecto. (7)

Hay varios modelos a seguir para el establecimiento de un proceso en el desarrollo de software, cada uno de los cuales describe un enfoque diferente para diferentes actividades que tienen lugar durante el proceso.

Los modelos de desarrollo se utilizan para lograr los siguientes objetivos (7):

- Un proceso de desarrollo controlado para obtener mejores aplicaciones con mayor calidad.
- Especificar los requerimientos de un software en forma apropiada.
- Construir productos bien documentados y de fácil mantenimiento.
- Ayudar a identificar las necesidades de cambio lo más pronto posible.
- Proporcionar un sistema que satisfaga a todas las personas involucradas.

Pressman define una serie de modelos a seguir para el desarrollo de un software, entre los que se encuentran (8):

- Modelo en cascada
- Modelo incremental
- Modelo en espiral
- Desarrollo basado en componentes
- Desarrollo de software orientado a aspectos

1.4 Evolución y desarrollo del software

La primera etapa del software comienza en 1950, durante estos primeros años de la computadora lo normal era que el hardware fuera de propósito general. Por otra parte, el software se diseñaba a medida para cada aplicación y tenía una distribución relativamente pequeña. El software como producto estaba en su infancia, la mayoría de los software se desarrollaban y eran utilizados por la misma persona, se trabajaba con la idea de codificar y corregir. Debido a este entorno personalizado del software, el diseño era un proceso implícito, realizado en la mente de alguien y la documentación normalmente no existía. La segunda etapa en la evolución de los sistemas de computadora comienza a partir de 1965 y se extiende hasta 1972, apareciendo la multiprogramación y los sistemas multiusuario introduciendo nuevos conceptos de interacción hombre – máquina. Los sistemas de tiempo real podían recoger, analizar y transformar datos de múltiples fuentes, controlando así los procesos y produciendo salidas en milisegundos en lugar de minutos. (9)

Los avances en los dispositivos de almacenamiento en línea condujeron a la primera generación de sistemas de gestión de bases de datos. Conforme crecía el número de sistemas informáticos, comenzaron a extenderse las bibliotecas de software de computadoras. Se desarrollaban proyectos en los que se producían programas de decenas de miles de sentencias fuente que tenían que ser corregidos cuando se detectaban fallos, modificados cuando cambiaban los requisitos de los usuarios o adaptados a nuevos dispositivos hardware que se hubieran adquirido, a esta actividad se le llamó mantenimiento del software. (9)

La tercera etapa de la evolución del software comenzó a partir de 1972 hasta 1989. En esta etapa aparecen los sistemas distribuidos, múltiples computadoras, cada una ejecutando funciones concurrentemente y comunicándose con alguna otra, incrementó notablemente la complejidad de los sistemas informáticos. La conclusión de la tercera etapa se caracterizó por la llegada y amplio uso de los microprocesadores y con esto la llegada de la computadora personal. La cuarta etapa comienza a partir de 1989 hasta la actualidad y se centra en potentes máquinas personales controladas por sistemas operativos sofisticados, acompañadas por aplicaciones de software avanzadas. Las arquitecturas informáticas cambiaron de entornos centralizados de grandes computadoras a entornos descentralizados cliente/servidor. Aparecen las redes neuronales, los sistemas expertos y los software de inteligencia artificial. (9)

El desarrollo de software es posiblemente una de las áreas que va avanzando a pasos agigantados con el paso del tiempo. El desarrollo de las nuevas tecnologías informáticas ha traído consigo cambios en la forma de actuar y de pensar en la sociedad, trazando nuevas pautas en el comportamiento humano, resultando incuestionable el auge cada vez mayor del uso y aplicación de las Tecnologías de la Información y las Comunicaciones (TICs).

1.5 Clasificación del software

En el campo de la informática se pueden encontrar diferentes clasificaciones del software, tales como (10):

- Software de sistema
- Software de programación
- Software de aplicación

1.5.1 Software de sistema

Es el software que permite tener una interacción con el hardware. Dicho sistema es un conjunto de programas que administran los recursos del hardware y proporcionan una interfaz al usuario. Es el software esencial para una computadora, sin él no podría funcionar, como ejemplo se puede mencionar a los sistemas operativos (Windows, Linux, Mac OS X), controladores de dispositivos y servidores.

1.5.2 Software de programación

Es un conjunto de aplicaciones que permiten a un programador desarrollar sus propios programas informáticos haciendo uso de sus conocimientos lógicos y lenguajes de programación. Como ejemplo se puede mencionar a los editores de texto, compiladores, intérpretes, depuradores y entornos de desarrollo integrados.

1.5.3 Software de aplicación

Son los programas que permiten realizar tareas específicas en el sistema. A diferencia del software de sistema, el software de aplicación está enfocado en un área específica para su utilización. La mayoría de los programas que se utilizan diariamente pertenecen a este tipo de software, ya que permiten realizar

diversos tipos de tareas en el sistema. Ejemplos de estos son los editores (Photoshop para el Diseño Gráfico), paquetes integrados. (Ofimática: Word, Excel, PowerPoint), software educativo, software médico y software de gestión.

1.6 Software de gestión

Es un software que permite gestionar todos los procesos de negocio de una empresa de forma integrada (11). Facilita la toma de decisiones basándose en datos objetivos y no conjeturas ya que se cuenta con toda la información de la empresa disponible y organizada.

1.6.1 Tipos de software de gestión

En la rama de la informática existen diferentes tipos de software de gestión que ayudan a las empresas en la organización y control de todos sus procesos, logrando que la toma de decisiones sea de forma rápida y concisa. Algunos de estos softwares son los siguientes:

- Sistema de gestión documental
- Sistema de gestión de calidad
- Sistema de gestión empresarial

1.6.1.1 Sistema de gestión documental

Un software de gestión documental es una aplicación que permite el tratamiento, conservación, publicación y trabajo sobre documentos electrónicos (ya sean documentos escaneados o que se haya creado originalmente en digital). El objetivo de este es almacenar y gestionar documentos electrónicos, que pueden ser originales o copias digitales de documentos en formato papel. (12)

Características de una gestión documental

Los sistemas de gestión documental ofrecen a las empresas diversas características para mejorar la organización de la documentación generada en su flujo de trabajo, dentro de las cuales se encuentran la Indexación y búsqueda que permite una búsqueda y recuperación rápida de dichos documentos; el Almacenaje de metadatos para guardar información identificativa del documento que se ha guardado y el

Control de versiones, permitiendo recuperar versiones anteriores. Otras de las características de los sistemas de gestión documental es que incluyen módulos para la gestión de los permisos y el acceso a los documentos, documentos sólo accesibles para determinados grupos de usuarios y además permiten la integración con programas ERP de gestión. (12)

Todas brindan en buena medida una serie de mecanismos para lograr que la gestión documental dentro de las empresas se haga de manera más eficiente.

Fases de una gestión documental

Los sistemas de gestión documental para su correcto funcionamiento constan de tres fases (12):

- Captura de documentos en un gestor documental: escanear documentos de papel para obtener imágenes digitales.
- Recuperación de documentos: recuperación rápida y flexible de documentos en base a diferentes búsquedas.
- Colaboración: un documento debe ser accesible por los usuarios que tienen permisos para modificarlos para que puedan revisar su contenido y trabajar colaborativamente en ellos.

Ventajas de implantar una gestión documental (12)

- Ahorro en costes de gestión de documentos.
- Distribución eficiente de la información entre los equipos de trabajo.
- Acceso y consulta rápida de los documentos.
- Seguridad y fiabilidad en la protección de documentos.
- Mejora la productividad al obtener un acceso ágil a la información.
- Potencia la gestión del conocimiento en los equipos.
- Mejora el trabajo colaborativo.

1.6.1.2 Sistema de gestión de calidad

Un sistema de gestión de calidad es una herramienta ideal para aquellas organizaciones que desean que sus productos y servicios cumplan con los máximos estándares de calidad y así lograr y mantener la satisfacción de sus clientes. (13)

Un buen sistema para la calidad no debería ser más que una nueva forma de hacer el trabajo, que le permita a la organización lograr sus objetivos de empresa, agregar valor a sus productos y servicios, satisfacer a sus clientes y con ello desarrollar y mantener ventajas competitivas, que la diferencien de sus competidores. (14)

Los sistemas de gestión de calidad se utilizan para (15):

- Identificar sus fortalezas y debilidades.
- Fijar sus políticas y objetivos de calidad.
- Documentar y registrar sus procesos.
- Identificar y asignar los recursos necesarios.
- Sistematizar sus operaciones.
- Actuar sobre las desviaciones.
- Sistematizar la mejora continua.

La estructura fundamental del sistema de calidad según la ISO 9001 / 2000¹ está constituida por un sistema de procesos que interactúan entre sí, que operan de acuerdo con parámetros, criterios y métodos establecidos, se monitorean a través de seguimiento, medición y análisis y se controlan de manera eficaz a través de la implementación de acciones que permiten alcanzar los resultados y la mejora continua de esos procesos. (15)

Los sistemas para la gestión de la calidad presentan ventajas para las empresas (14):

- La organización se asegura que funciona bien y de esta forma puede cumplir los objetivos propios de la institución.

¹ ISO 9001/2000: Requisitos de los Sistemas de Gestión de la Calidad.

- Se cuenta con un sistema permite gestionar, con calidad, el desarrollo de sus actividades. El sistema permite analizar el desempeño de forma integral y, además, poder detectar las oportunidades de mejora.
- La forma de organizarse para hacer el trabajo es mejor y más simple.

1.6.1.3 Sistemas de gestión empresarial

Los sistemas de gestión empresarial son un grupo de aplicaciones utilizadas en las empresas para la elaboración de cada paso administrativo de la misma. La estructura de estos sistemas es elaborada de forma piramidal, de tal manera es probable identificar con suma facilidad la importancia que tiene cada uno de los datos, así como que su utilización corresponda con la importancia de los mismos dentro del proceso de producción. (16)

Estos sistemas están elaborados bajo el concepto de potencializar cada uno de los puntos a través de los cuales es elaborado el proceso productivo de la empresa, tanto desde los recursos humanos como los recursos materiales. El desarrollo de nuevas tecnologías, ha permitido impulsar softwares completamente especializados para la gestión de bases de datos, la planificación, la verificación apropiada de procedimientos, así como de la correcta toma de decisiones a través de la exposición de los recursos entre todas las funciones que hacen parte del habitual proceso empresarial en cada negocio. (16)

El software empresarial tiene como objetivo lograr una producción de recursos mucho más efectiva a partir del adecuado manejo de la información, del cruce correcto de la misma y de su uso rápido, eficaz y seguro a través de las redes. Las herramientas para la correcta gestión de la información que se utilizan en los sistemas de gestión empresarial hacen posible el control, la planeación, la organización y dirección de cada uno de los procesos de la empresa durante todo el recorrido del producto.

Con un software para la gestión empresarial se tiene a la empresa bajo control y se incrementa la calidad de los servicios y productos que ofrece la entidad. La implantación de estos conlleva a la eliminación de barreras ínter departamentales, la información fluye por toda la empresa eliminando la improvisación por falta de información. Los sistemas empresariales se caracterizan por ser: integrados, con una base de datos centralizada, modulares, configurables y flexibles. (17)

Los sistemas de gestión empresarial ofrecen a las empresas diversas ventajas y desventajas. (17)

Ventajas de un sistema empresarial:

- Optimización de los procesos empresariales.
- Acceso a información confiable, precisa y oportuna.
- La posibilidad de compartir información entre todos los componentes de la organización.
- Eliminación de datos y operaciones innecesarias.
- Reducción de tiempo y costos de los procesos.

Desventajas de un sistema empresarial:

- La instalación del sistema puede ser muy costosa.
- El éxito depende de las habilidades y la experiencia de la fuerza de trabajo.
- Los sistemas pueden ser difíciles de usarse.

1.7 Proceso de planificación en el departamento de Ingeniería de Software

En el departamento de Ingeniería de Software de la facultad 3 en la Universidad de las Ciencias Informáticas, la planeación se concibe como un proceso permanente, orientado al cumplimiento de los objetivos que se tracen en cada una de las áreas de resultados claves (Profesional competente comprometido con la Revolución, Colectivo revolucionario de excelencia, Impacto económico y social y Gestión de la educación superior). Para cumplir con estos objetivos se requiere planificar los resultados a realizar por los profesores en cada una de las áreas. El proceso de planeación se concreta con la elaboración del plan de trabajo, el cual hace explícitas las actividades a desarrollar por los trabajadores para dar cumplimiento a los resultados trazados.

1.8 Análisis de soluciones existentes

A continuación se hace una caracterización de los diferentes sistemas homólogos estudiados para el desarrollo de la investigación, de los cuales se señalan sus beneficios y deficiencias.

Sistema de Gestión de la Actividad Docente en la Universidad de León

Este sistema es una aplicación *desktop* que permite la gestión de información sobre la carga docente en la Universidad de León, como pueden ser: controlar las altas y bajas de los profesores y de las asignaturas que imparten, controlar las modificaciones del plan docente existente, conocer disponibilidad de profesores, listar las asignaturas existentes por áreas y brindar información a los profesores sobre todo lo que ocurre en función de la docencia. Además brinda información sobre las actividades que se desarrollan en la universidad en las diferentes esferas (cultura, deportes, formación, entre otras) (18). Esta herramienta fue estudiada para el desempeño de la gestión en la planificación y evaluación de los profesores, lo que en este análisis para la propuesta se concluye que solo resuelve problemas referentes a la docencia.

Entre las principales desventajas que presenta este sistema se encuentran:

- No se gestiona la planificación de las actividades y el plan de resultados anual de cada profesor, por lo que no se puede llevar un control y seguimiento de las actividades a cumplir en un periodo determinado por un profesor.
- El sistema no es capaz de dar una propuesta del cumplimiento de las actividades del profesor.

Una vez terminado el estudio de este sistema se puede llegar a la conclusión de que no ofrece las funcionalidades necesarias para los objetivos que se persiguen con el desarrollo de la investigación.

P-TRAB

Primera herramienta orientada al modelo de Planificación por Objetivos (PO) cubano. Toda una revelación en su momento ya que resolvía perfectamente las necesidades de los planificadores según la forma en que aplicaban para entonces la PO. Este sistema fue desarrollado principalmente para las entidades pertenecientes a las FAR, diseñado solamente para MS-DOS. Permite el registro de los objetivos con sus actividades de aseguramiento, pero presenta muchas desventajas en cuanto a escalabilidad del sistema, accesibilidad y disposición de la información. Este sistema no se pensó para brindar la información generada en el proceso de planificación mediante una red. No es posible acceder a la información deseada en el momento deseado y aunque cuenta con un mecanismo de intercambio de información, basado en la

generación de ficheros con extensión dbf² a partir de la base de datos descentralizada que presenta, el solo hecho de no estar disponible en una red se salta una de las principales cuestiones con que debe contar un sistema informático para la planificación por objetivos. El sistema no es capaz de dar una propuesta de cumplimiento de las actividades del profesor. (19)

SIPAC

SIPAC es la herramienta para la planificación más completa que existe. Mantiene un alcance bastante amplio en cuanto a la planificación por objetivos en las entidades, pensado como sistema multiplataforma, escalable y con prestaciones para funcionar en red. Permite la planificación mediante el plan de trabajo individual y se puede conocer el avance de las actividades planificadas. Cuenta con un mecanismo que posibilita notificar a los trabajadores sobre la realización de una tarea. Realiza reportes sobre la cantidad de actividades cumplidas e incumplidas en un determinado plan. SIPAC cuenta con un mecanismo para efectuar el intercambio de información de diversas maneras. La vía más sencilla es donde exista una red y una base de datos centralizada, donde el acceso a la información está restringido totalmente por la compartimentación de la misma. Otras formas son diferentes bases de datos tributando información entre ellas mediante réplicas *online*, o mediante la réplica *offline* para el caso donde no exista una red entre las bases de datos de las partes que participan en el intercambio de información. (20)

La principal desventaja que presenta este sistema es que:

- No permite asociar las actividades de los planes mensuales del profesor con sus resultados anuales.

1.9 Valoración del estudio realizado

Para el desarrollo de la investigación se hizo un estudio de las diferentes clasificaciones de software, identificándose que el sistema a desarrollar está dentro de la clasificación de software de aplicación, ya que se diseña con el fin de que el usuario pueda realizar uno o más trabajos, permitiéndole ser más sencillo y ayudándole a solucionar problemas existentes. También se hace un estudio de diversos software de gestión, que ayudan a las empresas a lograr sus objetivos, obteniendo como resultado que el software ideal para

²Dbf: Se utiliza para los archivos de base de datos implementados por el software del sistema de gestión de base de datos dBASE para almacenar, recuperar y hacer referencia a los datos guardados en el formato DBF.

resolver los problemas de planificación en el departamento, es un software de gestión empresarial, que ayude en la organización, dirección, planificación y control de sus procesos.

Luego de haber realizado el estudio de diferentes sistemas para la evaluación de los profesores del departamento de Ingeniería de Software de la facultad 3, se evidencia la inexistencia de un sistema que cumpla con los requisitos necesarios para darle solución al problema planteado. La herramienta SIPAC gestiona una serie de requisitos necesarios para el departamento, pero presenta como desventaja que no permite asociar las actividades con los resultados del profesor, a pesar de esto, se tomaron un grupo de ideas de SIPAC entre ellas la forma de gestionar las actividades y los planes de trabajo. El estudio de estos sistemas permitió conocer su funcionamiento, lo cual sirvió de ayuda en el análisis, diseño e implementación de la solución. Por todo lo expresado anteriormente se hizo necesario el desarrollo de un sistema adaptado a las necesidades del departamento de Ingeniería de Software de la facultad 3.

1.10 Metodologías para el desarrollo del software

Una metodología es un conjunto integrado de técnicas y métodos que permiten abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo. Las metodologías se basan en una combinación de los modelos de proceso genéricos (cascada, incremental, entre otros). Definen artefactos, roles y actividades, junto con prácticas y técnicas recomendadas. (21)

1.10.1 Proceso Unificado Ágil (AUP-UCI)

El Proceso Unificado Ágil (AUP) es una versión simplificada del Proceso Unificado de Rational (RUP) desarrollada por Scott Ambler. Describe un enfoque simple, fácil de entender del desarrollo de software de aplicación de negocios usando técnicas y conceptos ágiles (21). AUP es una metodología flexible, lo que posibilita que pueda ser modificada y adaptada de acuerdo a las necesidades del software en desarrollo, además ha sido la metodología estandarizada para guiar los procesos de desarrollo en la UCI razón por la cual se ha seleccionado en la presente investigación.

La variación UCI del Proceso Unificado Ágil (AUP-UCI) propone tres fases: Inicio, Ejecución y Cierre. Para darle solución al problema se trabajará en la fase de Ejecución donde se ejecutan las actividades requeridas para desarrollar el software.

AUP-UCI propone siete disciplinas para la fase de Ejecución: Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas internas, Pruebas de liberación y Pruebas de aceptación. Para el desarrollo del software se trabajará en las disciplinas mencionadas anteriormente, generando nueve artefactos (Modelo conceptual, Historias de usuario, Prototipos de interfaz de usuario, Diagramas de clases del diseño, Diagramas de secuencia, Modelo de datos, Diagrama de componentes, Diagrama de despliegue, Diseño de casos de prueba).

1.11 Herramientas y tecnologías

1.11.1 Lenguaje de Modelado (UML 2.0)

UML son las siglas de Lenguaje Unificado de Modelado. Se trata de un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software (22).

Este lenguaje se centra en la representación gráfica de un sistema. Es un lenguaje de modelado para la especificación, visualización, construcción y documentación de los artefactos de un proceso. (23)

1.11.2 Herramienta CASE (Visual Paradigm 8.0)

Las herramientas CASE (*Computer Aided Software Engineering* o Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores. (24)

Visual Paradigm es una herramienta CASE 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, permitiendo la captura de requisitos, análisis, diseño e implementación. Visual Paradigm 8.0 permite el modelado del sistema ayudando a una rápida construcción del mismo, posee un soporte multiplataforma y proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. (25)

Emplea las últimas notaciones de UML, ingeniería inversa, generación de código, exportación e importación XML. Soporta aplicaciones web, exporta en formato HTML, está disponible en varios idiomas, es fácil de instalar y fácil de actualizar. Los desarrolladores pueden diseñar la documentación del sistema con plantillas de diseño. El analista de sistema puede estimar las consecuencias de los cambios con los diagramas de análisis de impacto, como la matriz y el diagrama de análisis. (26)

1.11.3 Lenguajes de programación

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo (27). En esta sección se especifican los diferentes lenguajes de programación empleados en la realización del producto.

1.11.3.1 Lenguaje de programación del lado del servidor

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. Los lenguajes del lado servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son el ASP, JSP, PERL y PHP. (28)

PHP 5.4.12

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor. Es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Este lenguaje es publicado bajo la licencia PHP, compatible con la Licencia Pública General de GNU. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión php. (29)

Entre sus ventajas se pueden encontrar:

- Tiene capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad. (29)

- Es un lenguaje multiplataforma, con capacidad de expandir su potencial utilizando módulos y además no requiere definición de tipos de variables. (30)
- Es fácil de aprender, posee una amplia documentación en su página oficial (www.php.net), entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda y también tiene manejo de excepciones. (31)
- Se caracteriza por ser un lenguaje muy rápido y permite las técnicas de Programación Orientada a Objetos. (32)
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos y cuenta con una biblioteca nativa de funciones sumamente amplia e incluida. (33)

1.11.3.2 Lenguaje de programación del lado del cliente

Los lenguajes de programación del lado del cliente se ejecutan en el navegador del usuario. Las páginas del cliente son muy dependientes del sistema donde se están ejecutando y esa es su principal desventaja. Cada navegador tiene sus propias características y lo que puede funcionar en un navegador puede no funcionar en otro. Como ventaja se puede decir que ofrecen respuestas inmediatas a las acciones del usuario y permiten la utilización de recursos de la máquina local. (34)

ExtJs 5

ExtJS es una biblioteca de JavaScript usada en el desarrollo de la capa de presentación de las aplicaciones web. Fue desarrollada por Sencha y tiene la gran funcionalidad de crear interfaces de usuario bastante funcionales. Se recomienda el uso del navegador web Firefox en conjugación con su extensión Firebug, la cual permite la inspección de código javascript y otros elementos del DOM. (35)

Esta biblioteca incluye (35):

- Componentes UI (interfaces de usuario) del alto performance y personalizables.
- Modelo de componentes extensibles.
- Un API (interfaz de programación de aplicaciones) fácil de usar.
- Licencias Open Source (GPL) y comerciales.

Ventajas (35):

- Una de las grandes ventajas de utilizar ExtJS es que permite crear aplicaciones complejas utilizando componentes predefinidos.
- Evita el problema de tener que validar el código para que funcione bien en cada uno de los navegadores.
- El funcionamiento de las ventanas flotantes lo pone por encima de cualquier otro.
- Permite que exista un balance entre el Cliente – Servidor, posibilitando que la carga de procesamiento se distribuya, permitiendo que el servidor al tener menor carga, pueda manejar más clientes al mismo tiempo.
- Eficiencia en la red, disminuyendo el tráfico en la red pues las aplicaciones cuentan con la posibilidad de elegir que datos desea transmitir al servidor y viceversa.
- Comunicación asíncrona, en este tipo de aplicación el motor de render³ puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente lo note.

1.11.4 Sistema Gestor de Bases de Datos

MySQL 5.6.12 es un sistema de gestión de base de datos relacional, lo que permite velocidad y flexibilidad, en el cual los datos están almacenados en tablas separadas enlazadas entre sí. Tiene licencia dual, pudiéndose usar de forma gratuita bajo licencia GNU/GPL (Licencia Pública General) o bien adquiriendo licencias comerciales de MySQL AB en el caso de no desear estar sujeto a los términos de la licencia GPL. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. (36)

Las principales características de este gestor de base de datos son las siguientes:

³ Render: Un **motor de render** es un software que toma contenido marcado (como HTML, XML, archivos de imágenes, etc.) e información de formateo (como CSS, XSL, etc.) y luego muestra el contenido ya formateado en la pantalla de aplicaciones. ExtJS como motor render permite crear aplicaciones mediante JavaScript, controla el cliente y se encarga de enviar y obtener información del servicio.

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo y posee vistas actualizables. (37)
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo. (38)
- Dispone de una Interfaz de Programación de Aplicaciones (API's) en gran cantidad de lenguajes (C, C++, Java, PHP). (39)
- Posee una gestión de usuarios y contraseñas, manteniendo muy buen nivel de seguridad en los datos y tiene soporte a multiplataforma y a SSL. (39)

1.11.5 Entorno de Desarrollo Integrado (PhpStorm 8.0)

PhpStorm es un IDE de programación desarrollado por JetBrains. Es uno de los entornos de programación más completos de la actualidad, permite la gestión de proyectos fácilmente y proporciona un fácil autocompletado de código. Es un editor de código inteligente, ofrece un excelente soporte para PHP (incluyendo las últimas versiones de idioma y marcos), HTML, JavaScript, CSS, Sass. Resalta los nombres de las funciones y clases, identifica variables, encuentra posibles errores, hace refactorización de php y tiene cientos de combinaciones de teclas que permiten programar sin casi tocar el mouse. PhpStorm tiene un excelente soporte para el marco de trabajo Symfony2. (40)

1.11.6 Marco de Trabajo

Symfony es un marco de trabajo PHP que permite utilizar la arquitectura MVC (Modelo - Vista - Controlador). Fue escrito para ser utilizado sobre la versión 5 de PHP ya que hace uso de la orientación a objetos que caracteriza a esta versión. (41)

Symfony 2.6.3 integra bibliotecas para ser utilizadas, entre ellas se encuentra el ORM llamado Doctrine, el cual es el encargado de la comunicación con la base de datos. Incluye otro marco de trabajo bastante conocido llamado Twig que constituye un poderoso motor de plantillas que permite separar el código PHP del código HTML. (41)

1.11.7 Servidor de Aplicaciones Web

Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. Se encarga de contestar

a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados (42). Básicamente consta de un intérprete HTTP el cual se mantiene a la espera de peticiones de clientes y le responde con el contenido según sea solicitado. El cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla (43).

Apache 2.4.4

Apache 2.4.4 es un servidor web que se utiliza para la visualización de sitios web. Su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa.

Presenta como características las siguientes (44):

- Es flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP.
- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Apache es una tecnología gratuita de código abierto.
- Es un servidor altamente configurable de diseño modular.
- Apache te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Soporta PHP como lenguaje de programación.

1.12 Conclusiones del capítulo

En el presente capítulo se elabora el marco teórico de la investigación, el cual permite obtener conocimiento sobre los diferentes softwares de gestión que se utilizan tanto nacional como internacionalmente, con el objetivo de conocer cómo se maneja el flujo de información en los sistemas de gestión. Se realiza un estudio sobre los sistemas homólogos relacionados con la planificación y propuesta de cumplimiento del plan de trabajo de los profesores, de los cuales la investigación se fortaleció con elementos de como se gestionan las actividades dentro de los mismos para diseñar la solución propuesta. El estudio de la metodología, los lenguajes, las herramientas y las tecnologías para el desarrollo del sistema confirmaron la validez del uso de las mismas para lograr los objetivos propuestos.

Capítulo 2: Características del sistema

2.1 Introducción

En el presente capítulo se elabora el modelo conceptual para conocer los principales conceptos que se deben tener en cuenta para la implementación del sistema, se realiza la captura de los requerimientos funcionales y no funcionales y se describen los requisitos mediante las historias de usuario. Se explican los patrones de diseño empleados en la solución y el patrón arquitectónico que se emplea en la construcción del software. Además, se modelan los diagramas que describen las relaciones entre las clases del diseño, el modelo de datos, los diagramas de secuencia y el diagrama de componentes.

2.2 Modelo conceptual

Un modelo conceptual define los principales conceptos de una solución, posibilitando de esta forma una mayor comprensión del negocio abordado. El modelo conceptual explica cuáles son y cómo se relacionan los conceptos relevantes en la descripción del problema. (45)

La Figura 1 representa el modelo conceptual, donde se manejan varios conceptos, los cuales tienen una estrecha relación entre ellos, centrándose en la planificación de las actividades a realizar por un profesor determinado, así como sus resultados para el año y el cumplimiento de los mismos. La descripción de los conceptos se encuentra en el artefacto modelo conceptual.

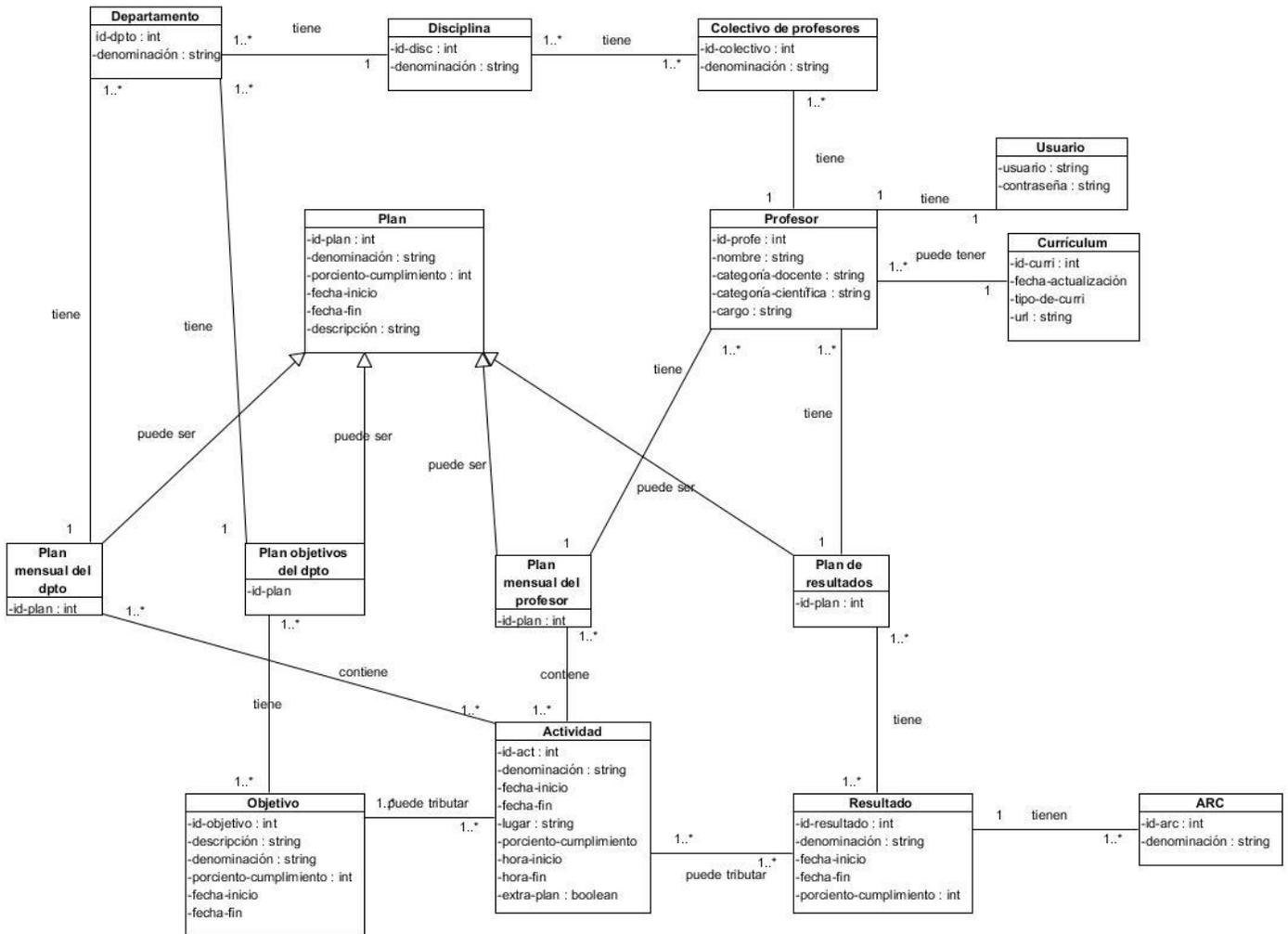


Figura 1: Modelo conceptual del departamento de Ingeniería de Software

2.3 Técnicas de captura de requisitos

Las técnicas de captura de requisitos permiten comprender las necesidades del cliente, analizando sus necesidades, negociando una solución razonable, especificando la solución sin ambigüedad y gestionando los requisitos para que se transformen en un sistema operacional. A continuación se explican las técnicas utilizadas para la captura de los requisitos:

Sistemas existentes: El punto de partida para muchos proyectos es muchas veces similar a otros sistemas ya existentes, que solucionan un problema semejante y pueden brindar nuevas ideas. Esto permite ahorrar tiempo en el proceso de captura de requisitos. En ciertas ocasiones estudiar sistemas que solucionen otro tipo de problemas puede contribuir a enriquecer las propuestas de solución. (46)

Esta técnica fue utilizada durante el análisis de varios sistemas existentes, tal es el caso de SIPAC, Sistema de Gestión de la Actividad Docente en la Universidad de León y PTRAB, analizando en todos los casos como se realiza el proceso de planificación de las actividades, tomando de ellos todas las características que se apegan a las necesidades del departamento de Ingeniería de Software.

Entrevistas: El contacto cara a cara con los usuarios a través de las entrevistas se puede considerar como una fuente para la obtención de requisitos. (47)

Se realizaron entrevistas informales al cliente con el objetivo de obtener una descripción global del proceso de planificación, permitiendo identificar las funcionalidades del sistema.

Revisión de la documentación: Esta técnica es utilizada para capturar requisitos, que luego deben ser validados por otras técnicas para su comprobación pues no se considera efectiva por sí sola. (48)

La revisión de la documentación permitió realizar un estudio sobre el proceso de planificación para un mejor entendimiento de la terminología que se maneja en el departamento, analizando documentos referentes al plan de resultados, al plan de objetivos y al plan de actividades.

2.4 Requisitos funcionales del sistema

Los requisitos son una especificación de lo que debe ser implementado, de cómo el sistema se debe comportar, de las propiedades y atributos del mismo (49). Seguidamente se muestran cuáles fueron los requisitos funcionales capturados para la implementación del sistema.

Tabla 1: Listado de requisitos funcionales

| Requisitos funcionales del sistema | |
|-------------------------------------------|------------------------|
| Gestionar departamento | |
| RF1 | Adicionar departamento |
| RF2 | Modificar departamento |

| | |
|-----------------------------------------------------------|-----------------------------------------|
| RF3 | Eliminar departamento |
| RF4 | Listar departamento |
| Gestionar disciplina | |
| RF5 | Adicionar disciplina |
| RF6 | Modificar disciplina |
| RF7 | Eliminar disciplina |
| RF8 | Listar disciplina |
| Gestionar colectivo de profesor | |
| RF9 | Adicionar colectivo de profesores |
| RF10 | Modificar colectivo de profesores |
| RF11 | Eliminar colectivo de profesores |
| RF12 | Listar colectivo de profesores |
| Gestionar profesor | |
| RF13 | Adicionar profesor |
| RF14 | Modificar profesor |
| RF15 | Eliminar profesor |
| RF16 | Listar profesor |
| Gestionar plan de objetivos del departamento (POD) | |
| RF17 | Adicionar POD |
| RF18 | Modificar POD |
| RF19 | Eliminar POD |
| RF20 | Listar POD |
| Gestionar plan de resultados | |
| RF21 | Adicionar plan de resultados |
| RF22 | Modificar plan de resultados |
| RF23 | Eliminar plan de resultados |
| RF24 | Listar plan de resultados |
| Gestionar plan mensual del departamento | |
| RF25 | Adicionar plan mensual del departamento |
| RF26 | Modificar plan mensual del departamento |

| | |
|--------------------------------------------|----------------------------------------|
| RF27 | Eliminar plan mensual del departamento |
| RF28 | Listar plan mensual del departamento |
| Gestionar plan mensual del profesor | |
| RF29 | Adicionar plan mensual del profesor |
| RF30 | Modificar plan mensual del profesor |
| RF31 | Eliminar plan mensual del profesor |
| RF32 | Listar plan mensual del profesor |
| Gestionar objetivo | |
| RF33 | Adicionar objetivos |
| RF34 | Modificar objetivos |
| RF35 | Eliminar objetivos |
| RF36 | Listar objetivos |
| Gestionar resultado | |
| RF37 | Adicionar resultado |
| RF38 | Modificar resultado |
| RF39 | Eliminar resultado |
| RF40 | Listar resultado |
| Gestionar actividad | |
| RF41 | Adicionar actividad |
| RF42 | Modificar actividad |
| RF43 | Eliminar actividad |
| RF44 | Listar actividad |
| RF45 | Obtener reporte de cumplimiento |
| Gestionar usuario | |
| RF46 | Adicionar usuario |
| RF47 | Modificar usuario |
| RF48 | Eliminar usuario |
| RF49 | Listar usuario |
| RF50 | Autenticación |
| Gestionar currículum | |

| | |
|-------------|----------------------|
| RF51 | Adicionar currículum |
| RF52 | Modificar currículum |
| RF53 | Eliminar currículum |
| RF54 | Listar currículum |

2.5 Historias de usuario

Las historias de usuario (HU) son utilizadas en las metodologías de desarrollo ágiles para la especificación de requisitos del software, utilizando el lenguaje común del usuario. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada HU es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas, además permiten responder rápidamente a los requisitos cambiantes. (50)

A continuación se describe la HU perteneciente al requisito Adicionar plan de resultados, para consultar el resto ver Anexos (13 - 23):

Tabla 2: Historia de usuario del requisito Adicionar plan de resultados

| Historia de usuario: Adicionar plan de resultados | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| Número: 21 | Nombre del requisito: Adicionar plan de resultados |
| Programador: Yaima | Iteración Asignada: 2 |
| Prioridad: Media | Tiempo Estimado: 0.5 |
| Riesgo en Desarrollo: Alto | Tiempo Real: 0.14 |
| <p>Descripción: Esta historia de usuario permite adicionar un nuevo plan de resultados. En caso de que los campos se llenen correctamente introduciendo caracteres válidos se adiciona un nuevo plan de resultados, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos y en caso de que haya campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos.</p> | |

Observaciones:

Prototipo de interfaz:

Adicionar plan de resultados

Denominación:

Descripción:

Fecha inicio:

Fecha fin:

Por ciento de cumplimiento:

Cancelar Aceptar

2.6 Requisitos no funcionales del sistema

Los requisitos no funcionales describen aspectos del sistema que son visibles por el usuario que no incluyen una relación directa con el comportamiento funcional del sistema, estos incluyen restricciones como el tiempo de respuesta (desempeño), la precisión, recursos consumidos, seguridad, entre otros. (51)

A continuación se muestran los requisitos no funcionales del sistema:

RNF1: Usabilidad

1. El sistema podrá ser utilizado por cualquier persona que posea conocimientos básicos en el manejo de la computadora.
2. El sistema deberá mostrar los mensajes y textos de la interfaz en el idioma español.
3. El sistema deberá contar con una interfaz de fácil entendimiento para que usuarios inexpertos puedan interactuar fácilmente con el software.

RNF2: Fiabilidad

1. Sólo se accederá a la base de datos desde la aplicación, nunca directamente desde el gestor de bases de datos.

RNF3: Rendimiento

1. Tiempos de respuesta no mayor de 10 segundos, al igual que la velocidad de procesamiento de la información.

RNF4: Seguridad

1. Confidencialidad: los usuarios se autentican para poder acceder al sistema según los permisos asignados.
2. Integridad: se garantiza la integridad de la información que se maneja en el sistema ya que solo podrá ser modificada por las personas autorizadas.
3. Disponibilidad: el sistema deberá estar disponible las 24 horas del día.

RNF5: Software

Para el cliente:

1. Navegador Mozilla Firefox 24.0 o superior
2. Sistema operativo Linux o Windows

Para el servidor:

1. Sistema operativo Linux
2. Servidor Apache 2.0
3. Servidor de base de datos PostgreSQL 9.1

RNF6: Hardware

Para el cliente:

1. Procesador Pentium IV a 3.00Ghz
2. 512 MB de memoria RAM

Para el servidor:

1. Procesador Intel Core i3 a 1.7GHz
2. 4Gb de memoria RAM

2.7 Técnicas de validación de requisitos

Para validar que los requisitos anteriormente identificados y descritos cumplan con las expectativas del cliente, se emplea las técnicas de validación de requisitos: Revisión de requisitos y Prototipado. Las cuales se explican a continuación (47):

Revisiones de requisitos: en la revisión formal el equipo de desarrollo conduce al cliente a través de los requerimientos, explicándole las implicaciones de cada uno. Los conflictos, contradicciones, errores y omisiones deben señalarse durante la revisión.

Estas revisiones fueron realizadas conjuntamente con el cliente para validar que los requisitos estaban acorde con lo que se necesitaba y verificando la comprensibilidad, rastreabilidad y adaptabilidad de cada uno de los requisitos de software.

Prototipado: el método del prototipado consiste en construir una maqueta del futuro sistema a partir de los requisitos recogidos en la especificación. Esta maqueta será evaluada por el cliente y usuarios para comprobar su corrección y completitud.

Esta técnica se utilizó en la construcción de prototipos para cada requisito, siendo evaluados por el cliente para comprobar que cumple con sus necesidades reales.

2.8 Modelo del diseño

2.8.1 Diagrama de clases

Los diagramas de clases especifican las diferentes clases que serán utilizadas en el sistema y las relaciones que existen entre ellas. A continuación se muestra el diagrama de clases perteneciente al requisito Gestionar plan de resultados y una breve descripción del mismo, para consultar el resto ver Anexos (1 - 12):

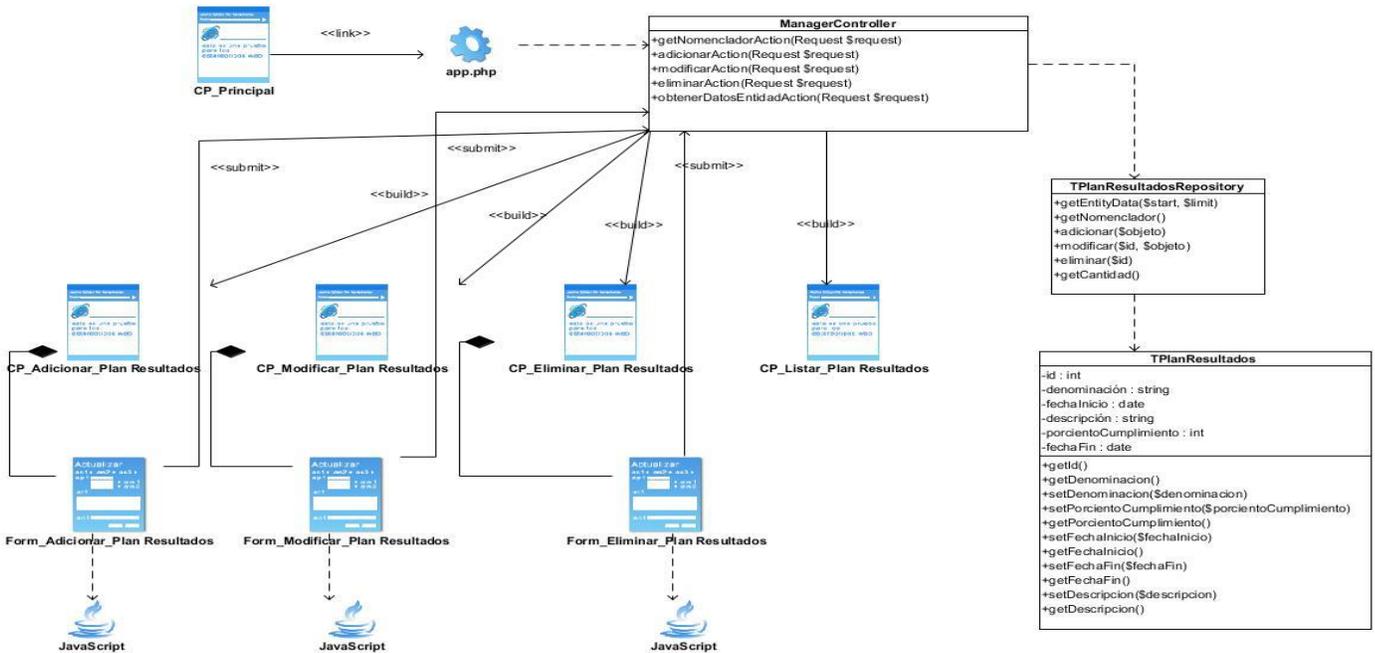


Figura 2: Diagrama de clases de la agrupación de requisitos del Gestionar plan de resultados

Las páginas clientes *CP_Agregar_Plan Resultado*, *CP_Modificar_Plan Resultado*, *CP_Eliminar_Plan Resultado* representan las páginas web encargadas de mostrar los formularios de información al usuario. La página servidor *app.php* es el servidor que se encarga de direccionar según las peticiones que haga el usuario. La clase *ManagerController* es la encargada de manejar la comunicación entre la vista y la clase entidad *TPlanResultados* que contiene toda la información del plan de resultados.

2.8.2 Diagrama de secuencia

Los diagramas de secuencia describen la interacción entre los objetos de una aplicación y los mensajes recibidos y enviados por los objetos. (52)

A continuación se muestra el diagrama de secuencia perteneciente al requisito Adicionar actividad y una breve descripción del mismo.

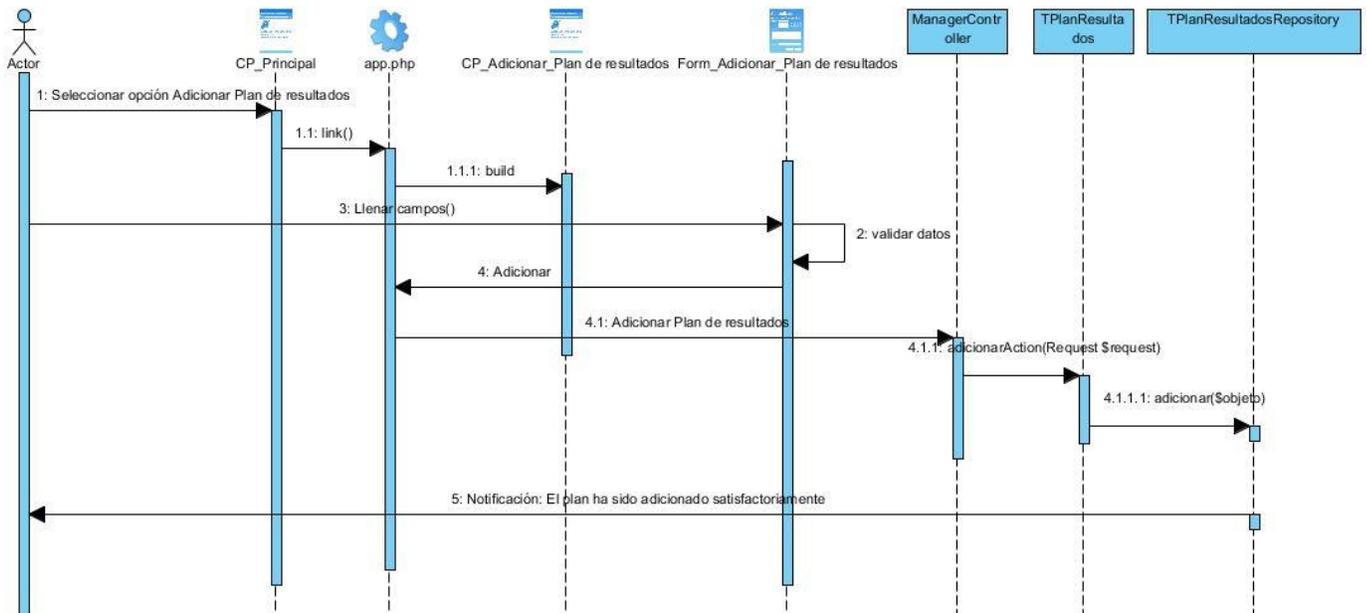


Figura 3: Diagrama de secuencia del requisito Adicionar plan de resultados

En el modelo se puede observar la *página cliente* principal, primeramente se selecciona la opción Plan de resultados y según la funcionalidad (Adicionar, Modificar o Eliminar) que se ejecute redirecciona hacia la *página cliente* indicada (CP_Adicionar_Plan de resultados, CP_Modificar_Plan de resultados o CP_Eliminar_Plan de resultados), luego se construye el formulario correspondiente, el cual al introducir los datos válidos los envía a la clase *ManagerController*, encargándose esta del proceso de enviar las acciones a la base de datos.

2.8.3 Modelo de datos

El modelo de datos es un artefacto que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades. Las entidades son conjuntos de datos formados por atributos, que son datos atómicos o indivisibles. A partir de estas entidades se pueden crear estructuras complejas de datos mediante la definición de relaciones entre estas. (53)

En el modelo de datos (Figura 4) de la solución para la evaluación de los profesores del departamento de Ingeniería de Software de la facultad 3, se definen las relaciones, ajustándose a las necesidades del sistema. El mismo se encuentra en la tercera forma normal y cuenta con 19 tablas.

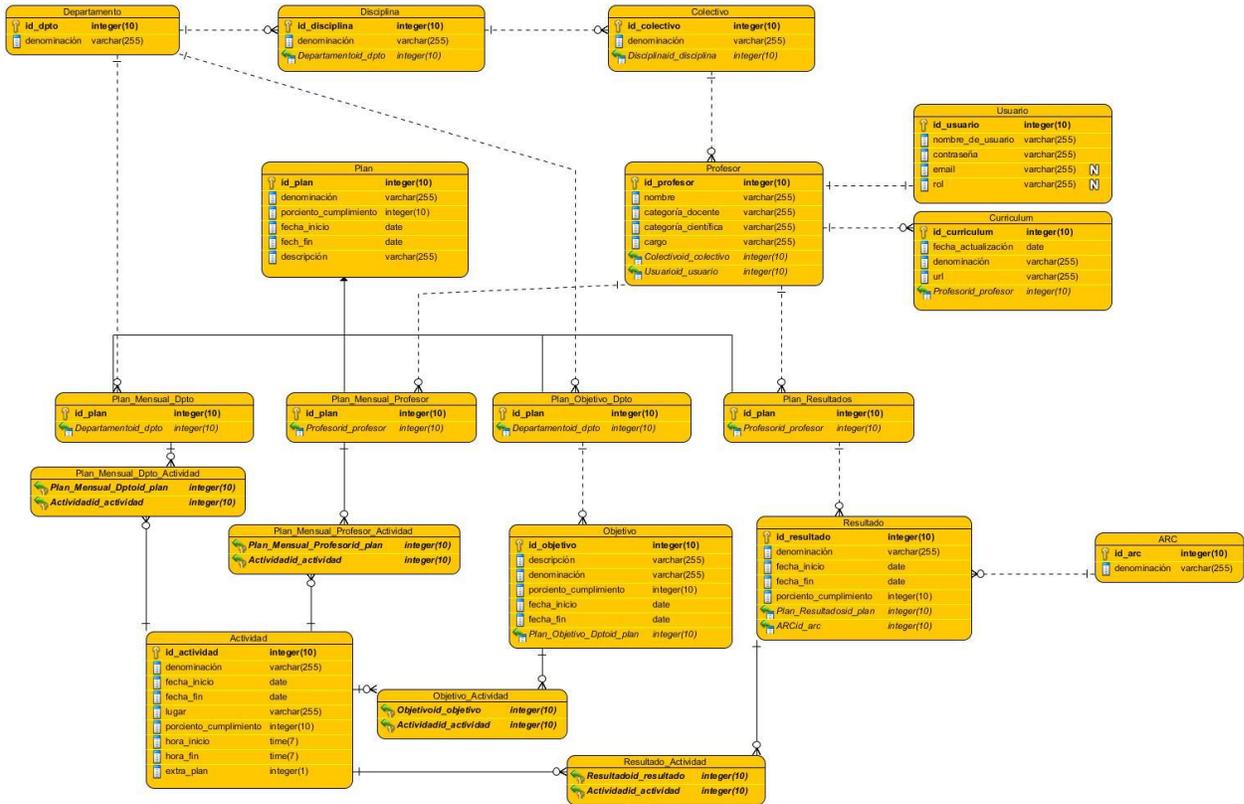


Figura 4: Modelo de datos de la solución

2.9 Patrón arquitectónico

Para la implementación de la solución se propone utilizar el estilo en capas (capa de presentación, capa de negocio, capa de acceso a datos y capa de datos) y el patrón arquitectónico Modelo - Vista – Controlador (MVC).

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC, separa la lógica de negocio de la interfaz de usuario, facilita la evolución por separado de ambos aspectos e Incrementa la reutilización y flexibilidad (54), está formado por tres niveles:

- Modelo: representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- Vista: transforma el modelo en una página web que permite al usuario interactuar con ella.

- Controlador: se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

El principio más importante del patrón arquitectónico MVC (Figura 5) es la separación del código del programa en tres capas, dependiendo de su naturaleza. La lógica relacionada con los datos se incluye en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador.

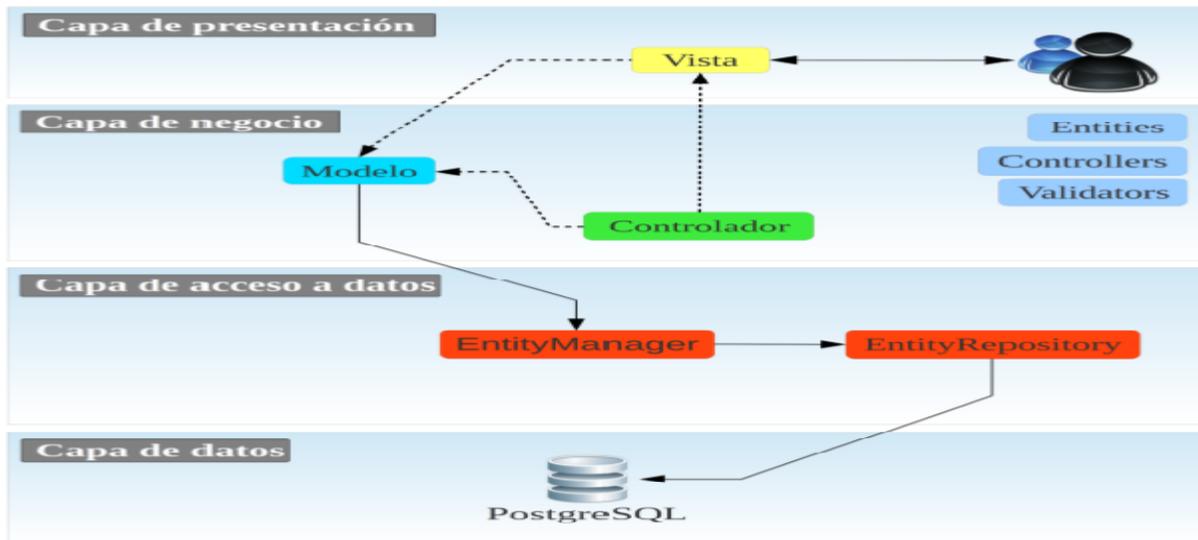


Figura 5: Diseño arquitectónico de Symfony

2.10 Patrones de diseño

Los patrones de diseño son soluciones para problemas típicos y recurrentes que se pueden encontrar a la hora de desarrollar una aplicación, brindan una solución ya probada y documentada a problemas de desarrollo de software y ayudan a estandarizar el código. (55)

A continuación se enuncian los diferentes patrones de diseño utilizados durante el proceso de desarrollo de la solución.

2.10.1 Patrones generales de software para asignación de responsabilidades (GRASP)

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. Los patrones no se proponen descubrir ni expresar nuevos principios de la ingeniería del software. A continuación se describen los patrones utilizados (56):

Experto en Información: este patrón consiste en asignar una responsabilidad al experto en información que es la clase que cuenta con la información necesaria para cumplir dicha responsabilidad.

Symfony 2 utiliza el ORM Doctrine 2 para realizar su capa de abstracción en el modelo, encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades. Este patrón se evidencia en la clase TPlanResultados.php, donde esta posee toda la información referente al plan de resultados.

Creador: consiste en crear una nueva instancia por la clase que tiene la información necesaria para realizar la creación del objeto, usa directamente las instancias creadas del objeto, almacena o maneja varias instancias de la clase o contiene o agrega la clase.

El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que debe conectarse con el objeto producido en cualquier evento. Este patrón se evidencia en las clases repositorio en los métodos de adicionar.

Ejemplo en la clase TPlanResultadosRepository.php

```

public function adicionar($objeto) {
    try {
        $em = $this->getEntityManager();
        $pr = new TPlanResultados();
        $dateFin = explode('/', $objeto->fechafin);
        $dateIni = explode('/', $objeto->fechaini);
        $dateFin = $dateFin[2] . '-' . $dateFin[1] . '-' . $dateFin[0];
        $dateIni = $dateIni[2] . '-' . $dateIni[1] . '-' . $dateIni[0];
        $profesor = $em->getRepository('AppBundle:TProfesor')->find($objeto->idprofesor);
        $pr->setDenominacion($objeto->denominacion)
            ->setPorcentajeCumplimiento($objeto->porcentajeCumplimiento)
            ->setFechaini(new \DateTime($dateIni))
            ->setFechafin(new \DateTime($dateFin))
            ->setProfesor($profesor);
        if ($objeto->descripcion !== '')
            $pr->setDescripcion($objeto->descripcion);
        $em->persist($pr);
        $em->flush();
        return array('codMsg' => 1, 'msg' => 'El plan ha sido adicionado satisfactoriamente.');
```

```

    } catch (Exception $e) {
        print_r('Excepción capturada: ' . $e->getMessage() . "\n");
    }
}

```

Figura 6: Método adicionar de la clase TPlanResultadosRepository

Alta cohesión: Symfony 2 permite asignar responsabilidades con una alta cohesión. La alta cohesión se refiere a asignar a las clases las actividades lo más relacionadas y a fin posibles. Una clase con baja cohesión realiza muchas actividades, es una clase muy cargada, almacena o maneja varias instancias de clases y contiene o agrega clases, es difícil de entender y de reutilizar. Esto se evidencia en la clase TPlanResultadosRepository.php donde mantiene una alta cohesión, debido a que solo tiene asignado la responsabilidad de realizar las funcionalidades que le corresponden.

Controlador: asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, entre otras). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Este patrón se evidencia en la clase ManagerController, encargada de controlar el flujo de eventos.

Bajo acoplamiento: el uso de este patrón garantiza asignar a cada clase la responsabilidad que le corresponde, obtener el menor número de relaciones y dependencias entre clases, de forma tal que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases y aumentar las posibilidades de reusabilidad de las mismas. Este patrón se evidencia en las clases repositorios que heredan de la clase Repository.

Ejemplo: TPlanResultadosRepository.php

```

<?php

namespace AppBundle\Repository;

use ...

class TPlanResultadosRepository extends EntityRepository

```

Figura 7: Patrón bajo acoplamiento en la clase TPlanResultadosRepository

2.11 Diagrama de componentes

En los diagramas de componentes se muestran los elementos de diseño de un sistema de software. Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema y el comportamiento de los servicios que estos componentes proporcionan y utilizan a través de las interfaces (57). A continuación se muestra el diagrama de componentes elaborado para los planes de trabajo y una breve descripción del mismo, ver todos los componentes del sistema en el Anexo 24.

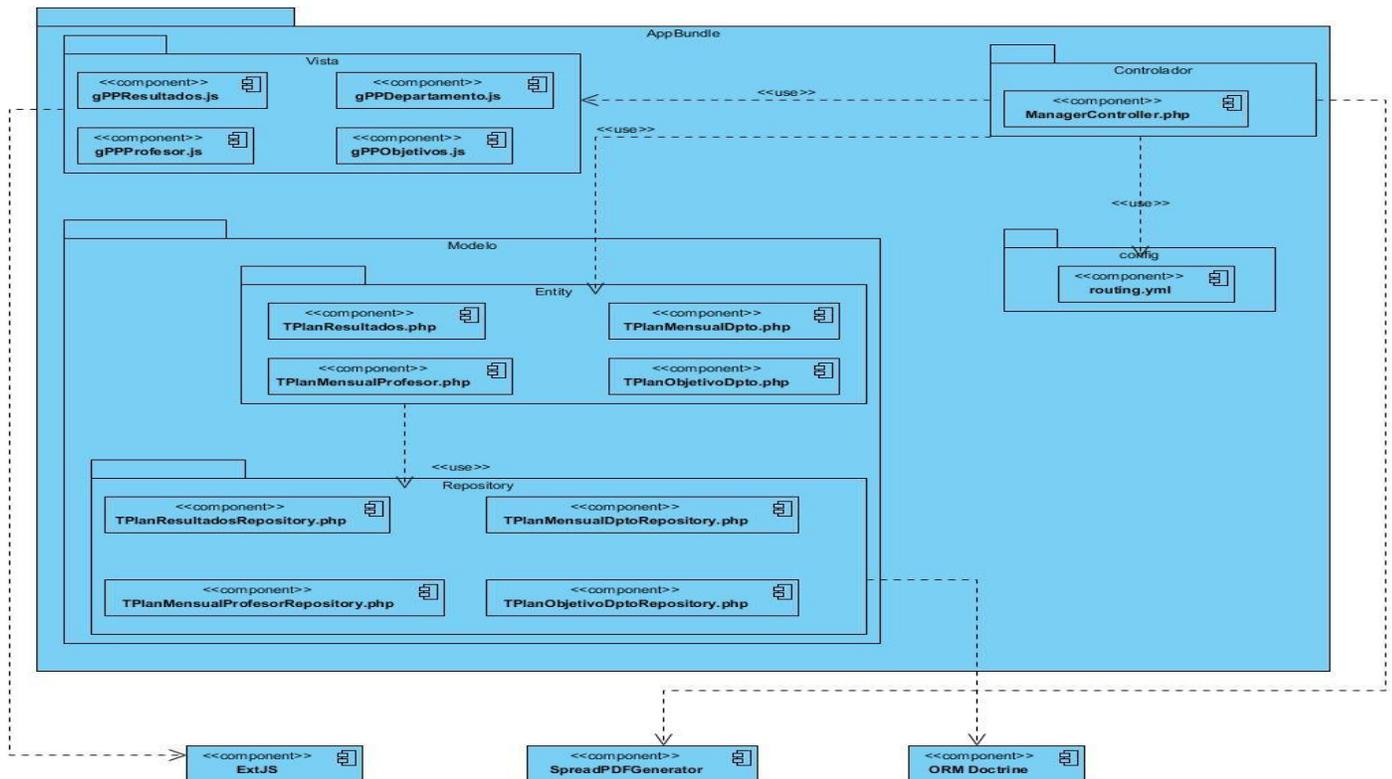


Figura 8: Diagrama de componentes

Para representar los distintos componentes de la solución implementada se diseñó el diagrama anterior que justifica la utilización del patrón arquitectónico Modelo – Vista – Controlador. El mismo está compuesto por un solo componente AppBundle encargado de manejar todo lo relacionado con el sistema, y este a su vez está asociado a otros componentes haciendo uso de ExtJs para las vistas, SpreadPDFGenerator para generar PDF y del ORM Doctrine como mapeador de objetos-relacional.

2.12 Validación del diseño mediante métricas

Antes de comenzar la disciplina de implementación es necesario comprobar la calidad del diseño elaborado, para ello se emplean métricas. Las mismas son una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado (58). Se emplean las métricas Tamaño operacional de clase (TOC) y Relaciones entre clases (RC), diseñadas para evaluar los siguientes atributos de calidad:

- Responsabilidad: consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto de la problemática propuesta.
- Complejidad de implementación: consiste en el grado de dificultad que tiene que implementar un diseño de clases determinado.
- Reutilización: consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.
- Acoplamiento: consiste en el grado de dependencia o interconexión de una clase o estructura de clase con otras, está muy ligada a la característica de Reutilización.
- Complejidad del mantenimiento: consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta pero fuertemente en los costes y la planificación del proyecto.
- Cantidad de pruebas: consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (unidad) del producto (componente, módulo, clase, conjunto de clases, entre otros) diseñado.

2.12.1 Métrica Tamaño operacional de clase

Tamaño operacional de clase (TOC): Está dado por el número de métodos asignados a una clase.

A continuación se muestran dos tablas enfocadas a un mejor entendimiento de la utilización de esta métrica.

Tabla 3: Métrica Tamaño operacional de clase

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

Tabla 4: Rango de valores para la métrica TOC

| Atributo | Categoría | Criterio |
|--------------------------------------|-----------|---------------------------------|
| Responsabilidad | Baja | \leq Promedio |
| | Media | Entre Promedio y 2^* Promedio |
| | Alta | $> 2^*$ Promedio |
| Complejidad de Implementación | Baja | \leq Promedio |
| | Media | Entre Promedio y 2^* Promedio |
| | Alta | $> 2^*$ Promedio |
| Reutilización | Baja | $> 2^*$ Promedio |
| | Media | Entre Promedio y 2^* Promedio |
| | Alta | \leq Promedio |

Resultados de la aplicación de la métrica TOC

La métrica TOC se aplicó a las clases controladoras, las entidades y a los repositorios para un total de 41 clases.

En el siguiente gráfico se muestra la representación de los resultados obtenidos agrupados en los intervalos definidos. El mismo refleja que la mayoría de las clases tienen de 1 a 5 procedimientos. Esto demuestra que el funcionamiento general del sistema está distribuido equitativamente entre las diferentes clases.

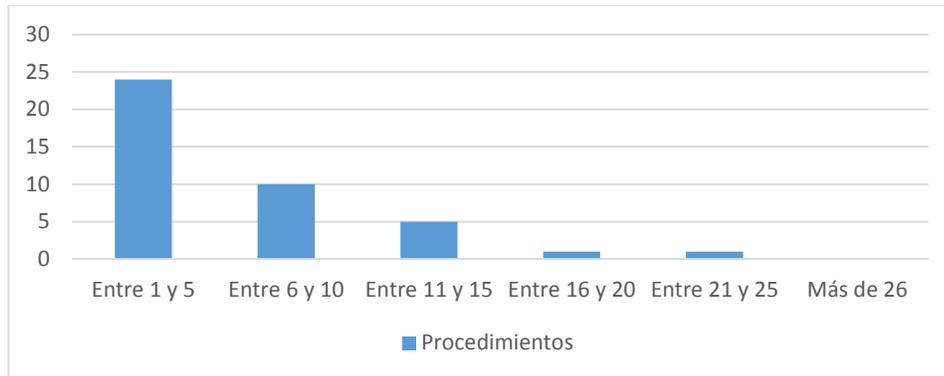


Figura 9: Representación de la evaluación de la métrica TOC

En la siguiente figura se muestran los resultados obtenidos en porcentaje agrupados en los intervalos definidos.

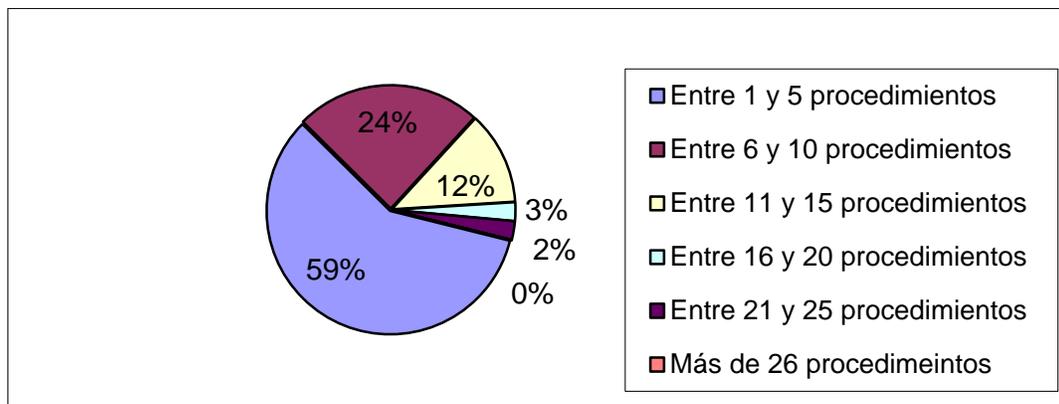


Figura 10: Representación en porcentaje de los resultados obtenidos en la evaluación de la métrica TOC

A continuación (Figura 11) se observa la representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo responsabilidad. Este gráfico muestra un resultado satisfactorio pues el 73% de las clases poseen una baja responsabilidad. Esta característica permite que en caso de fallos, como la responsabilidad está distribuida de forma equilibrada, ninguna clase es demasiado crítica como para dejar al sistema fuera de servicio.

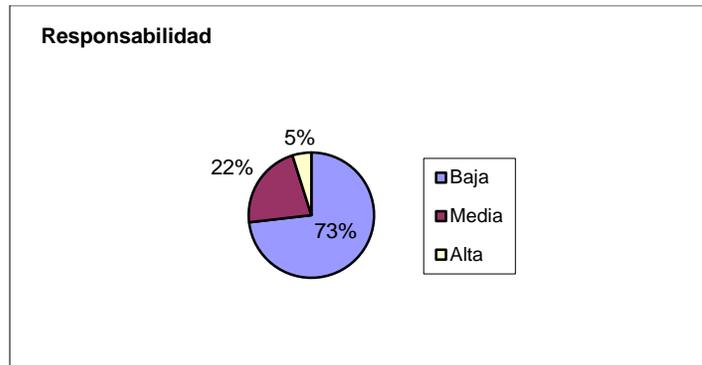


Figura 11: Resultados de la evaluación de la métrica TOC para el atributo responsabilidad

A continuación (Figura 12) se observa la representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo complejidad de implementación. Este gráfico muestra un resultado satisfactorio pues el 73% de las clases poseen una baja complejidad de implementación. Esta característica permite mejorar el mantenimiento y soporte de estas clases.

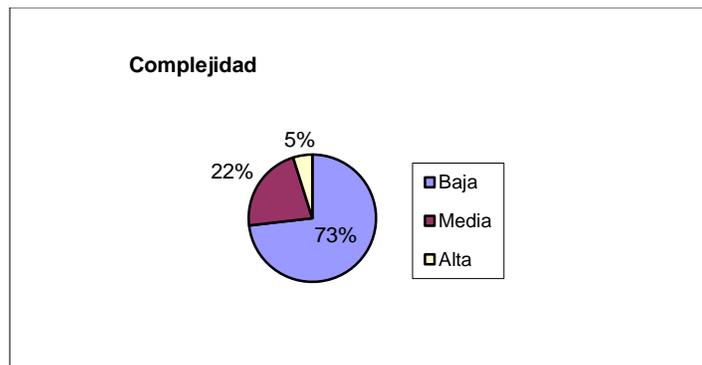


Figura 12: Resultados de la evaluación de la métrica TOC para el atributo complejidad

A continuación (Figura 13) se muestra la representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo reutilización. El diseño del sistema tiene un grado de eficiencia aceptable pues el 73% de las clases poseen una alta reutilización.



Figura 13: Resultados de la evaluación de la métrica TOC para el atributo reutilización

Analizando los resultados obtenidos de la aplicación de la métrica TOC, se puede concluir que el diseño del sistema tiene una calidad aceptable teniendo en cuenta los resultados arrojados por los atributos analizados. El 73% de las clases presentan una baja responsabilidad, una baja complejidad de implementación y una alta reutilización, lo cual demuestra que el resultado es satisfactorio.

2.12.2 Métrica Relaciones entre Clases

Relaciones entre clases (RC): está dada por el número de relaciones de uso de una clase con otras. A continuación se muestran las Tabla 5 y la Tabla 6 encaminadas a un mejor entendimiento de la utilización de esta métrica.

Tabla 5: Atributos de calidad evaluados por la métrica RC

| Atributo que afecta | Modo en que lo afecta |
|--------------------------------------|------------------------------------------------------------------------------------------------------------|
| Acoplamiento | Un aumento del RC implica un aumento del acoplamiento de la clase. |
| Complejidad del mantenimiento | Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase. |
| Reutilización | Un aumento del RC implica una disminución en el grado de reutilización de la clase. |
| Cantidad de pruebas | Un aumento del RC implica un aumento de la cantidad de pruebas de unidad necesarias para probar una clase. |

Tabla 6: Rango de valores para la métrica RC

| Atributo | Categoría | Criterio |
|--------------------------------------|-----------|------------------------------|
| Acoplamiento | Ninguna | 0 |
| | Baja | 1 |
| | Media | 2 |
| | Alta | >2 |
| Complejidad del mantenimiento | Baja | <= Promedio |
| | Media | Entre Promedio y 2* Promedio |
| | Alta | > 2* Promedio |
| Reutilización | Baja | > 2* Promedio |
| | Alta | Entre Promedio y 2* Promedio |
| | Media | <=Promedio |
| Cantidad de pruebas | Baja | <=Promedio |
| | Media | Entre Promedio y 2* Promedio |
| | Alta | > 2* Promedio |

Resultados de la aplicación de la métrica RC

La métrica RC se aplicó a las clases controladoras, las entidades y a los repositorios para un total de 41 clases. El promedio utilizado para evaluar el criterio es el resultado del cálculo del promedio de la columna Cantidad de relaciones entre clases, en este caso el promedio es 1,59.

El gráfico de la siguiente figura refleja que el 56% de las clases tienen una dependencia, y el 37% dos dependencias con otra clase. Este resultado es positivo pues demuestra que el 93% de las clases se encuentran dentro de los niveles aceptables.

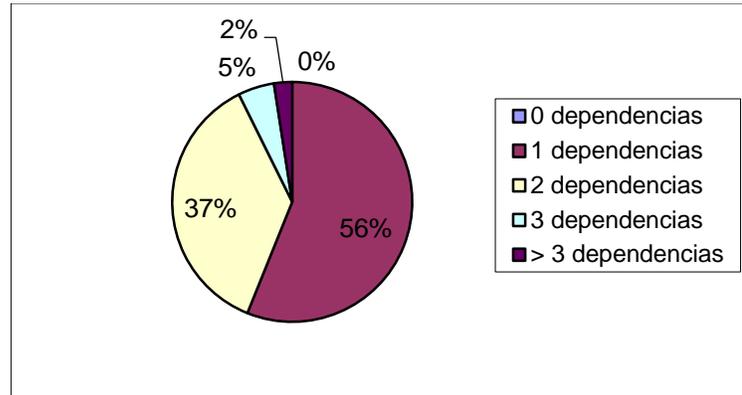


Figura 14: Representación en por ciento de los resultados obtenidos en los intervalos definidos según la métrica RC

En la siguiente figura se muestra la representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo acoplamiento. Se evidencia un bajo acoplamiento entre las clases pues el 56% de las clases presentan una dependencia con otra y el 37% dos dependencias con otra. Este resultado es muy favorable para el diseño del sistema pues al existir poca dependencia entre las clases aumenta el grado de reutilización del sistema.

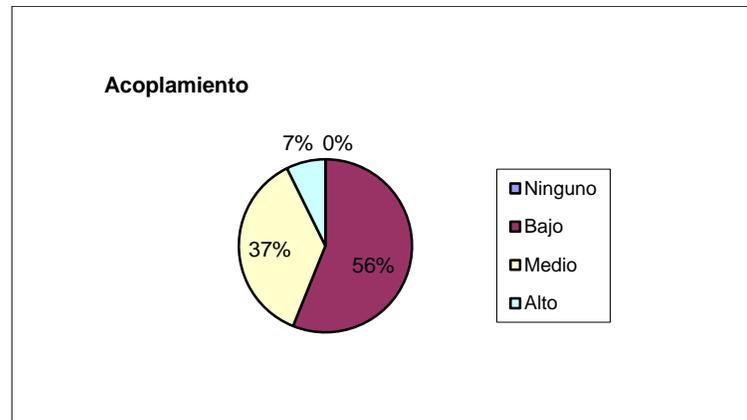


Figura 15: Resultados de la evaluación de la métrica RC para el atributo acoplamiento

En la siguiente figura se muestra la representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo complejidad de mantenimiento. El gráfico refleja un resultado aceptable del atributo pues el 93% de las clases presentan una baja complejidad de mantenimiento.

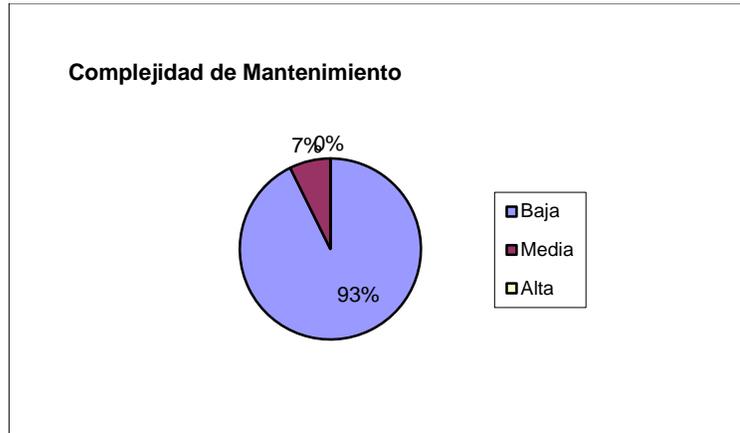


Figura 16: Resultados de la evaluación de la métrica RC para el atributo complejidad de mantenimiento

En la siguiente figura se muestra la representación de la incidencia de los resultados de la evaluación del atributo reutilización. Esto evidencia que el 93% de las clases poseen una alta reutilización lo que es un factor fundamental que debe ser tenido en cuenta en el desarrollo de software.

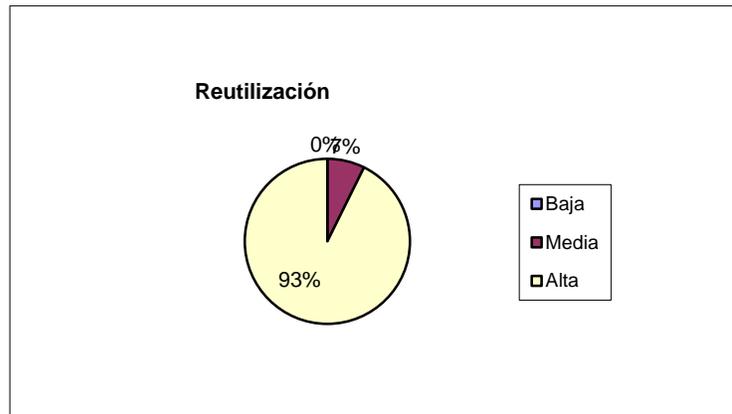


Figura 17: Resultados de la evaluación de la métrica RC para el atributo reutilización

En la siguiente figura se muestra la representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo cantidad de pruebas.

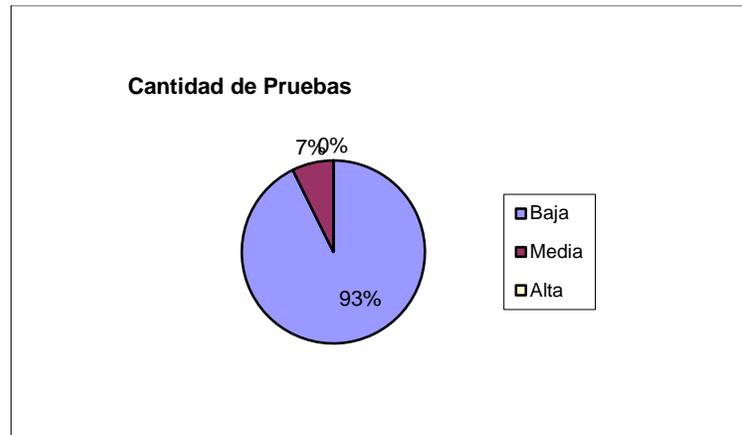


Figura 18: Resultados de la evaluación de la métrica RC para el atributo cantidad de pruebas

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica RC, se puede concluir que el diseño del sistema informático para la evaluación de los profesores del departamento de Ingeniería de Software tiene una calidad aceptable.

El 93% de las clases que conforman el sistema poseen menos de tres dependencias con otras clases. Los atributos de calidad se encuentran en un nivel satisfactorio; en el 56% de las clases el nivel de acoplamiento es bajo y en un 37% es medio.

La complejidad de mantenimiento y la cantidad de pruebas son bajas a un 91%, lo que representa valores favorables para el diseño realizado. Así mismo, existe un alto grado de reutilización al 91%, comportamiento también favorable para este atributo de calidad.

2.13 Conclusiones del capítulo

En el presente capítulo se enunciaron los aspectos fundamentales que se llevan a cabo durante el proceso de análisis y diseño del sistema para la planificación y evaluación de los profesores del departamento de Ingeniería de Software de la facultad 3. Se describieron los requisitos funcionales y no funcionales, propiciando la identificación de las funcionalidades del sistema. Se elaboraron los diagramas de clases y de secuencia correspondientes al sistema. Se construyó el modelo de datos con el propósito de hacer persistir los datos de manera coherente y eficaz y dar paso a la implementación del sistema. Se validó el diseño propuesto mediante las métricas Tamaño Operacional de Clases y Relaciones entre Clases, que permitieron demostrar que el sistema cuenta con un diseño satisfactorio.

Capítulo 3: Implementación y pruebas

3.1 Introducción

En el presente capítulo se desarrollan las disciplinas de implementación y pruebas según lo propone la metodología de desarrollo AUP-UCI. Se muestra el diagrama de despliegue que permite modelar las relaciones físicas de los distintos elementos que componen el sistema. Se describen las pruebas realizadas al sistema, que tienen como objetivo garantizar la calidad del mismo y el total cumplimiento de los requisitos establecidos con el cliente, además se muestran los resultados de las pruebas.

3.2 Implementación

Dentro del ciclo de vida de un software se encuentra la disciplina de implementación, es la más costosa por contar con muchas personas, herramientas y recursos involucrados en ella y es la que requiere además la mayor cantidad de tiempo. En esta se toman los requisitos y los productos de la fase de diseño y se implementan mediante las tecnologías apropiadas.

3.2.1 Estándares de codificación

Los estándares de codificación son un conjunto de reglas a seguir por los desarrolladores con el objetivo de establecer un orden y un formato común en el código fuente del software en desarrollo.

Para el desarrollo de la solución se utilizaron estándares de codificación, con el propósito de estandarizar las nomenclaturas en la implementación del sistema y obtener un producto estable y eficiente.

A continuación se muestra la notación utilizada con ejemplos:

Notación CamelCase: la palabra como tal CamelCase la podríamos traducir como el uso de mayúsculas y minúsculas (59). Esta notación indica que un conjunto de palabras se escribirán utilizando solo letras mayúsculas y minúsculas para diferenciarlas entre ellas (60).

Existen dos tipos de notación CamelCase:

UpperCamelCase: Las primeras letras de cada palabra en mayúscula.

- Ejemplo: *EstoEsUpperCamelCase*

lowerCamelCase: La primera letra de la primera palabra en minúscula y el resto de las primeras letras quedan en mayúsculas.

- Ejemplo: *estoEsLowerCamelCase*

Para los atributos y métodos se utiliza *lowerCamelCase* y para las clases del sistema *UpperCamelCase*.

Ejemplo de Notacion CamelCase:

Atributos de las clases: el nombre de los atributos de una clase comienza con la primera letra en minúscula, en caso de un nombre compuesto se emplea la notación *lowerCamelCase*. Si es una sola palabra se mantiene todo en minúscula.

Ejemplos:

- `private $nombre;`
- `private $categoriaCientifica;`

Métodos de las clases: el nombre de los métodos de una clase comienza con la primera letra en minúscula, en caso de un nombre compuesto se emplea la notación *lowerCamelCase*. Si es una sola palabra se mantiene todo en minúscula.

Ejemplos:

- ```
public function adicionar($objeto) {
 try {
 $em = $this->getEntityManager();
 $dpto = new TDepartamento();
 $dpto->setDenominacion($objeto->denominacion);
 $em->persist($dpto);
 $em->flush();
 return array('codMsg' => 1, 'msg' => 'El departamento ha sido adicionado satisfactoriamente.');
```

```
 } catch (Exception $e) {
 print_r('Excepción capturada: ' . $e->getMessage() . "\n");
```

```

 }
}
➤ public function getNomenclador() {
 try {
 $em = $this->getEntityManager();
 $q = $em->createQueryBuilder();
 $q->select('dep.id, dep.denominacion')
 ->from('AppBundle:TDepartamento', 'dep')
 ->orderBy('dep.denominacion', 'ASC');
 $res = $q->getQuery()->getArrayResult();
 return array('data' => $res, 'cant_filas' => self::getCantidad());
 } catch (Exception $e) {
 print_r('Excepción capturada: ' . $e->getMessage() . "\n");
 }
}

```

**Clases del sistema:** en el nombre de las clases las primeras letras de cada palabra comienzan con mayúscula, en caso de un nombre compuesto, se emplea la notación *UpperCamelCase*. Si es una sola palabra la primera letra en mayúscula.

Ejemplos:

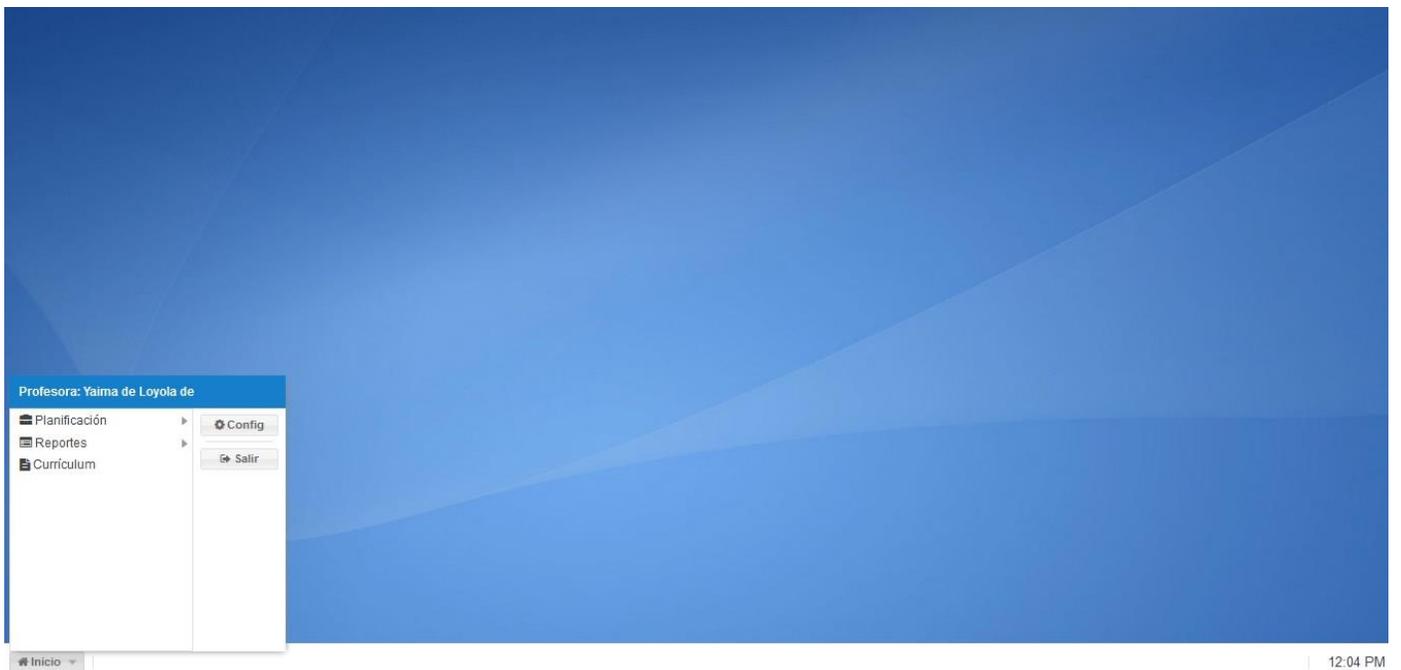
- **class** TProfesorRepository **extends** EntityRepository
- **class** TProfesor
- **class** ManagerController **extends** Controller

### 3.3 Interfaces del sistema

En la aplicación hay dos tipos de roles, el rol jefe de departamento que va a tener permisos de administrador, accediendo a todas las funcionalidades del sistema (Estructura, Planificación, Reportes y Currículum) y el rol profesor que solo tendrá acceso a aquellas funcionalidades relacionadas con su rol (Planificación, Reportes y Currículum). A continuación se muestra dos interfaces relacionadas con las funcionalidades asociadas a cada rol.



**Figura 19:** Pantalla de inicio asociada al rol jefe de departamento



**Figura 20:** Pantalla de inicio asociada al rol profesor

### 3.4 Diagrama de despliegue

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. (61)

La mayoría de las veces el modelado de la vista de despliegue estática implica modelar la topología del hardware sobre el que se ejecuta el sistema. Los diagramas de despliegue son fundamentalmente diagramas de clases que se ocupan de modelar los nodos de un sistema. Con este diagrama un ingeniero de software puede especificar la plataforma sobre la que se ejecuta el software del sistema y para que un ingeniero de sistemas pueda manejar la frontera entre el hardware y el software, cuando se trata de la relación entre hardware y software se utilizan los diagramas de despliegue para razonar sobre la topología de procesadores y dispositivos sobre los que se ejecuta el software. (61)

En el diagrama de despliegue se evidencia la presencia de 3 nodos: PC Cliente, Servidor Web de aplicaciones y Servidor de BD. La relación entre estos inicia cuando la PC Cliente solicita mediante el protocolo HTTP una petición al Servidor Web, este a su vez se comunica mediante TCP/IP con el Servidor de BD, donde se verifica que el usuario está autorizado a acceder a la aplicación y una vez que tenga los permisos necesarios se le muestra la información consecuente con su rol.

A continuación se muestra el diagrama de despliegue:



Figura 21: Diagrama de despliegue

### **3.5 Pruebas de software**

Pressman, declara que las pruebas de software son un elemento crucial para garantizar la calidad del producto y permiten validar las especificaciones, el diseño y la programación. Estas tienen como objetivo, además de descubrir errores, medir el grado en que el software cumple con los requerimientos definidos. (62)

Existen dos métodos de pruebas dentro de las pruebas unitarias, el método de prueba de Caja negra y el método de prueba de Caja blanca.

#### **3.5.1 Pruebas de Caja negra**

Las pruebas de caja negra, también denominadas pruebas de comportamiento se centran en los requisitos funcionales del software, o sea, la prueba de caja negra permite a un ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa.

La prueba de caja negra intenta encontrar errores en las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación. (62)

Para las pruebas de caja negra existen varias técnicas, algunas de ellas son (62):

Partición equivalente: permite examinar los valores válidos e inválidos de las entradas existentes en el software. Además descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. Esto permite reducir el número de casos de prueba a elaborar.

Análisis de valores límites: los errores tienden a darse más en los límites del campo de entrada que en el centro. Esta es una técnica que complementa la partición equivalente. En lugar de seleccionar cualquier elemento de una clase de equivalencia, lleva a elección de casos de prueba en los extremos de la clase.

Prueba de comparación: este tipo de pruebas se emplea cuando la fiabilidad del software es algo crítico, (por ejemplo cuando se desarrolla para aeronaves o plantas nucleares) varios equipos de ingeniería del software desarrollan versiones independientes de la misma aplicación, usando los mismos requisitos. Todas

las versiones son probadas con los mismos datos, para asegurar que todas proporcionan una salida idéntica.

Prueba de la tabla ortogonal: esta prueba puede aplicarse a problemas en que el dominio de entrada es relativamente pequeño pero demasiado grande para solicitar pruebas exhaustivas. El método de la tabla ortogonal es útil al encontrar errores asociados con fallos localizados.

Analizadas estas técnicas se concluye que son útiles siempre que se escoja bien el contexto en el cual se van a aplicar. De estas, la seleccionada fue Partición equivalente la cual permite examinar los valores válidos e inválidos de las entradas existentes en el software. Para la aplicación de esta técnica se realizan los diseños de casos de prueba los cuales se basan en una evaluación de las clases de equivalencia para una condición de entrada.

### **3.5.2 Prueba de Caja blanca**

Otro tipo de pruebas que se aplica con frecuencia al software son las de caja blanca, este es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba.

Mediante los métodos de prueba de caja blanca se garantiza que se recorra por lo menos una vez los caminos independientes de cada módulo, que se ejecuten todas las decisiones lógicas en sus opciones verdadera y falsa, que se ejerciten todos los bucles en sus límites y por último que se usen las estructuras internas de datos para asegurar su validez. (62)

Existen varias técnicas de pruebas de caja blanca, algunas de ellas son (62):

Camino básico: permite obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Para obtener el conjunto de caminos independientes se construye el grafo de flujo asociado y se calcula su complejidad ciclomática. Los casos de pruebas obtenidos garantizan que se ejecute al menos una vez cada sentencia del programa.

Prueba de condición: es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa. El propósito de esta técnica es detectar no solo errores en las condiciones, sino también otro tipo de errores.

Pruebas de flujos de datos: selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variantes del programa.

Prueba de bucles: se centra exclusivamente en la validez de las construcciones de bucles. Se pueden definir cuatro clases diferentes de bucles: bucles simples, bucles concatenados, bucles anidados y bucles no estructurados.

Inicialmente puede parecer que una prueba de caja blanca profunda nos puede llevar a tener programas correctos, definiendo todos los caminos lógicos, generar casos de prueba que examinen exhaustivamente la lógica del programa. Lamentablemente, estas pruebas incluso para programas pequeños el número de caminos lógicos que genera puede ser enorme. Las pruebas de caja blanca no se deben desechar como impracticables, se deben escoger una serie de caminos importantes a ejecutar.

La técnica optada fue Camino básico, dado que permite obtener una medida de la complejidad lógica del diseño y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución.

### **3.5.3 Resultados de las pruebas**

#### **Resultado de la prueba de Caja blanca**

La técnica utilizada para la aplicación de esta prueba fue la del Camino básico. A continuación se enumeran las sentencias de código del método adicionar (\$objeto, \$resultados) de la clase TActividadRepository, a modo de ejemplo, el mismo se encarga de adicionar una nueva actividad.

```
public function adicionar($objeto, $resultados) {
 try {
 $em = $this->getEntityManager();//1
 $actividad = new Tactividad();//1
 $dateFin = explode('/', $objeto->fechafin);//1
 $dateIni = explode('/', $objeto->fechaini);//1
 $dateFin = $dateFin[2] . '-' . $dateFin[1] . '-' . $dateFin[0];//1
 $dateIni = $dateIni[2] . '-' . $dateIni[1] . '-' . $dateIni[0];//1
 $horaIni = explode(' ', $objeto->horaini)[0];//1
 $horafin = explode(' ', $objeto->horafin)[0];//1
 $extra = (isset($objeto->extraPlan) && $objeto->extraPlan == 'on') ? 1 : 0;//1
 $actividad->setFechaini(new \DateTime($dateIni));//1
 ->setHoraFin(new \DateTime($horafin))//1
 ->setFechaFin(new \DateTime($dateFin))//1
 ->setDenominacion($objeto->denominacion)//1
 ->setHoraini(new \DateTime($horaIni))//1
 ->setPorcentajeCumplimiento($objeto->porcentajeCumplimiento)//1
 ->setExtraPlan($extra);//1
 if ($objeto->lugar != '')//2
 $actividad->setLugar($objeto->lugar);//3
 $em->persist($actividad);//4
 $em->flush();//4
 $em->getRepository('AppBundle:TPlanMensualProfesorActividad')->adicionar($actividad->getId(), $objeto->idplanmensualprofesor);//4
 if (count($resultados))//5
 $em->getRepository('AppBundle:TResultadoActividad')->adicionar($actividad->getId(), $resultados);//6
 return array('codMsg' => 1, 'msg' => 'La actividad ha sido adicionada satisfactoriamente.');//7
 }
}
```

Figura 22: Código fuente de la funcionalidad adicionar (\$objeto, \$resultado)

Para el cálculo de la complejidad ciclomática se representa el grafo del flujo asociado al código antes presentado a través de nodos, aristas y regiones.

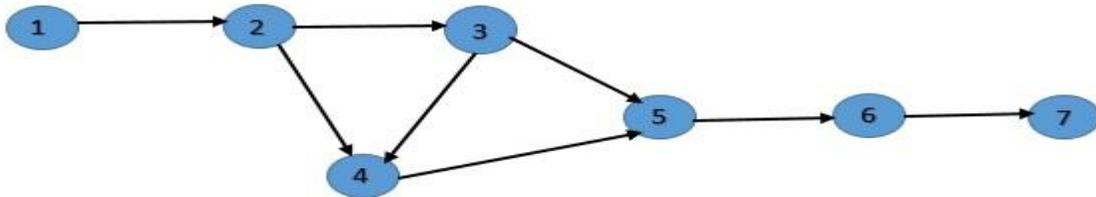


Figura 23: Grafo de flujo asociado a la funcionalidad adicionar (\$objeto, \$resultado)

La complejidad ciclomática es la métrica de software que proporciona una medición cuantitativa de la complejidad lógica de un programa. Esta métrica calcula la cantidad de caminos independientes de cada una de las funcionalidades del programa. También provee el límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez. (62)

Una vez construido el grafo de flujo asociado al algoritmo anterior se calcula la complejidad, el cálculo es necesario efectuarlo mediante tres fórmulas utilizando el mismo grafo en cada caso.

➤  $V(G) = (A - N) + 2$

Donde **A** es la cantidad total de aristas y **N** la cantidad total de nodos.

**Resultado:**  $V(G) = (8-7) + 2$

$$V(G) = 3$$

➤  $V(G) = P + 1$

Donde **P** es la cantidad de nodos predicados<sup>4</sup>

**Resultado:**  $V(G) = 2 + 1$

$$V(G) = 3$$

➤  $V(G) = R$

Donde **R** es la cantidad total de regiones, se incluye el área exterior del grafo, contando como una región más.

**Resultado:**  $V(G) = 3$

El cálculo efectuado mediante las fórmulas ha dado el mismo valor, por lo que se puede decir que la complejidad ciclomática es de 3, lo que significa que existen 3 posibles caminos por donde el flujo puede circular, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado. Seguidamente se especifican los caminos básicos que puede tomar el algoritmo durante su ejecución.

**Camino básico #1:** 1, 2, 3, 4, 5, 6, 7

**Camino básico #2:** 1, 2, 4, 5, 6, 7

**Camino básico #3:** 1, 2, 3, 5, 6, 7

Luego de obtenidos los caminos básicos, se realizan los casos de prueba para cada uno de ellos.

---

<sup>4</sup> Nodos predicados: nodos de los cuales partes dos o más aristas.

**Tabla 7:** Caso de prueba para el camino básico #1

| <b>Camino básico #1: 1, 2, 3, 4, 5, 6, 7</b> |                                                                                                          |
|----------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Descripción                                  | A partir de la entrada de los datos, se verifican si estos son válidos para posteriormente adicionarlos. |
| Condición de ejecución                       | if (\$objeto->lugar !== "<br>if (count(\$resultados))                                                    |
| Entrada                                      | \$objeto, \$resultado                                                                                    |
| Resultado esperado                           | Mostrar un mensaje notificando que la actividad ha sido añadida satisfactoriamente.                      |

**Tabla 8:** Caso de prueba para el camino básico #2

| <b>Camino básico #2: 1, 2, 3, 5, 6, 7</b> |                                                                                                          |
|-------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Descripción                               | A partir de la entrada de los datos, se verifican si estos son válidos para posteriormente adicionarlos. |
| Condición de ejecución                    | if (\$objeto->lugar !== "                                                                                |
| Entrada                                   | \$objeto, \$resultado                                                                                    |
| Resultado esperado                        | Mostrar un mensaje notificando que la actividad ha sido añadida satisfactoriamente.                      |

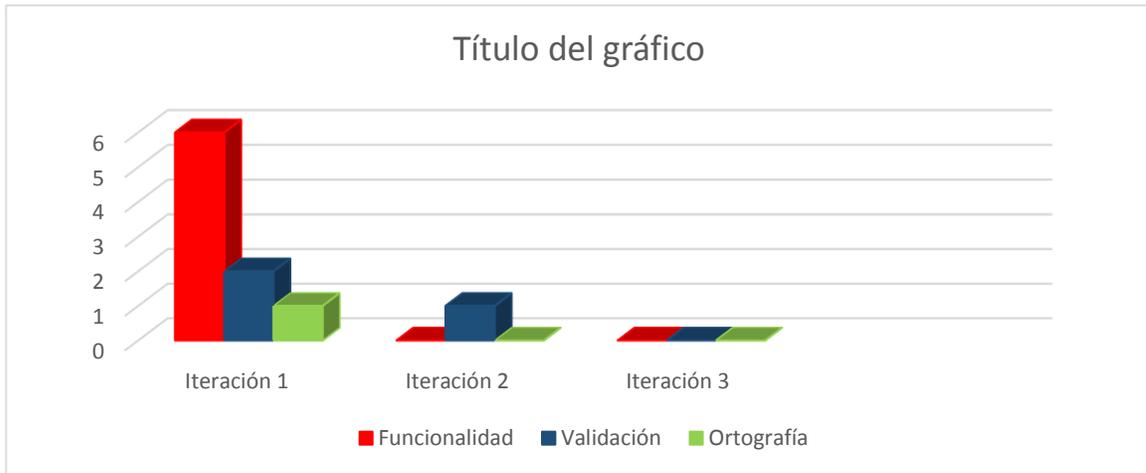
**Tabla 9:** Caso de prueba para el camino básico #3

| <b>Camino básico #1: 1, 2, 4, 5, 6, 7</b> |                                                                                                          |
|-------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Descripción                               | A partir de la entrada de los datos, se verifican si estos son válidos para posteriormente adicionarlos. |
| Condición de ejecución                    | if (count(\$resultados))                                                                                 |
| Entrada                                   | \$objeto, \$resultado                                                                                    |
| Resultado esperado                        | Mostrar un mensaje notificando que la actividad ha sido añadida satisfactoriamente.                      |

Se puede concluir que al aplicar la prueba el resultado arrojado fue que para que se ejecute cada sentencia al menos una vez, el límite superior de casos de pruebas debe ser 3.

### Resultados de las prueba de Caja negra

Para realizar las pruebas funcionales se utiliza la técnica de caja negra, haciendo uso del método de partición equivalente. Para aplicarla se confeccionaron los diseños de casos de prueba para cada una de las funcionalidades del sistema. Las pruebas funcionales al sistema se realizaron en el departamento de Calidad del Centro CEIGE, revisado por el ingeniero Yoan Antonio López y la ingeniera Yisel Niño Benitez en 3 iteraciones, a continuación, en la Figura 24 se refleja la cantidad de no conformidades detectadas en cada una de estas iteraciones:



**Figura 24:** Resultados de la prueba de Caja negra

En la iteración 1 se detectaron un total de 9 no conformidades, 6 de funcionalidad, 2 de validación y 1 de ortografía. Se realizó una segunda iteración para rectificar los errores detectados en la primera iteración de pruebas. Fue detectada 1 no conformidad, la misma se correspondía con la de la iteración anterior, ya que el sistema no validaba que la entrada de los datos en el campo denominación de las disciplinas, colectivos y planes de trabajo fueran números y letras. En la tercera iteración de prueba no se detectaron no conformidades, por lo que se puede confirmar que el software tiene una calidad aceptable y se emite el acta de liberación.

### **3.6 Validación de la investigación**

#### Pruebas de Aceptación

Otra de las pruebas realizadas fueron las de aceptación con el cliente, que en este caso es la jefa del departamento de Ingeniería de Software de la facultad 3, donde se evidencia la aprobación del correcto funcionamiento del sistema, debido a que satisface los requisitos funcionales que se identificaron, ver Anexo 25.

### **3.7 Conclusiones del capítulo**

En el presente capítulo se enunciaron los aspectos fundamentales de la disciplina de implementación donde se definieron los estándares de codificación que hicieron más entendible la implementación a la hora de plasmar las líneas de código dentro de la aplicación, logrando así un mejor entendimiento tanto de la implementación como de futuras versiones que se puedan hacer al software. Se confeccionó un diagrama de despliegue y se explica el mismo para un mejor entendimiento de la manera en que se relacionan los diferentes dispositivos que pueden interactuar en el flujo de la aplicación. Se realizaron pruebas de software al sistema, donde se aplicaron las pruebas unitarias con el método de Caja blanca y pruebas funcionales con el método de Caja negra, utilizando las técnicas: camino básico, permitiendo conocer el mínimo de casos de pruebas a realizar; partición equivalente, que permite examinar los valores válidos e inválidos de las entradas existentes en el software; arrojando resultados satisfactorios en cada uno de los métodos aplicados.

## **Conclusiones Generales**

Luego de realizada la investigación para el desarrollo de un sistema informático para la evaluación de los profesores del departamento de Ingeniería de Software de la facultad 3, se arriba a las siguientes conclusiones:

- Se logra un mejor entendimiento del objeto de estudio, de los sistemas de gestión y de las herramientas y tecnologías utilizadas en la solución del problema, apoyándose en los conceptos y aspectos teóricos abordados en la primera etapa de la investigación.
- Se obtuvo un sistema para la evaluación de los profesores del departamento de Ingeniería de Software de la facultad 3 acorde a las necesidades del proceso de planificación, utilizando los artefactos generados por la metodología AUP-UCI, que facilitaron un mejor entendimiento de la situación a resolver.
- Se realiza la validación y verificación de las funcionalidades del sistema mediante las pruebas de Caja blanca, Caja negra y de Aceptación.

## **Recomendaciones**

Se recomienda para futuras iteraciones y versiones del sistema propuesto:

- Conectar el software con el servidor de dominio LDAP (Protocolo de Acceso Ligerero a Directorios) de la Universidad de las Ciencias Informáticas (UCI), para lograr una autenticación de manera más segura ya que LDAP soporta la Capa de conexión segura (SSL) y la Seguridad de la capa de transporte (TLS), donde los datos confidenciales se pueden proteger.

## Bibliografía

1. [En línea] 9 de febrero de 2015. <http://www.cadenagramonte.cu/articulos/ver/36935:informatizacion-de-la-sociedad-cubana-avanza-a-pesar-del-bloqueo-yanqui..>
2. C. General, . *Resolución No.60/11*. La Habana : s.n., 2015.
3. Definición de. [En línea] [Citado el: 15 de marzo de 2015.] <http://definicion.de/software/>.
4. Definición de. [En línea] [Citado el: 16 de marzo de 2015.] <http://definicion.de/planificacion>.
5. [En línea] 8 de febrero de 2015. <http://definicion.de/plan-de-trabajo>.
6. Pressman. *El producto y el proceso, 5ta edición* . 2015.
7. Cataldi, Zulma. [En línea] 13 de marzo de 2015. <http://sedici.unlp.edu.ar/handle/10915/4055>.
8. Pressman. *Modelo de procesos 6ta edición*. La Habana : s.n., 2015.
9. Rivera, Calvo, Uvaldo. [En línea] <http://es.slideshare.net/linarodriguez16/historia-del-software-contable>.
10. *Noticias Darkub*. [En línea] 8 de marzo de 2015. <https://darkub.wordpress.com/2008/12/20/%C2%BFque-tipos-de-software-hay-y-como-se-clasifican/>.
11. [En línea] 9 de febrero de 2015. <http://es.slideshare.net/Pablorton/software-de-gestion>.
12. [En línea] 10 de marzo de 2015. <http://www.inforolot.com/index.php/es/sistema-de-gestion-documental.html>.
13. *ISOtools*. [En línea] 11 de marzo de 2015. <http://www.isotools.org/2014/10/22/iso-9001-fundamentos-sistemas-gestion-calidad/>.
14. Centro Nacional de Información de la Calidad. Asocioación española para la calidad. [En línea] 6 de mayo de 2015. [http://www.aec.es/c/document\\_library/get\\_file?uuid=0fed9322-3dea-4211-b748-a1e041a60b01&groupId=10128](http://www.aec.es/c/document_library/get_file?uuid=0fed9322-3dea-4211-b748-a1e041a60b01&groupId=10128).

- 
15. *redgrafica*. [En línea] 11 de marzo de 2015. [http://redgrafica.com/IMG/pdf/Fundamentos\\_Sistema\\_Calidad.pdf](http://redgrafica.com/IMG/pdf/Fundamentos_Sistema_Calidad.pdf).
  16. [En línea] 15 de marzo de 2015. <http://gm3s.com.mx/blog/sistemas-de-gestion-empresarial/>.
  17. Informática Hoy. [En línea] 15 de marzo de 2015. <http://www.informatica-hoy.com.ar/software-erp/Caracteristicas-fundamentales-del-ERP.php>.
  18. Universidad de león. [En línea] Universidad de león, 2011. [Citado el: 6 de mayo de 2015.] <https://www.unileon.es>.
  19. Ballester, Pérez, Liu y Fitó, Vázquez, Jiorqui. *Componente Gestor de Intercambio de Información JIT para el Sistema para la Planificación de Actividades (SIPAC)*. La Habana : s.n., 2011.
  20. Pérez Ballester, Liu y Vázquez Fitó, Jiorqui. *Componente Gestor de Intercambio de Información JIT para el Sistema para la Planificación de Actividades (SIPAC)*. Tesis. La Habana : s.n., 2012.
  21. Instituto Nacional de Tecnologías de la Comunicación (INTECO). *INGENIERÍA DEL SOFTWARE: METODOLOGÍAS Y CICLOS DE VIDA*. España : s.n., 2009.
  22. [En línea] [Citado el: 22 de marzo de 2015.] [http://aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=688:ique-es-y-para-que-sirve-uml-versiones-de-uml-lenguaje-unificado-de-modelado-tipos-de-diagramas-uml&catid=46:lenguajes-y-entornos&Itemid=163](http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=688:ique-es-y-para-que-sirve-uml-versiones-de-uml-lenguaje-unificado-de-modelado-tipos-de-diagramas-uml&catid=46:lenguajes-y-entornos&Itemid=163).
  23. [En línea] [Citado el: 22 de marzo de 2015.] <http://users.dcc.uchile.cl/~psalinas/uml/introduccion.html>.
  24. *academia.edu*. [En línea] [Citado el: 1 de mayo de 2015.] [http://www.academia.edu/5012832/Herramienta\\_CASE](http://www.academia.edu/5012832/Herramienta_CASE).
  25. [En línea] [Citado el: 22 de marzo de 2015.] <http://www.ie.inf.uc3m.es/grupo/docencia/reglada/ls1y2/PracticaVP.pdf>.
  26. Febles Diaz, Orestes. Slideshare. [En línea] [Citado el: 22 de Marzo de 2015.] <http://es.slideshare.net/ofebles/visual-paradign>.

27. Kioskea. Kioskea. [En línea] [Citado el: 28 de 04 de 2015.] <http://es.kioskea.net/contents/304-lenguajes-de-programacion>.
28. Lenguajes de programación del lado del servidor. Lenguajes de programación del lado del servidor. [En línea] [Citado el: 28 de 04 de 2015.] <https://docs.google.com/presentation/d/1ZQglsW4KCylfH8NyiQH9ueanM83Ym7bjitaxyQKS0hs/edit?pli=1#slide=id.i7>.
29. PHP. PHP. [En línea] [Citado el: 19 de 01 de 2015.] <http://php.net/manual/es/intro-what-is.php>.
30. 4R. 4R. [En línea] 12 de 07 de 2012. [Citado el: 28 de 04 de 2015.] <http://www.4rsoluciones.com/las-ventajas-de-php-para-el-desarrollo-de-aplicaciones-y-sitios-web/>.
31. Villalobos, Jorge. Código Programación. [En línea] 05 de 09 de 2010. [Citado el: 28 de 04 de 2015.] <http://codigoprogramacion.com/articulos/programacionweb/ventajas-de-usar-php.html>.
32. Marley , Jimi. Programación en castellano. [En línea] [Citado el: 28 de 04 de 2015.] [http://programacion.net/articulo/por\\_que\\_elegir\\_php\\_143](http://programacion.net/articulo/por_que_elegir_php_143).
33. Brezo, M<sup>a</sup> Paz. Lance Talent. [En línea] [Citado el: 28 de 04 de 2015.] <http://www.lancetalent.com/blog/6-buenos-motivos-para-trabajar-con-php/>.
34. Torre, Aníbal de la. Adelat. [En línea] 2006. [Citado el: 20 de 01 de 2015.] [http://www.adelat.org/media/docum/nuke\\_publico/lenguajes\\_del\\_lado\\_servidor\\_o\\_cliente.html](http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html).
35. Sencha. [En línea] [Citado el: 12 de abril de 2015.] [www.sencha.com](http://www.sencha.com).
36. Gracia, J. Web estilo. [En línea] [Citado el: 15 de 12 de 2014.] <http://www.webestilo.com/mysql/>.
37. PaCko. PaCko. [En línea] [Citado el: 28 de 04 de 2015.] <https://packo.wikispaces.com/Caracteristicas+de+MYSQL>.
38. Londoño O., Andres Felipe y Oviedo P., Cristian Andrey. Slidershare. [En línea] 28 de 05 de 2009. [Citado el: 28 de 04 de 2015.] <http://es.slideshare.net/anfelon/mysql-1503822>.
39. MySql. MySql. [En línea] 07 de 09 de 2008. [Citado el: 28 de 04 de 2015.] <http://sistemaspyt.blogspot.com/2008/09/principales-caracteristicas.html>.

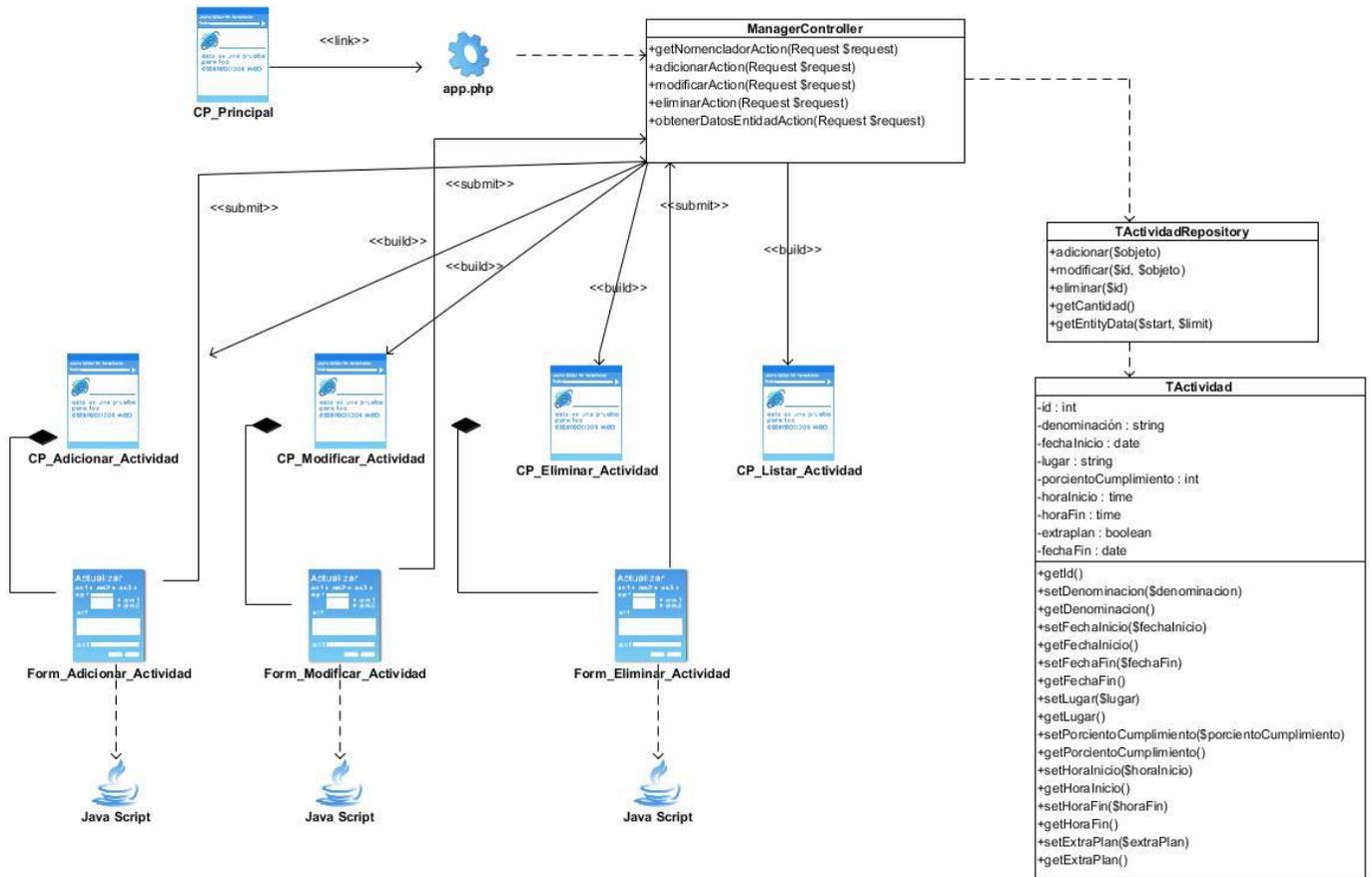
- 
40. JetBRAINS. [En línea] [Citado el: 22 de Marzo de 2015.] <https://www.jetbrains.com/phpstorm/>.
41. Maestros del Web. [En línea] [Citado el: 22 de marzo de 2015.] <http://www.maestrosdelweb.com/curso-symfony2-introduccion-instalacion/>.
42. Sierra , Manuel. Aprende a programar. [En línea] [Citado el: 30 de 04 de 2015.] [http://aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftppop3-y-smtp-dhcp&catid=57:herramientas-informaticas&Itemid=179](http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftppop3-y-smtp-dhcp&catid=57:herramientas-informaticas&Itemid=179).
43. Mas adelante. Mas adelante. [En línea] [Citado el: 30 de 04 de 2015.] <http://www.masadelante.com/faqs/servidor>.
44. Ciberaula. [En línea] [Citado el: 23 de marzo de 2015.] [http://linux.ciberaula.com/articulo/linux\\_apache\\_intro](http://linux.ciberaula.com/articulo/linux_apache_intro).
45. [En línea] [Citado el: 2015 de mayo de 19.] [http://www-2.dc.uba.ar/materias/isoft1/teoricas\\_2009\\_1/05a-ModeloConceptual\\_BN.pdf](http://www-2.dc.uba.ar/materias/isoft1/teoricas_2009_1/05a-ModeloConceptual_BN.pdf).
46. Universidad Distrital Francisco José de Caldas. *Proceso de desarrollo OPEN UP/OAS*.
47. Sommerville, Ian. *Ingeniería del Software 7ma edición*. Madrid : s.n., 2005.
48. López, Rodríguez y Antonio, Yoan. Definición de los requerimientos funcionales del módulo Tesorería, Préstamos y Depósitos del proyecto Banco Nacional. *Tesis*. 2008.
49. Sommerville, Ian y Sawyer, Pete. *Requirements Engineering: A Good Practice Guide*. 1997.
50. Orjuela, Duarte, Ailin y Rojas, Mauricio. *Las metodologías de desarrollo ágil como una oportunidad para la Ingeniería de Software Educativo*. Colombia : s.n., 2008.
51. Pablo, Quiroga, Juan. *Requerimientos funcionales y no funcionales*. Andes : Dpto. de Ingeniería de Sistemas y Computación Universidad de los Andes .
52. ALTOVA. *Diagramas de secuencia UML*. [En línea] [Citado el: 8 de abril de 2015.] <http://www.altova.com/es/umodel/sequence-diagrams.html>.

53. Que es un modelo de datos. [En línea] [Citado el: 19 de mayo de 2015.] [www.odette.es/SGC/downloads/cursoEDI/406.pps](http://www.odette.es/SGC/downloads/cursoEDI/406.pps).
54. Pavón Mestras, Juan. *Estructura de las Aplicaciones Orientadas a Objetos. El patrón Modelo-Vista-Controlador (MVC)*. Madrid : s.n., 2008-2009.
55. Mejía, Fracisco. ISSUU-Patrones de diseño web. [En línea] [http://issuu.com/franciscomejia99/docs/patrones\\_de\\_dise\\_o\\_web\\_](http://issuu.com/franciscomejia99/docs/patrones_de_dise_o_web_).
56. Larman, Craig. *UML y Patrones*.
57. Microsoft. [En línea] [Citado el: 31 de mayo de 2015.] <https://msdn.microsoft.com/es-es/library/dd409390.aspx>.
58. slideshare. [En línea] [Citado el: 1 de mayo de 2015.] <http://es.slideshare.net/edybest/metricas-de-proceso-y-proyecto-8166786>.
59. CODEJOBS. [En línea] [Citado el: 30 de Mayo de 2015.] <http://www.codejobs.biz/es/blog/2014/10/10/de-donde-proviene-el-camelcase#sthash.J9BUdvTj.dpbs>.
60. Jonathan, Nieto y Marco, Martínez. [En línea] [Citado el: 30 de Marzo de 2015.] <http://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CBwQFjAA&url=http%3A%2F%2Fayni.googlecode.com%2Fsvn%2Fdocs%2FVSE%2FVSE-EquipoD%2F%2528grupal%2529%2520Estandares%2520de%2520Programacion.docx.doc&ei=VSVuVZqMApKvyQTWwYGw>.
61. Marca, Huallpara, Hugo Michael y Quisbert Limanchi, Nancy Susana. *Diagrama de despliegue*.
62. Pressman, Roger S. *Ingeniería de Software. Un enfoque práctico. Quinta edición*. 2005.
63. PostgreSQL-es. [En línea] [Citado el: 23 de marzo de 2015.] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).
64. PostgreSQL-es. [En línea] [Citado el: 23 de marzo de 2015.] <http://www.postgresql.org.es/node/655>.

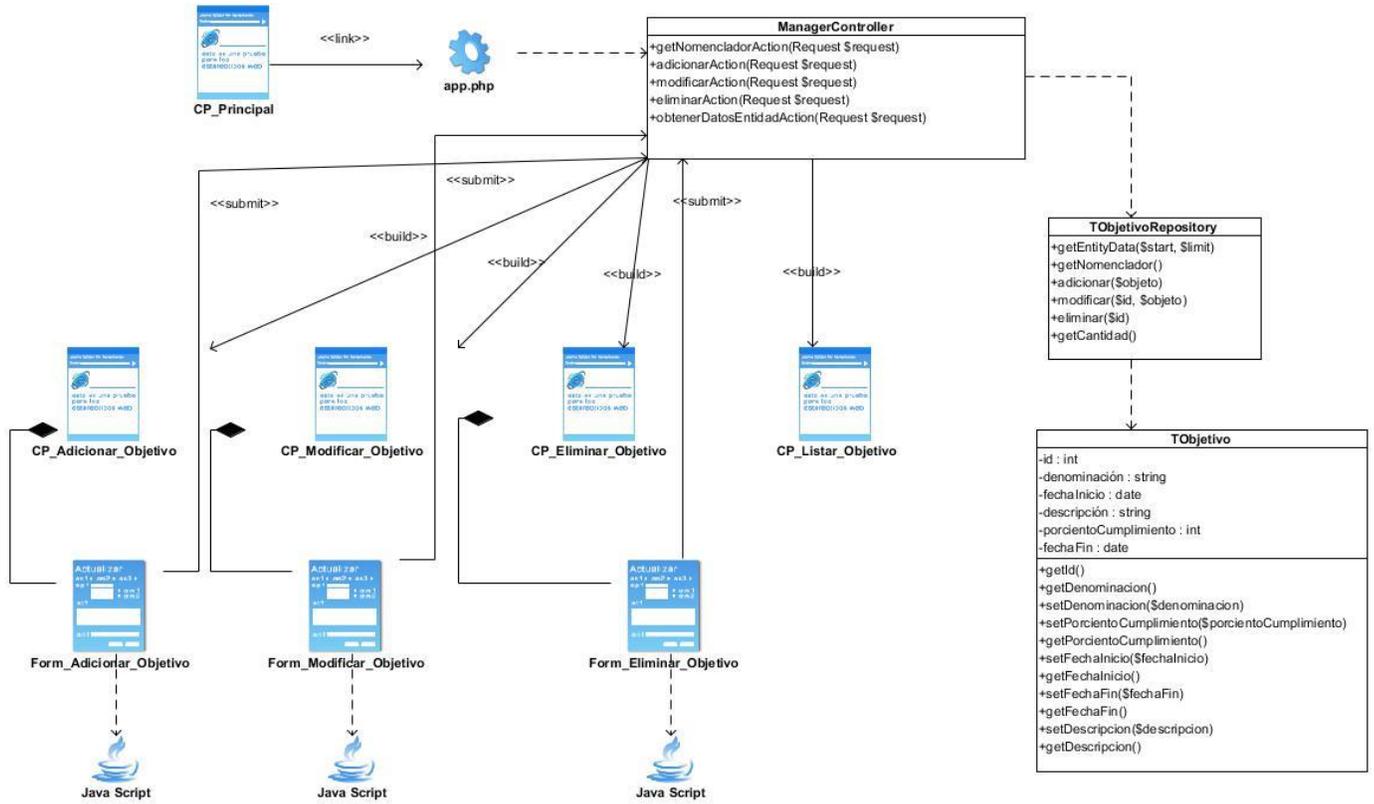
65. [En línea] [Citado el: 25 de Mayo de 2015.] La arquitectura Cliente/Servidor fue la utilizada para el desarrollo de la aplicación. Según IBM, es la arquitectura que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo.
66. kioskea. [En línea] [Citado el: 25 de Mayo de 2015.] <http://es.kioskea.net/contents/148-entorno-cliente-servidor>.

## Anexos

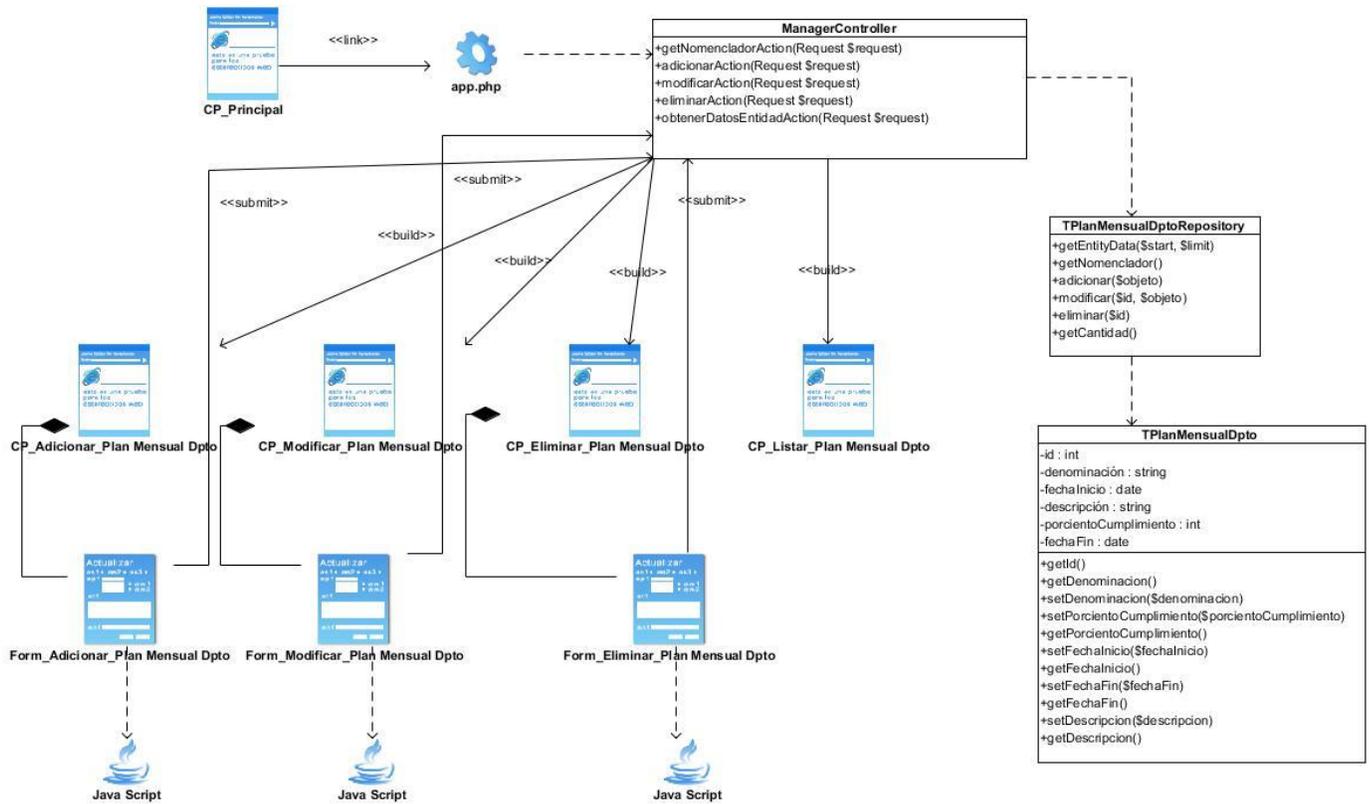
### Anexo 1. Diagrama de clases de la agrupación Gestionar actividad



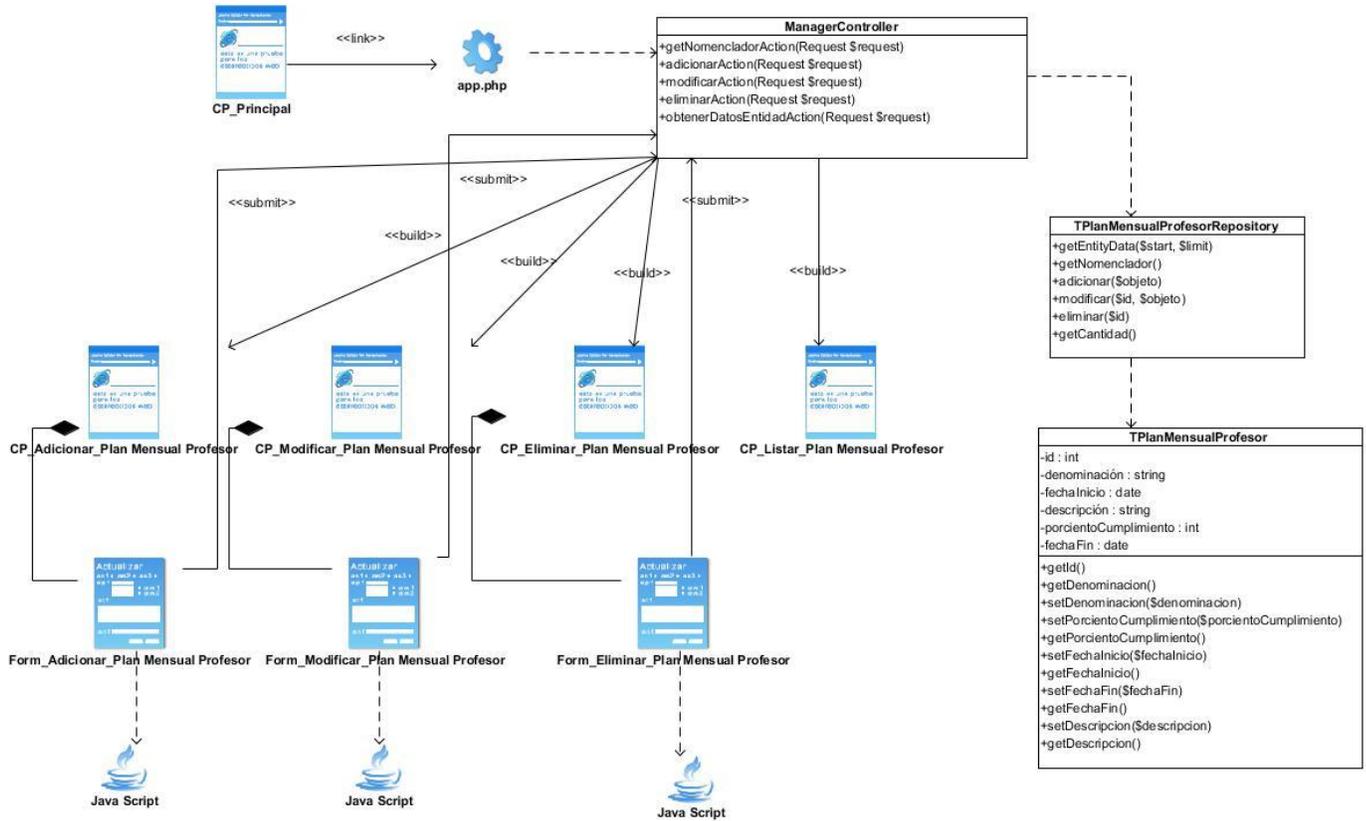
## Anexo 2. Diagrama de clases de la agrupación Gestionar objetivos



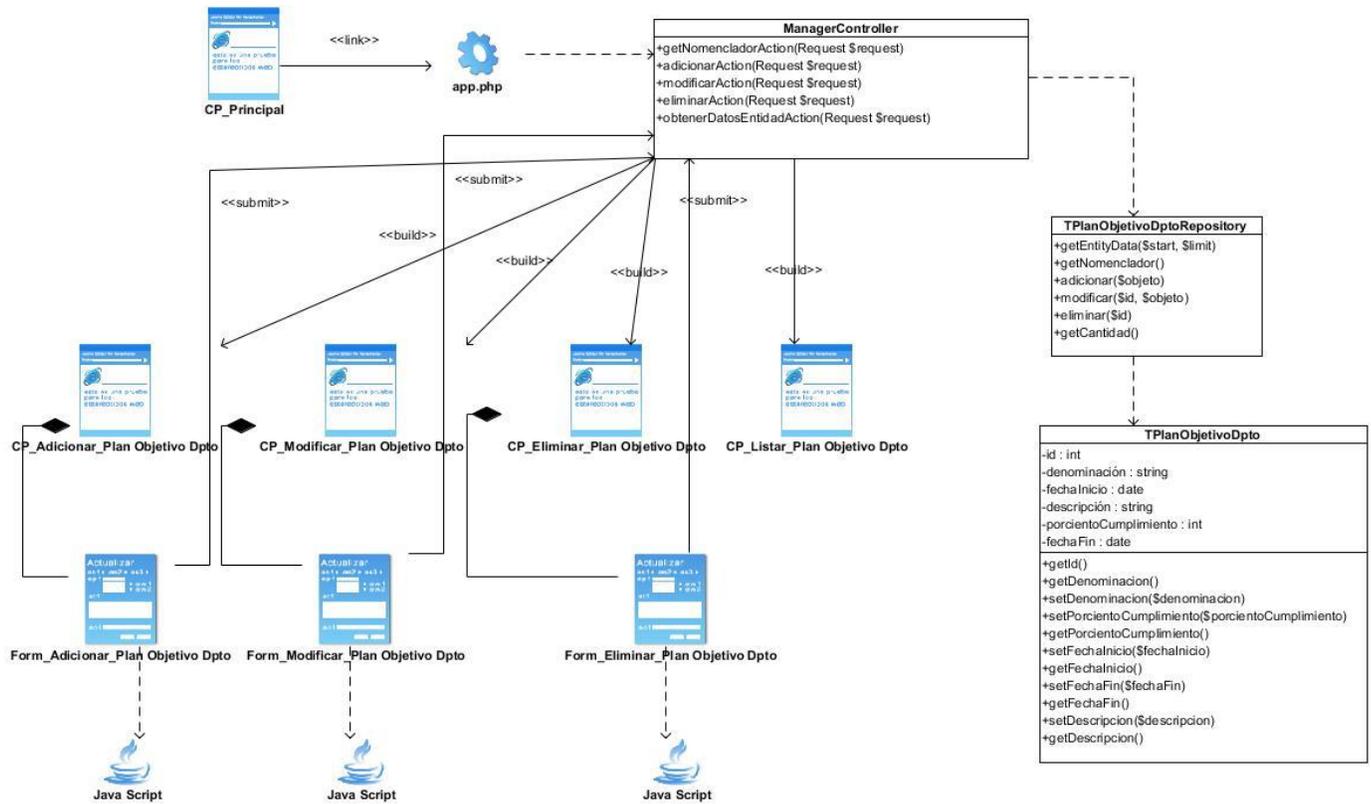
### Anexo 3. Diagrama de clases de la agrupación Gestionar plan mensual del departamento



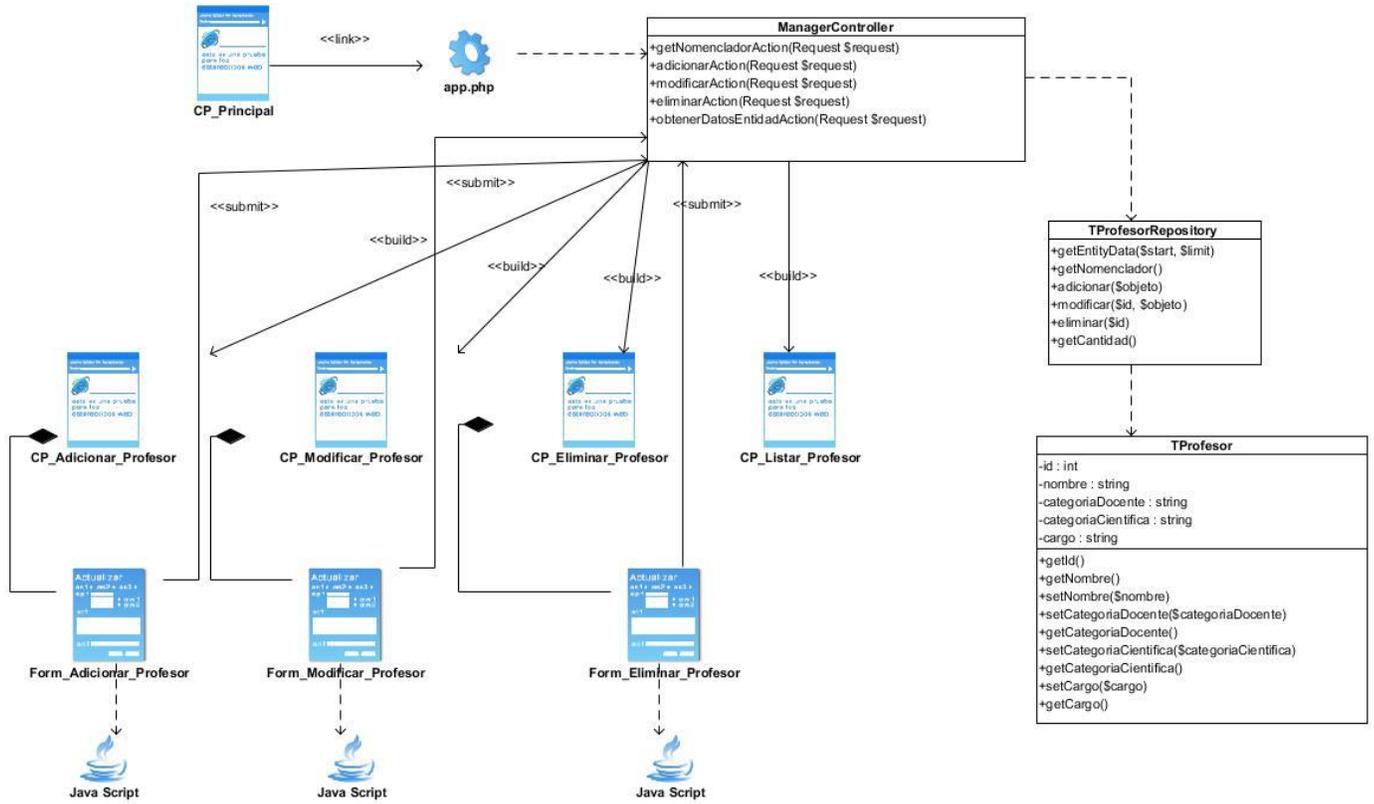
### Anexo 4. Diagrama de clases de la agrupación Gestionar plan mensual del profesor



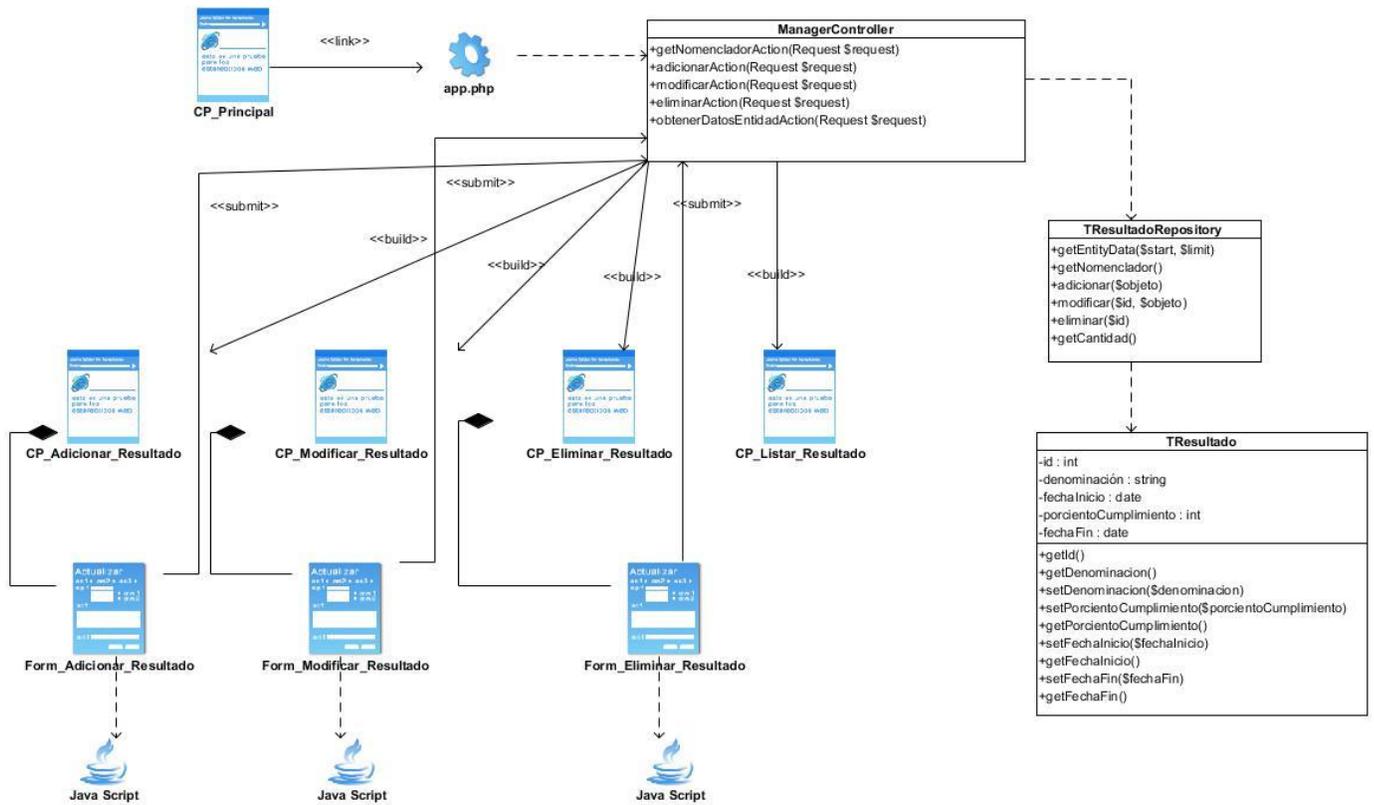
### Anexo 5. Diagrama de clases de la agrupación Gestionar plan de objetivos



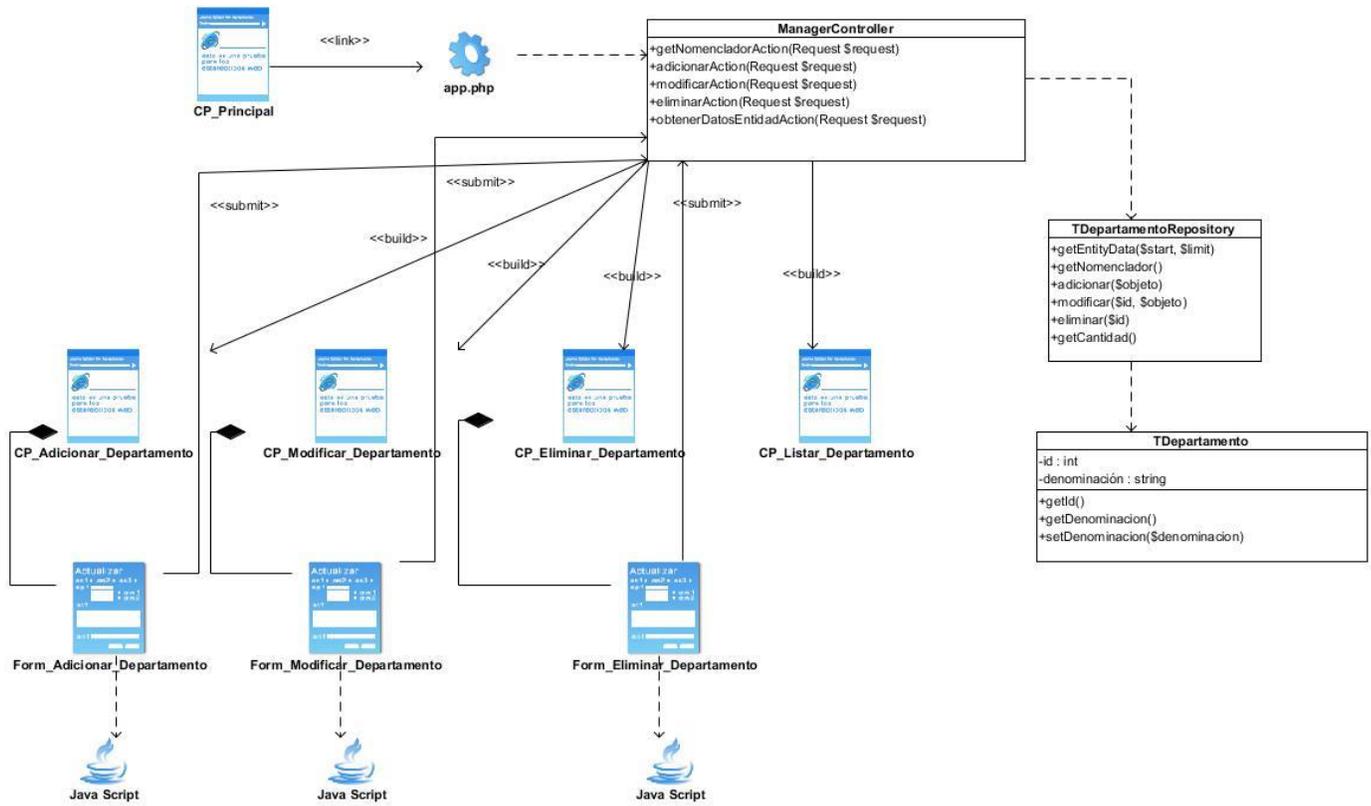
## Anexo 6. Diagrama de clases de la agrupación Gestionar profesor



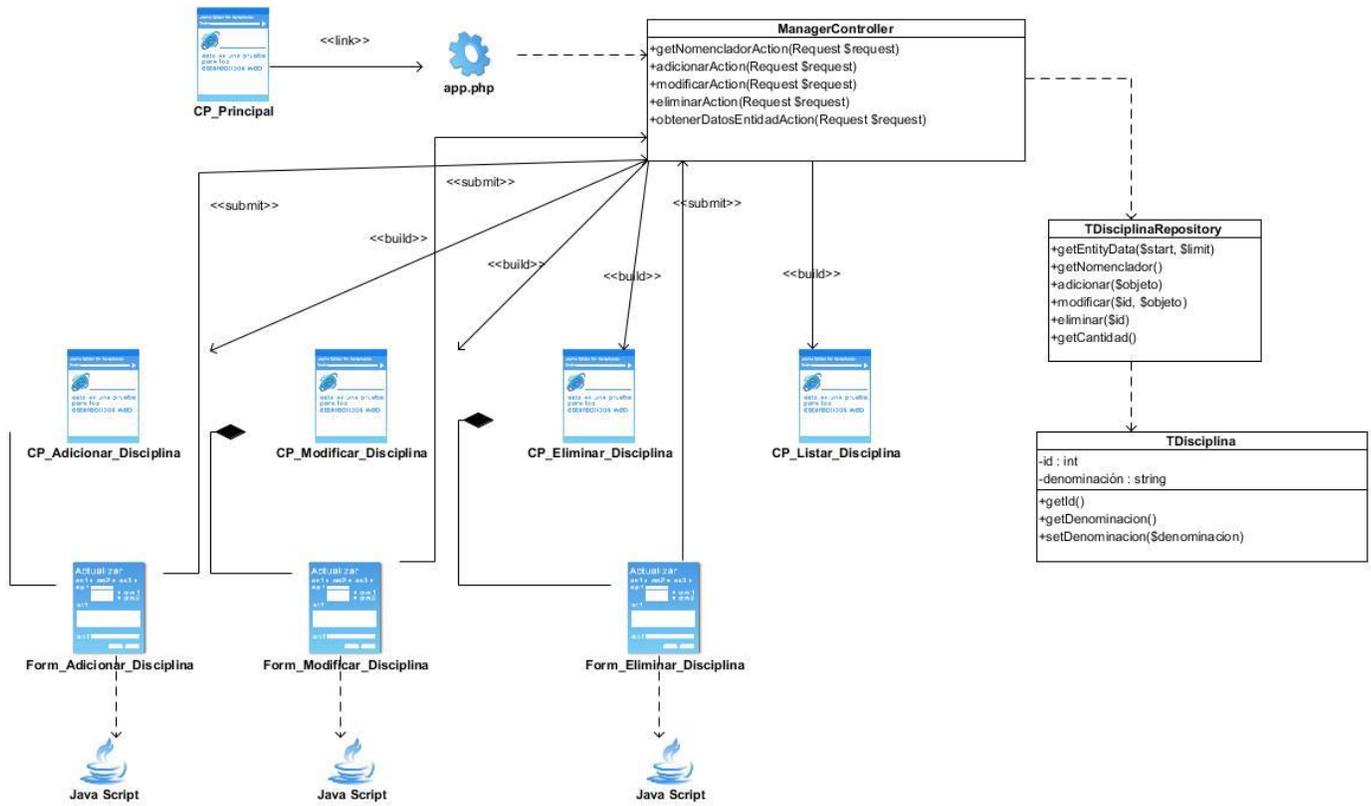
## Anexo 7. Diagrama de clases de la agrupación Gestionar resultados



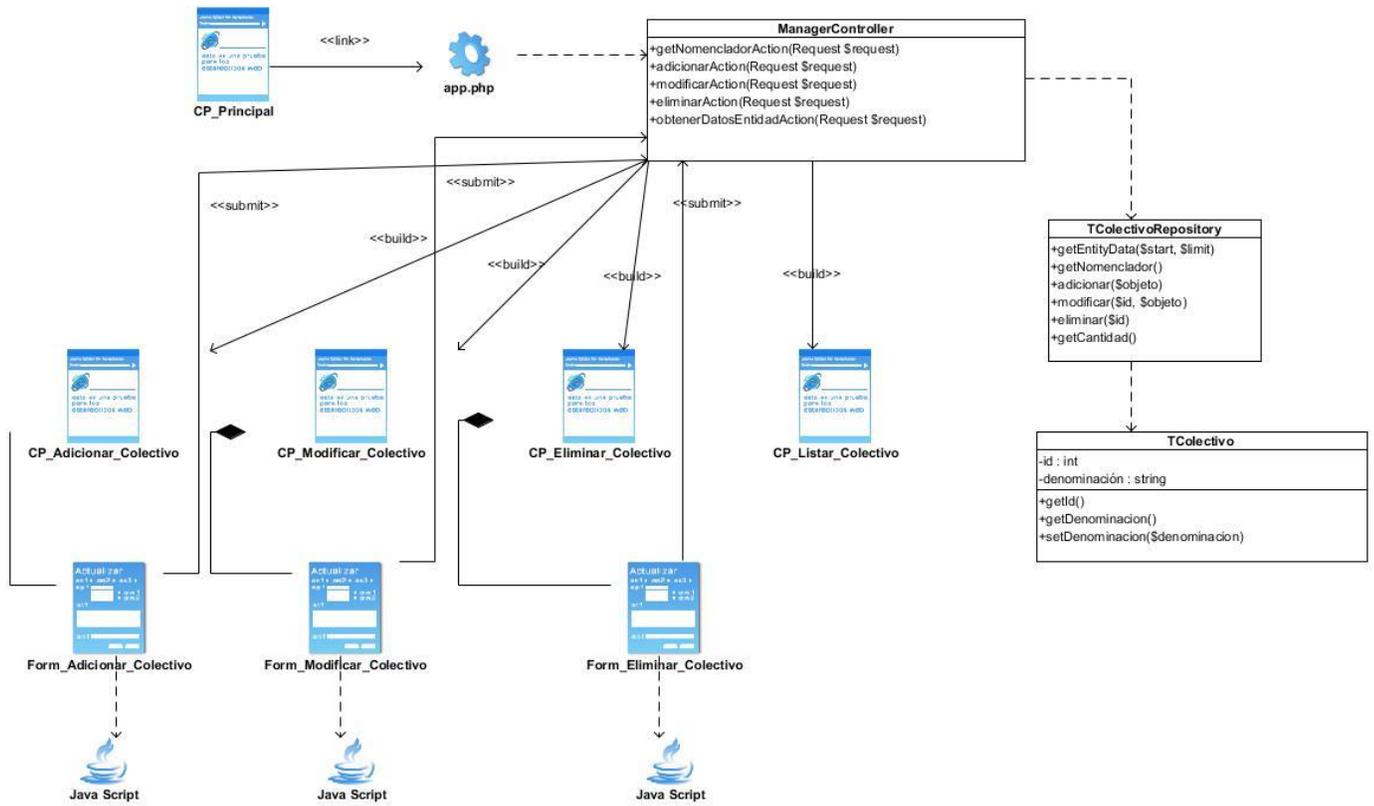
### Anexo 8. Diagrama de clases de la agrupación Gestionar departamento



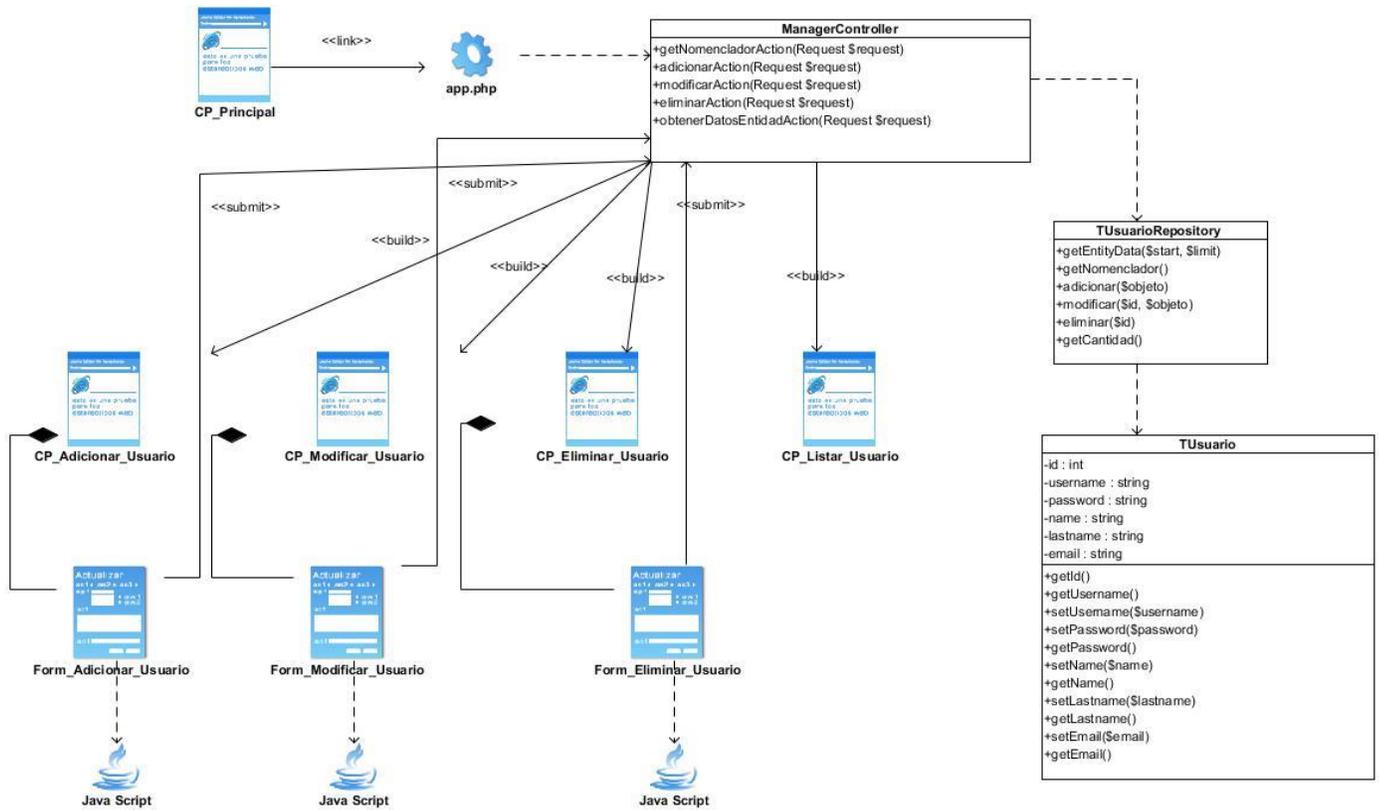
### Anexo 9. Diagrama de clases de la agrupación Gestionar disciplina



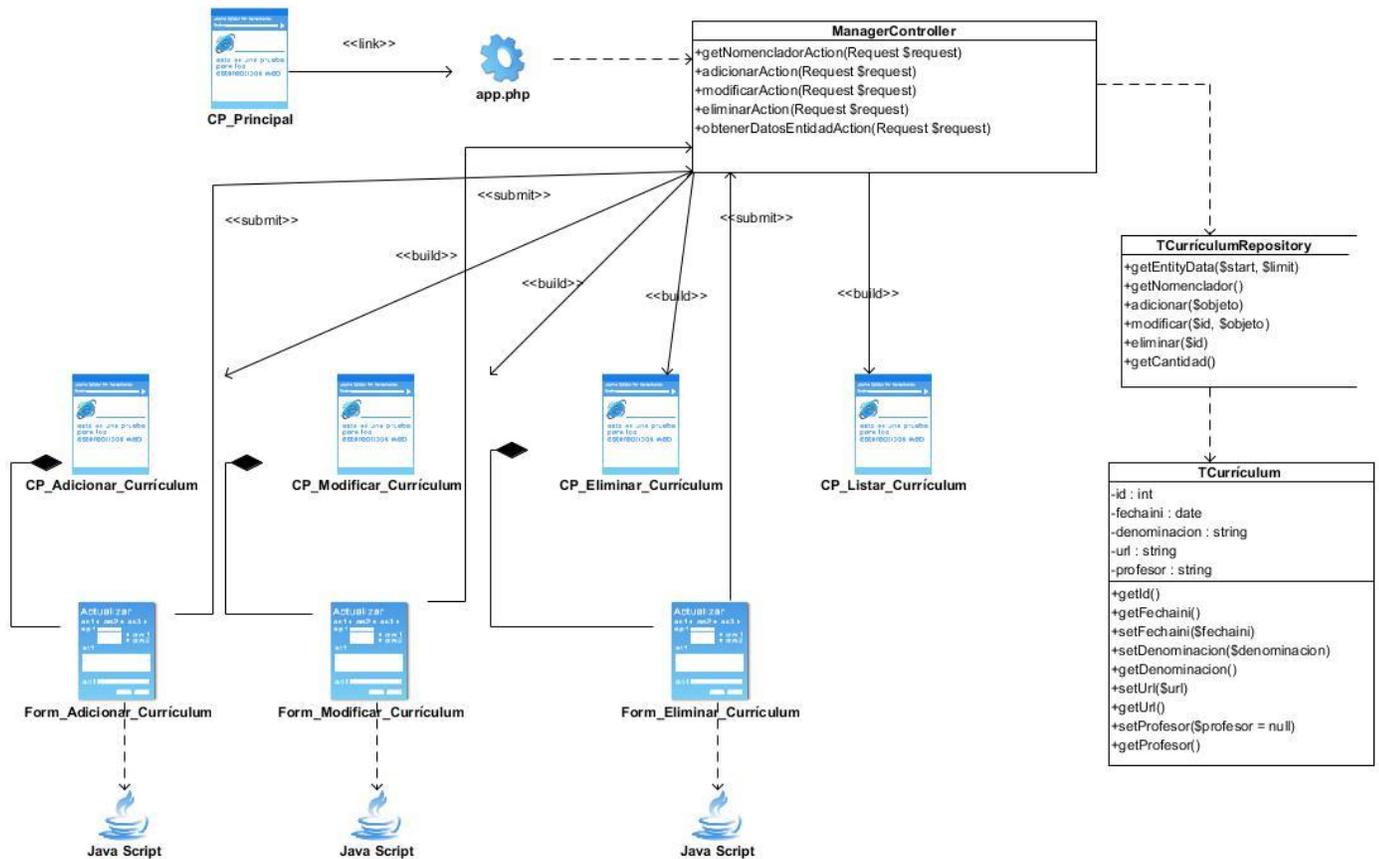
### Anexo 10. Diagrama de clases de la agrupación Gestionar colectivo



## Anexo 11. Diagrama de clases de la agrupación Gestionar Usuario

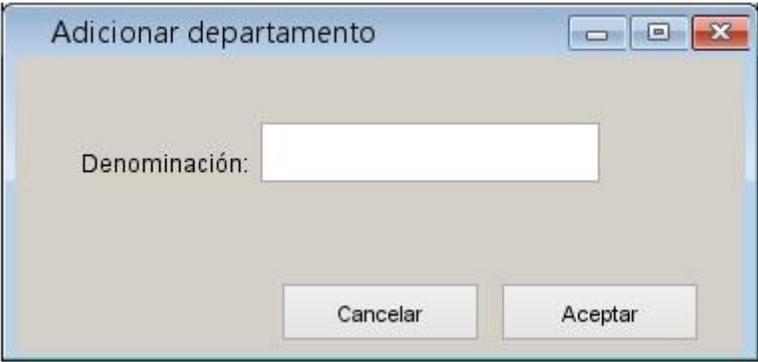


**Anexo 12. Diagrama de clases de la agrupación Gestionar Usuario**



**Anexo 13. Historias de usuario perteneciente a la agrupación por requisitos Gestionar departamento**

| Historia de usuario: Adicionar departamento |                                              |
|---------------------------------------------|----------------------------------------------|
| Número: 1                                   | Nombre del requisito: Adicionar departamento |
| Programador: Yaima                          | Iteración Asignada: 1                        |
| Prioridad: Alta                             | Tiempo Estimado: 0.5                         |

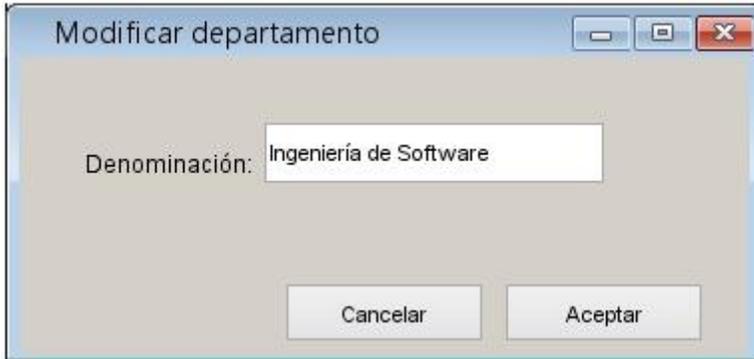
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                          |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| <b>Riesgo en Desarrollo:</b> Bajo                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <b>Tiempo Real:</b> 0.14 |
| <p><b>Descripción:</b> Esta historia de usuario permite adicionar un nuevo departamento docente. En caso de que los campos se llenen correctamente introduciendo caracteres válidos se adiciona un nuevo departamento docente, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos y en caso de que haya campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos.</p> |                          |
| <p><b>Observaciones:</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                          |
| <p><b>Prototipo de interfaz:</b></p>                                                                                                                                                                                                                                                                                                                                                                                |                          |

| Historia de usuario: Modificar departamento                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| <b>Número:</b> 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | <b>Nombre del requisito:</b> Modificar departamento |
| <b>Programador:</b> Yaima                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | <b>Iteración Asignada:</b> 1                        |
| <b>Prioridad:</b> Alta                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | <b>Tiempo Estimado:</b> 0.5                         |
| <b>Riesgo en Desarrollo:</b> Bajo                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <b>Tiempo Real:</b> 0.14                            |
| <p><b>Descripción:</b> Esta historia de usuario permite modificar un departamento docente. En caso de que se realicen cambios y los campos se llenen correctamente introduciendo caracteres válidos se modifica el departamento docente, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos, en caso de que se deje campos vacíos se muestra un mensaje indicándole al usuario que hay</p> |                                                     |

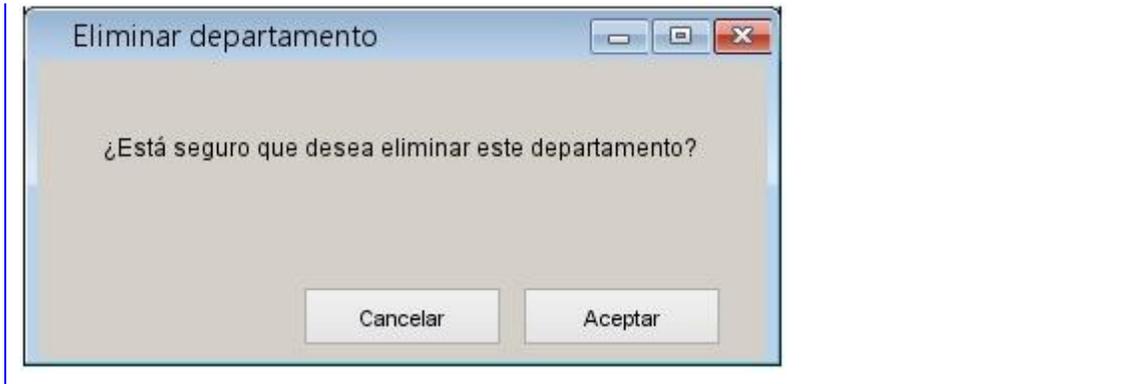
campos en blancos y en caso de que no se realice ningún cambio se muestra al usuario un mensaje indicándole que no realizaron cambios.

**Observaciones:**

**Prototipo de interfaz:**



| Historia de usuario: Eliminar departamento                                                                                                                                                                                       |                                                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| <b>Número:</b> 3                                                                                                                                                                                                                 | <b>Nombre del requisito:</b> Eliminar departamento |
| <b>Programador:</b> Yaima                                                                                                                                                                                                        | <b>Iteración Asignada:</b> 1                       |
| <b>Prioridad:</b> Alta                                                                                                                                                                                                           | <b>Tiempo Estimado:</b> 0.5                        |
| <b>Riesgo en Desarrollo:</b> Bajo                                                                                                                                                                                                | <b>Tiempo Real:</b> 0.14                           |
| <b>Descripción:</b> Esta historia de usuario permite eliminar un departamento docente. Se selecciona el departamento que se desea eliminar y se le muestra al usuario un mensaje de si está seguro de eliminar ese departamento. |                                                    |
| <b>Observaciones:</b>                                                                                                                                                                                                            |                                                    |
| <b>Prototipo de interfaz:</b>                                                                                                                                                                                                    |                                                    |



| Historia de usuario: Listar departamento                                                                 |                                                  |
|----------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| <b>Número:</b> 4                                                                                         | <b>Nombre del requisito:</b> Listar departamento |
| <b>Programador:</b> Yaima                                                                                | <b>Iteración Asignada:</b> 1                     |
| <b>Prioridad:</b> Alta                                                                                   | <b>Tiempo Estimado:</b> 0.5                      |
| <b>Riesgo en Desarrollo:</b> Bajo                                                                        | <b>Tiempo Real:</b> 0.14                         |
| <b>Descripción:</b> Esta historia de usuario permite listar todos los departamentos docentes existentes. |                                                  |
| <b>Observaciones:</b>                                                                                    |                                                  |
| <b>Prototipo de interfaz:</b>                                                                            |                                                  |

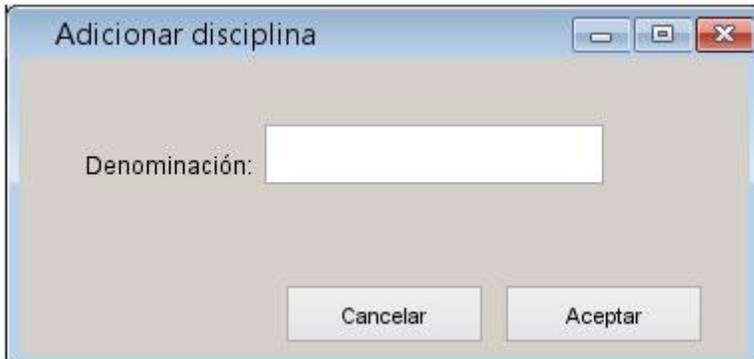
**Anexo 14. Historias de usuario perteneciente a la agrupación por requisitos Gestionar disciplina**

| Historia de usuario: Adicionar disciplina                                                           |                                                   |
|-----------------------------------------------------------------------------------------------------|---------------------------------------------------|
| <b>Número:</b> 5                                                                                    | <b>Nombre del requisito:</b> Adicionar disciplina |
| <b>Programador:</b> Yaima                                                                           | <b>Iteración Asignada:</b> 1                      |
| <b>Prioridad:</b> Alta                                                                              | <b>Tiempo Estimado:</b> 0.5                       |
| <b>Riesgo en Desarrollo:</b> Bajo                                                                   | <b>Tiempo Real:</b> 0.14                          |
| <b>Descripción:</b> Esta historia de usuario permite adicionar una nueva disciplina. En caso de que |                                                   |

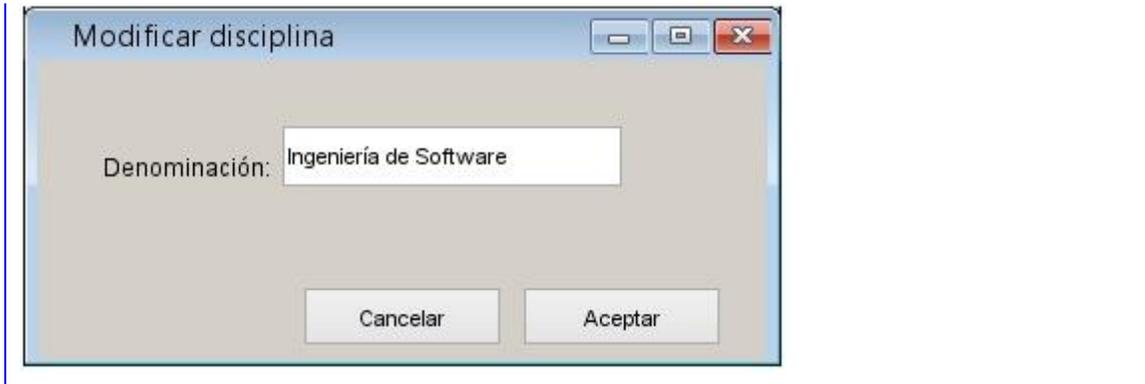
los campos se llenen correctamente introduciendo caracteres válidos se adiciona una nueva disciplina, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos y en caso de que haya campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos.

**Observaciones:**

**Prototipo de interfaz:**



| Historia de usuario: Modificar disciplina                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                   |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| <b>Número:</b> 6                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | <b>Nombre del requisito:</b> Modificar disciplina |
| <b>Programador:</b> Yaima                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | <b>Iteración Asignada:</b> 1                      |
| <b>Prioridad:</b> Alta                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <b>Tiempo Estimado:</b> 0.5                       |
| <b>Riesgo en Desarrollo:</b> Bajo                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | <b>Tiempo Real:</b> 0.14                          |
| <p><b>Descripción:</b> Esta historia de usuario permite modificar una disciplina. En caso de que se realicen cambios y los campos se llenen correctamente introduciendo caracteres válidos se modifica la disciplina, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos, en caso de que se dejen campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos y en caso de que no se realice ningún cambio se muestra al usuario un mensaje indicándole que no se realizó cambios.</p> |                                                   |
| <b>Observaciones:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                   |
| <b>Prototipo de interfaz:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                   |

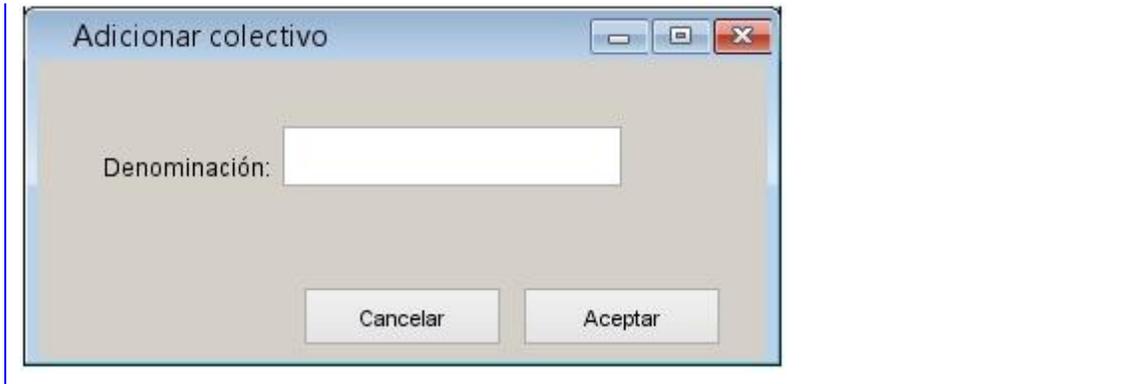


| Historia de usuario: Eliminar disciplina                                                                                                                                                                                |                                                  |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| <b>Número:</b> 7                                                                                                                                                                                                        | <b>Nombre del requisito:</b> Eliminar disciplina |
| <b>Programador:</b> Yaima                                                                                                                                                                                               | <b>Iteración Asignada:</b> 1                     |
| <b>Prioridad:</b> Alta                                                                                                                                                                                                  | <b>Tiempo Estimado:</b> 0.5                      |
| <b>Riesgo en Desarrollo:</b> Bajo                                                                                                                                                                                       | <b>Tiempo Real:</b> 0.14                         |
| <p><b>Descripción:</b> Esta historia de usuario permite eliminar una disciplina. Se selecciona la disciplina que se desea eliminar y se muestra al usuario un mensaje de si está seguro de eliminar esa disciplina.</p> |                                                  |
| <p><b>Observaciones:</b></p>                                                                                                                                                                                            |                                                  |
| <p><b>Prototipo de interfaz:</b></p>                                                                                                                                                                                    |                                                  |

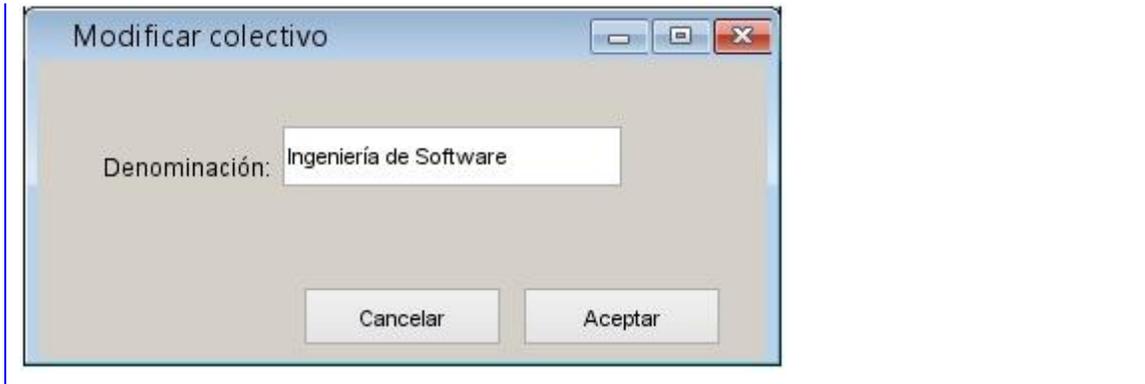
| Historia de usuario: Listar disciplina                                                        |                                          |
|-----------------------------------------------------------------------------------------------|------------------------------------------|
| Número: 8                                                                                     | Nombre del requisito: Listar disciplinas |
| Programador: Yaima                                                                            | Iteración Asignada: 1                    |
| Prioridad: Alta                                                                               | Tiempo Estimado: 0.5                     |
| Riesgo en Desarrollo: Bajo                                                                    | Tiempo Real: 0.14                        |
| <b>Descripción:</b> Esta historia de usuario permite listar todas las disciplinas existentes. |                                          |
| <b>Observaciones:</b>                                                                         |                                          |
| <b>Prototipo de interfaz:</b>                                                                 |                                          |

#### Anexo 15. Historias de usuario perteneciente a la agrupación por requisitos Gestionar colectivo

| Historia de usuario: Adicionar colectivo                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| Número: 9                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Nombre del requisito: Adicionar colectivo |
| Programador: Yaima                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Iteración Asignada: 1                     |
| Prioridad: Alta                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Tiempo Estimado: 0.5                      |
| Riesgo en Desarrollo: Bajo                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Tiempo Real: 0.14                         |
| <b>Descripción:</b> Esta historia de usuario permite adicionar un nuevo colectivo de profesores. En caso de que los campos se llenen correctamente introduciendo caracteres válidos se adiciona un nuevo colectivo de profesores, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos y en caso de que haya campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos. |                                           |
| <b>Observaciones:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                           |
| <b>Prototipo de interfaz:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                           |



| Historia de usuario: Modificar colectivo                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| <b>Número:</b> 10                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <b>Nombre del requisito:</b> Modificar colectivo |
| <b>Programador:</b> Yaima                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | <b>Iteración Asignada:</b> 1                     |
| <b>Prioridad:</b> Alta                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | <b>Tiempo Estimado:</b> 0.5                      |
| <b>Riesgo en Desarrollo:</b> Bajo                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <b>Tiempo Real:</b> 0.14                         |
| <p><b>Descripción:</b> Esta historia de usuario permite modificar un colectivo de profesores. En caso de que se realicen cambios y los campos se llenen correctamente introduciendo caracteres válidos se modifica el colectivo, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos, en caso de que se dejen campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos y en caso de que no se realice ningún cambio se muestra al usuario un mensaje indicándole que no se realizó cambios.</p> |                                                  |
| <b>Observaciones:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                  |
| <b>Prototipo de interfaz:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                  |



| Historia de usuario: Eliminar colectivo                                                                                                                                                                                                         |                                                 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| <b>Número:</b> 11                                                                                                                                                                                                                               | <b>Nombre del requisito:</b> Eliminar colectivo |
| <b>Programador:</b> Yaima                                                                                                                                                                                                                       | <b>Iteración Asignada:</b> 1                    |
| <b>Prioridad:</b> Alta                                                                                                                                                                                                                          | <b>Tiempo Estimado:</b> 0.5                     |
| <b>Riesgo en Desarrollo:</b> Bajo                                                                                                                                                                                                               | <b>Tiempo Real:</b> 0.14                        |
| <p><b>Descripción:</b> Esta historia de usuario permite eliminar un colectivo de profesores. Se selecciona el colectivo de profesores que se desea eliminar y se muestra al usuario un mensaje de si está seguro de eliminar ese colectivo.</p> |                                                 |
| <p><b>Observaciones:</b></p>                                                                                                                                                                                                                    |                                                 |
| <p><b>Prototipo de interfaz:</b></p>                                                                                                                                                                                                            |                                                 |

| Historia de usuario: Listar colectivo                                                                      |                                        |
|------------------------------------------------------------------------------------------------------------|----------------------------------------|
| Número: 12                                                                                                 | Nombre del requisito: Listar colectivo |
| Programador: Yaima                                                                                         | Iteración Asignada: 1                  |
| Prioridad: Alta                                                                                            | Tiempo Estimado: 0.5                   |
| Riesgo en Desarrollo: Bajo                                                                                 | Tiempo Real: 0.14                      |
| <b>Descripción:</b> Esta historia de usuario permite listar todos los colectivos de profesores existentes. |                                        |
| <b>Observaciones:</b>                                                                                      |                                        |
| <b>Prototipo de interfaz:</b>                                                                              |                                        |

#### Anexo 16. Historias de usuario perteneciente a la agrupación por requisitos Gestionar profesor

| Historia de usuario: Adicionar profesor                                                                                                                                                                                                                                                                                                                                                                                                                                |                                          |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------|
| Número: 13                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Nombre del requisito: Adicionar profesor |
| Programador: Yaima                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Iteración Asignada: 1                    |
| Prioridad: Alta                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Tiempo Estimado: 0.5                     |
| Riesgo en Desarrollo: Bajo                                                                                                                                                                                                                                                                                                                                                                                                                                             | Tiempo Real: 0.14                        |
| <b>Descripción:</b> Esta historia de usuario permite adicionar un nuevo profesor. En caso de que los campos se llenen correctamente introduciendo caracteres válidos se adiciona un nuevo profesor, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos y en caso de que haya campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos. |                                          |
| <b>Observaciones:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                          |
| <b>Prototipo de interfaz:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                          |

The image shows a software dialog box titled "Adicionar profesor". It has a standard Windows-style title bar with minimize, maximize, and close buttons. The main area contains four input fields: two text boxes for "Nombre" and "Apellidos", and three dropdown menus for "Categoría docente", "Categoría científica", and "Cargo". At the bottom, there are two buttons: "Cancelar" and "Aceptar".

| Historia de usuario: Modificar profesor                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| <b>Número:</b> 14                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <b>Nombre del requisito:</b> Modificar profesor |
| <b>Programador:</b> Yaima                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | <b>Iteración Asignada:</b> 1                    |
| <b>Prioridad:</b> Alta                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | <b>Tiempo Estimado:</b> 0.5                     |
| <b>Riesgo en Desarrollo:</b> Bajo                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <b>Tiempo Real:</b> 0.14                        |
| <p><b>Descripción:</b> Esta historia de usuario permite modificar un profesor. En caso de que se realicen cambios y los campos se llenen correctamente introduciendo caracteres válidos se modifica el profesor, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos, en caso de que se dejen campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos y en caso de que no se realice ningún cambio se muestra al usuario un mensaje indicándole que no se realizó cambios.</p> |                                                 |
| <b>Observaciones:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                 |
| <b>Prototipo de interfaz:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                 |

Modificar profesor

Nombre:  Categoría docente:

Apellidos:  Categoría científica:

Cargo:

Cancelar Aceptar

| Historia de usuario: Eliminar profesor                                                                                                                                                                           |                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| Número: 15                                                                                                                                                                                                       | Nombre del requisito: Eliminar profesor |
| Programador: Yaima                                                                                                                                                                                               | Iteración Asignada: 1                   |
| Prioridad: Alta                                                                                                                                                                                                  | Tiempo Estimado: 0.5                    |
| Riesgo en Desarrollo: Bajo                                                                                                                                                                                       | Tiempo Real: 0.14                       |
| <p><b>Descripción:</b> Esta historia de usuario permite eliminar un profesor. Se selecciona el profesor que se desea eliminar y se muestra al usuario un mensaje de si está seguro de eliminar ese profesor.</p> |                                         |
| <p><b>Observaciones:</b></p>                                                                                                                                                                                     |                                         |
| <p><b>Prototipo de interfaz:</b></p> <p>Eliminar profesor</p> <p>¿Está seguro que desea eliminar este profesor?</p> <p>Cancelar Aceptar</p>                                                                      |                                         |

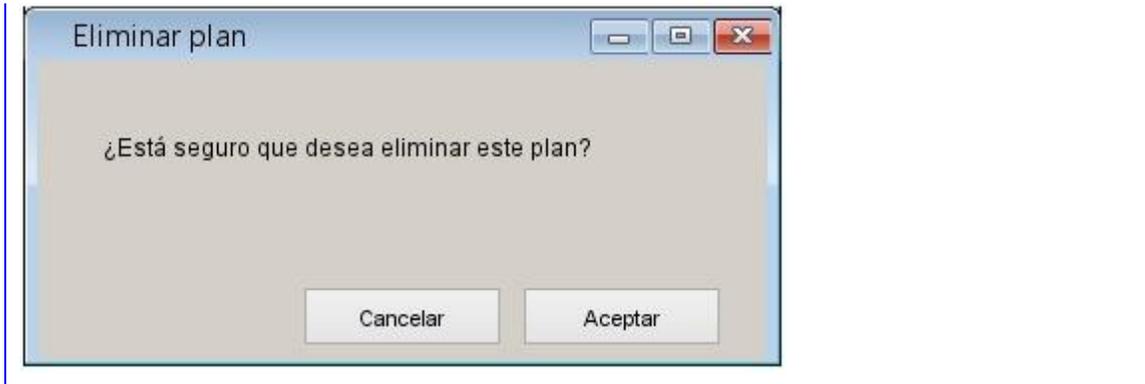
| Historia de usuario: Listar profesor                                                         |                                       |
|----------------------------------------------------------------------------------------------|---------------------------------------|
| Número: 16                                                                                   | Nombre del requisito: Listar profesor |
| Programador: Yaima                                                                           | Iteración Asignada: 1                 |
| Prioridad: Alta                                                                              | Tiempo Estimado: 0.5                  |
| Riesgo en Desarrollo: Bajo                                                                   | Tiempo Real: 0.14                     |
| <b>Descripción:</b> Esta historia de usuario permite listar todos los profesores existentes. |                                       |
| <b>Observaciones:</b>                                                                        |                                       |
| <b>Prototipo de interfaz:</b>                                                                |                                       |

#### Anexo 17. Historias de usuario perteneciente a la agrupación por requisitos Gestionar plan de objetivos del departamento

| Historia de usuario: Adicionar plan de objetivos del departamento (POD)                                                                                                                                                                                                                                                                                                                                                                                      |                                     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
| Número: 17                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Nombre del requisito: Adicionar POD |
| Programador: Yaima                                                                                                                                                                                                                                                                                                                                                                                                                                           | Iteración Asignada: 2               |
| Prioridad: Media                                                                                                                                                                                                                                                                                                                                                                                                                                             | Tiempo Estimado: 0.5                |
| Riesgo en Desarrollo: Alto                                                                                                                                                                                                                                                                                                                                                                                                                                   | Tiempo Real: 0.14                   |
| <b>Descripción:</b> Esta historia de usuario permite adicionar un nuevo POD. En caso de que los campos se llenen correctamente introduciendo caracteres válidos se adiciona un nuevo POD, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos y en caso de que haya campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos. |                                     |
| <b>Observaciones:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                     |
| <b>Prototipo de interfaz:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                |                                     |

| Historia de usuario: Modificar plan de objetivos del departamento (POD)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| <b>Número:</b> 18                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | <b>Nombre del requisito:</b> Modificar POD |
| <b>Programador:</b> Yaima                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | <b>Iteración Asignada:</b> 2               |
| <b>Prioridad:</b> Media                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <b>Tiempo Estimado:</b> 0.5                |
| <b>Riesgo en Desarrollo:</b> Alto                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | <b>Tiempo Real:</b> 0.14                   |
| <p><b>Descripción:</b> Esta historia de usuario permite modificar un plan de objetivos. En caso de que se realicen cambios y los campos se llenen correctamente introduciendo caracteres válidos se modifica el plan de objetivos, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos, en caso de que se dejen campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos y en caso de que no se realice ningún cambio se muestra al usuario un mensaje indicándole que no se realizó cambios.</p> |                                            |
| <b>Observaciones:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                            |
| <b>Prototipo de interfaz:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                            |

| Historia de usuario: Eliminar plan de objetivos del departamento (POD)                                                                                                                                     |                                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| <b>Número:</b> 19                                                                                                                                                                                          | <b>Nombre del requisito:</b> Eliminar POD |
| <b>Programador:</b> Yaima                                                                                                                                                                                  | <b>Iteración Asignada:</b> 2              |
| <b>Prioridad:</b> Media                                                                                                                                                                                    | <b>Tiempo Estimado:</b> 0.5               |
| <b>Riesgo en Desarrollo:</b> Alto                                                                                                                                                                          | <b>Tiempo Real:</b> 0.14                  |
| <b>Descripción:</b> Esta historia de usuario permite eliminar un plan de objetivos. Se selecciona el plan que se desea eliminar y se muestra al usuario un mensaje de si está seguro de eliminar ese plan. |                                           |
| <b>Observaciones:</b>                                                                                                                                                                                      |                                           |
| <b>Prototipo de interfaz:</b>                                                                                                                                                                              |                                           |



| Historia de usuario: Listar plan de objetivos del departamento (POD)                           |                                  |
|------------------------------------------------------------------------------------------------|----------------------------------|
| Número: 20                                                                                     | Nombre del requisito: Listar POD |
| Programador: Yaima                                                                             | Iteración Asignada: 2            |
| Prioridad: Media                                                                               | Tiempo Estimado: 0.5             |
| Riesgo en Desarrollo: Alto                                                                     | Tiempo Real: 0.14                |
| Descripción: Esta historia de usuario permite listar todos los planes de objetivos existentes. |                                  |
| Observaciones:                                                                                 |                                  |
| Prototipo de interfaz:                                                                         |                                  |

**Anexo 18. Historias de usuario perteneciente a la agrupación por requisitos Gestionar plan de resultados**

| Historia de usuario: Adicionar plan de resultados                                            |                                                    |
|----------------------------------------------------------------------------------------------|----------------------------------------------------|
| Número: 21                                                                                   | Nombre del requisito: Adicionar plan de resultados |
| Programador: Yaima                                                                           | Iteración Asignada: 2                              |
| Prioridad: Media                                                                             | Tiempo Estimado: 0.5                               |
| Riesgo en Desarrollo: Alto                                                                   | Tiempo Real: 0.14                                  |
| Descripción: Esta historia de usuario permite adicionar un nuevo plan de resultados. En caso |                                                    |

de que los campos se llenen correctamente introduciendo caracteres válidos se adiciona un nuevo plan de resultados, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos y en caso de que haya campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos.

**Observaciones:**

**Prototipo de interfaz:**

| Historia de usuario: Modificar plan de resultados                                                                                                                                                                                                                                                                                                                                                        |                                                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| Número: 22                                                                                                                                                                                                                                                                                                                                                                                               | Nombre del requisito: Modificar plan de resultados |
| Programador: Yaima                                                                                                                                                                                                                                                                                                                                                                                       | Iteración Asignada: 2                              |
| Prioridad: Media                                                                                                                                                                                                                                                                                                                                                                                         | Tiempo Estimado: 0.5                               |
| Riesgo en Desarrollo: Alto                                                                                                                                                                                                                                                                                                                                                                               | Tiempo Real: 0.14                                  |
| <p><b>Descripción:</b> Esta historia de usuario permite modificar un plan de resultados. En caso de que se realicen cambios y los campos se llenen correctamente introduciendo caracteres válidos se modifica el plan de resultados, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos, en caso</p> |                                                    |

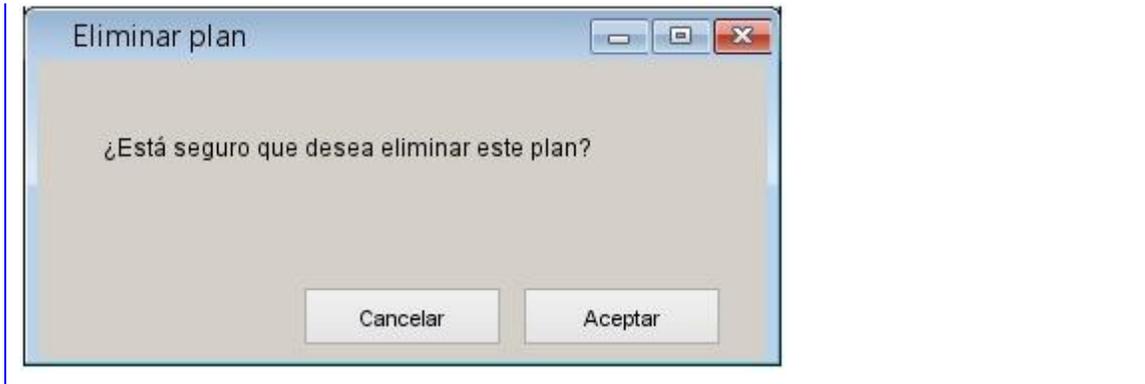
de que se dejen campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos y en caso de que no se realice ningún cambio se muestra al usuario un mensaje indicándole que no se realizó cambios.

**Observaciones:**

**Prototipo de interfaz:**

**Historia de usuario: Eliminar plan de resultados**

|                                                                                                                                                                                                             |                                                          |                              |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|------------------------------|
| <b>Número:</b> 23                                                                                                                                                                                           | <b>Nombre del requisito:</b> Eliminar plan de resultados |                              |
| <b>Programador:</b> Yaima                                                                                                                                                                                   |                                                          | <b>Iteración Asignada:</b> 2 |
| <b>Prioridad:</b> Media                                                                                                                                                                                     |                                                          | <b>Tiempo Estimado:</b> 0.5  |
| <b>Riesgo en Desarrollo:</b> Alto                                                                                                                                                                           |                                                          | <b>Tiempo Real:</b> 0.14     |
| <b>Descripción:</b> Esta historia de usuario permite eliminar un plan de resultados. Se selecciona el plan que se desea eliminar y se muestra al usuario un mensaje de si está seguro de eliminar ese plan. |                                                          |                              |
| <b>Observaciones:</b>                                                                                                                                                                                       |                                                          |                              |
| <b>Prototipo de interfaz:</b>                                                                                                                                                                               |                                                          |                              |



| Historia de usuario: Listar plan de resultados                                                         |                                                        |
|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| <b>Número:</b> 24                                                                                      | <b>Nombre del requisito:</b> Listar plan de resultados |
| <b>Programador:</b> Yaima                                                                              | <b>Iteración Asignada:</b> 2                           |
| <b>Prioridad:</b> Media                                                                                | <b>Tiempo Estimado:</b> 0.5                            |
| <b>Riesgo en Desarrollo:</b> Alto                                                                      | <b>Tiempo Real:</b> 0.14                               |
| <b>Descripción:</b> Esta historia de usuario permite listar todos los planes de resultados existentes. |                                                        |
| <b>Observaciones:</b>                                                                                  |                                                        |
| <b>Prototipo de interfaz:</b>                                                                          |                                                        |

**Anexo 19. Historias de usuario perteneciente a la agrupación por requisitos Gestionar plan mensual del departamento**

| Historia de usuario: Adicionar plan mensual del departamento                             |                                                                      |
|------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| <b>Número:</b> 25                                                                        | <b>Nombre del requisito:</b> Adicionar plan mensual del departamento |
| <b>Programador:</b> Yaima                                                                | <b>Iteración Asignada:</b> 2                                         |
| <b>Prioridad:</b> Media                                                                  | <b>Tiempo Estimado:</b> 0.5                                          |
| <b>Riesgo en Desarrollo:</b> Alto                                                        | <b>Tiempo Real:</b> 0.14                                             |
| <b>Descripción:</b> Esta historia de usuario permite adicionar un nuevo plan mensual del |                                                                      |

departamento. En caso de que los campos se llenen correctamente introduciendo caracteres válidos se adiciona un nuevo plan mensual del departamento, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos y en caso de que haya campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos.

**Observaciones:**

**Prototipo de interfaz:**

**Historia de usuario: Modificar plan mensual del departamento**

|                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                      |  |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|--|
| <b>Número:</b> 26                                                                                                                                                                                                                                                                                                                                                                                     | <b>Nombre del requisito:</b> Modificar plan mensual del departamento |  |
| <b>Programador:</b> Yaima                                                                                                                                                                                                                                                                                                                                                                             | <b>Iteración Asignada:</b> 2                                         |  |
| <b>Prioridad:</b> Media                                                                                                                                                                                                                                                                                                                                                                               | <b>Tiempo Estimado:</b> 0.5                                          |  |
| <b>Riesgo en Desarrollo:</b> Alto                                                                                                                                                                                                                                                                                                                                                                     | <b>Tiempo Real:</b> 0.14                                             |  |
| <p><b>Descripción:</b> Esta historia de usuario permite modificar el plan mensual del departamento. En caso de que se realicen cambios y los campos se llenen correctamente introduciendo caracteres válidos se modifica el plan mensual del departamento, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos</p> |                                                                      |  |

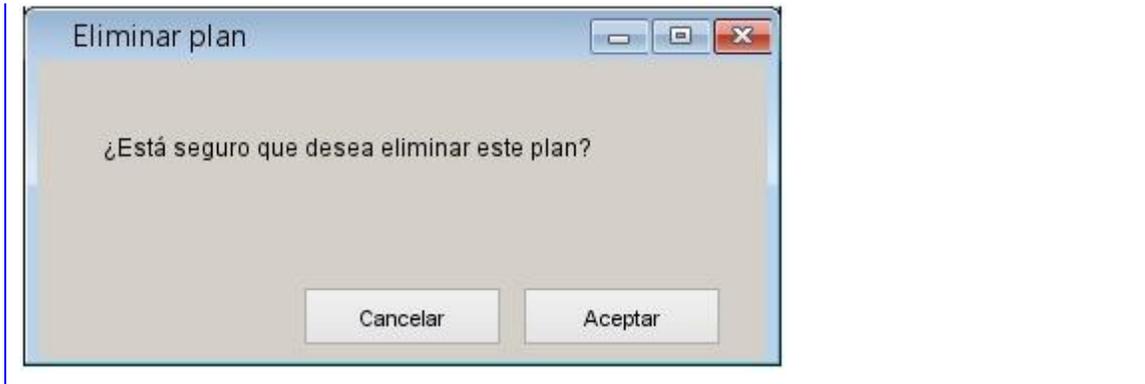
son incorrectos, en caso de que se dejen campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos y en caso de que no se realice ningún cambio se muestra al usuario un mensaje indicándole que no se realizó cambios.

**Observaciones:**

**Prototipo de interfaz:**

**Historia de usuario: Eliminar plan mensual del departamento**

|                                                                                                                                                                                                                        |                                                                     |  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|--|
| <b>Número:</b> 27                                                                                                                                                                                                      | <b>Nombre del requisito:</b> Eliminar plan mensual del departamento |  |
| <b>Programador:</b> Yaima                                                                                                                                                                                              | <b>Iteración Asignada:</b> 2                                        |  |
| <b>Prioridad:</b> Media                                                                                                                                                                                                | <b>Tiempo Estimado:</b> 0.5                                         |  |
| <b>Riesgo en Desarrollo:</b> Alto                                                                                                                                                                                      | <b>Tiempo Real:</b> 0.14                                            |  |
| <b>Descripción:</b> Esta historia de usuario permite eliminar un plan mensual del departamento. Se selecciona el plan que se desea eliminar y se muestra al usuario un mensaje de si está seguro de eliminar ese plan. |                                                                     |  |
| <b>Observaciones:</b>                                                                                                                                                                                                  |                                                                     |  |
| <b>Prototipo de interfaz:</b>                                                                                                                                                                                          |                                                                     |  |



| Historia de usuario: Listar plan mensual del departamento                                                    |                                                            |
|--------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| Número: 28                                                                                                   | Nombre del requisito: Listar plan mensual del departamento |
| Programador: Yaima                                                                                           | Iteración Asignada: 2                                      |
| Prioridad: Media                                                                                             | Tiempo Estimado: 0.5                                       |
| Riesgo en Desarrollo: Alto                                                                                   | Tiempo Real: 0.14                                          |
| Descripción: Esta historia de usuario permite listar todos los planes mensuales del departamento existentes. |                                                            |
| Observaciones:                                                                                               |                                                            |
| Prototipo de interfaz:                                                                                       |                                                            |

**Anexo 20. Historias de usuario perteneciente a la agrupación por requisitos Gestionar plan mensual del profesor**

| Historia de usuario: Adicionar plan mensual del profesor |                                                           |
|----------------------------------------------------------|-----------------------------------------------------------|
| Número: 29                                               | Nombre del requisito: Adicionar plan mensual del profesor |
| Programador: Yaima                                       | Iteración Asignada: 2                                     |
| Prioridad: Media                                         | Tiempo Estimado: 0.5                                      |
| Riesgo en Desarrollo: Alto                               | Tiempo Real: 0.14                                         |

**Descripción:** Esta historia de usuario permite adicionar un nuevo plan mensual del profesor. En caso de que los campos se llenen correctamente introduciendo caracteres válidos se adiciona un nuevo plan mensual del profesor, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos y en caso de que haya campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos.

**Observaciones:**

**Prototipo de interfaz:**

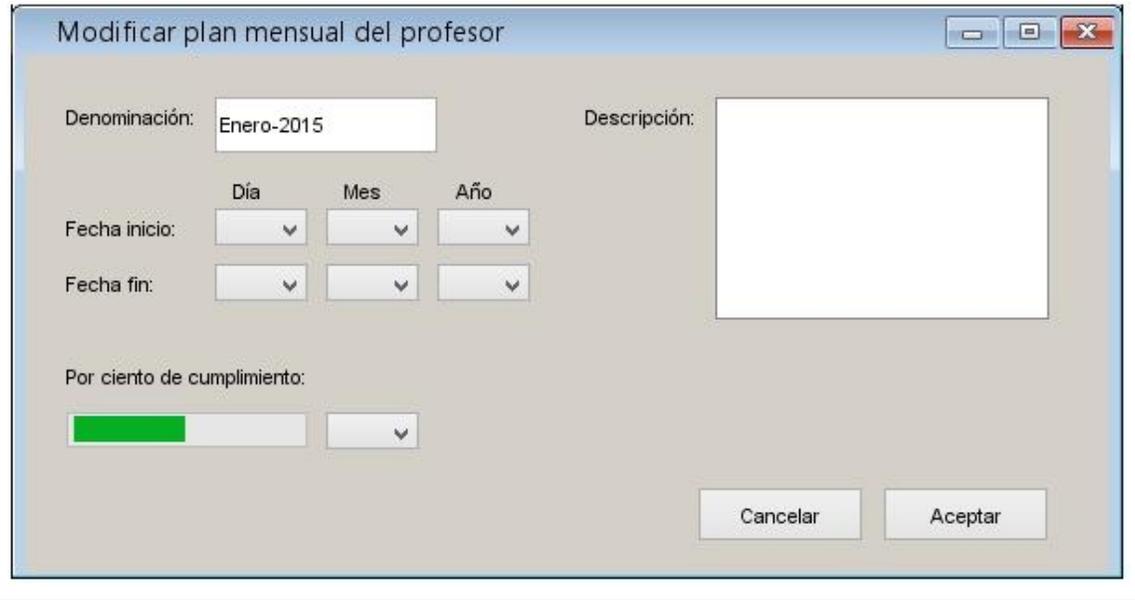
**Historia de usuario: Modificar plan mensual del profesor**

|                                                                                                                                                                                                                                                                                               |                                                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
| <b>Número:</b> 30                                                                                                                                                                                                                                                                             | <b>Nombre del requisito:</b> Modificar plan mensual del profesor |
| <b>Programador:</b> Yaima                                                                                                                                                                                                                                                                     | <b>Iteración Asignada:</b> 2                                     |
| <b>Prioridad:</b> Media                                                                                                                                                                                                                                                                       | <b>Tiempo Estimado:</b> 0.5                                      |
| <b>Riesgo en Desarrollo:</b> Alto                                                                                                                                                                                                                                                             | <b>Tiempo Real:</b> 0.14                                         |
| <b>Descripción:</b> Esta historia de usuario permite modificar el plan mensual del profesor. En caso de que se realicen cambios y los campos se llenen correctamente introduciendo caracteres válidos se modifica el plan mensual del profesor, en caso de que los datos introducidos no sean |                                                                  |

válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos, en caso de que se dejen campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos y en caso de que no se realice ningún cambio se muestra al usuario un mensaje indicándole que no se realizó cambios.

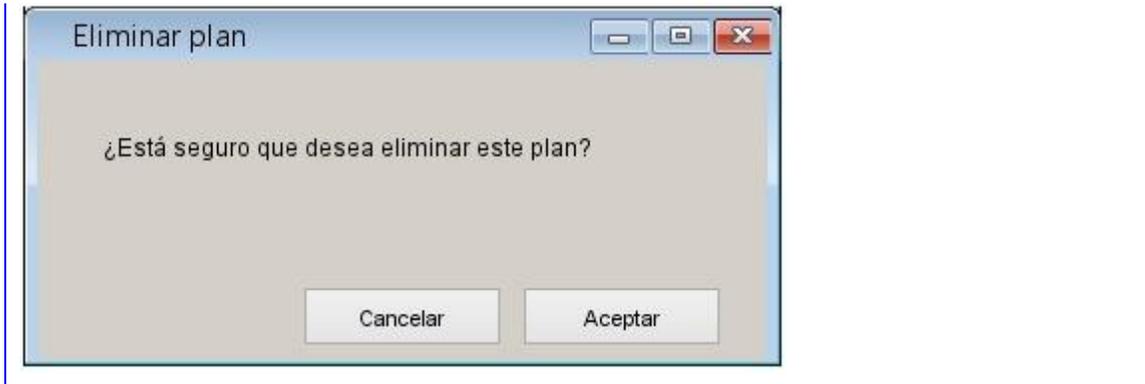
**Observaciones:**

**Prototipo de interfaz:**



**Historia de usuario: Eliminar plan mensual del profesor**

|                                                                                                                                                                                                                    |                                                                 |                              |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|------------------------------|
| <b>Número:</b> 31                                                                                                                                                                                                  | <b>Nombre del requisito:</b> Eliminar plan mensual del profesor |                              |
| <b>Programador:</b> Yaima                                                                                                                                                                                          |                                                                 | <b>Iteración Asignada:</b> 2 |
| <b>Prioridad:</b> Media                                                                                                                                                                                            |                                                                 | <b>Tiempo Estimado:</b> 0.5  |
| <b>Riesgo en Desarrollo:</b> Alto                                                                                                                                                                                  |                                                                 | <b>Tiempo Real:</b> 0.14     |
| <b>Descripción:</b> Esta historia de usuario permite eliminar un plan mensual del profesor. Se selecciona el plan que se desea eliminar y se muestra al usuario un mensaje de si está seguro de eliminar ese plan. |                                                                 |                              |
| <b>Observaciones:</b>                                                                                                                                                                                              |                                                                 |                              |
| <b>Prototipo de interfaz:</b>                                                                                                                                                                                      |                                                                 |                              |



| Historia de usuario: Listar plan mensual del profesor                                                           |                                                               |
|-----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| <b>Número:</b> 32                                                                                               | <b>Nombre del requisito:</b> Listar plan mensual del profesor |
| <b>Programador:</b> Yaima                                                                                       | <b>Iteración Asignada:</b> 2                                  |
| <b>Prioridad:</b> Media                                                                                         | <b>Tiempo Estimado:</b> 0.5                                   |
| <b>Riesgo en Desarrollo:</b> Alto                                                                               | <b>Tiempo Real:</b> 0.14                                      |
| <b>Descripción:</b> Esta historia de usuario permite listar todos los planes mensuales del profesor existentes. |                                                               |
| <b>Observaciones:</b>                                                                                           |                                                               |
| <b>Prototipo de interfaz:</b>                                                                                   |                                                               |

**Anexo 21. Diagrama de clases de la agrupación por requisitos Gestionar objetivos**

| Historia de usuario: Adicionar objetivos                                                             |                                                  |
|------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| <b>Número:</b> 33                                                                                    | <b>Nombre del requisito:</b> Adicionar objetivos |
| <b>Programador:</b> Yaima                                                                            | <b>Iteración Asignada:</b> 3                     |
| <b>Prioridad:</b> Baja                                                                               | <b>Tiempo Estimado:</b> 0.5                      |
| <b>Riesgo en Desarrollo:</b> Alto                                                                    | <b>Tiempo Real:</b> 0.14                         |
| <b>Descripción:</b> Esta historia de usuario permite adicionar un nuevo objetivo. En caso de que los |                                                  |

campos se llenen correctamente introduciendo caracteres válidos se adiciona un nuevo objetivo, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos y en caso de que haya campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos.

**Observaciones:**

**Prototipo de interfaz:**

| Historia de usuario: Modificar objetivos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| <b>Número:</b> 34                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <b>Nombre del requisito:</b> Modificar objetivos |
| <b>Programador:</b> Yaima                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <b>Iteración Asignada:</b> 3                     |
| <b>Prioridad:</b> Baja                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <b>Tiempo Estimado:</b> 0.5                      |
| <b>Riesgo en Desarrollo:</b> Alto                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <b>Tiempo Real:</b> 0.14                         |
| <p><b>Descripción:</b> Esta historia de usuario permite modificar un objetivo. En caso de que se realicen cambios y los campos se llenen correctamente introduciendo caracteres válidos se modifica el objetivo, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos, en caso de que se dejen campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos y en caso de que no se realice ningún cambio se muestra al usuario un mensaje indicándole que no</p> |                                                  |

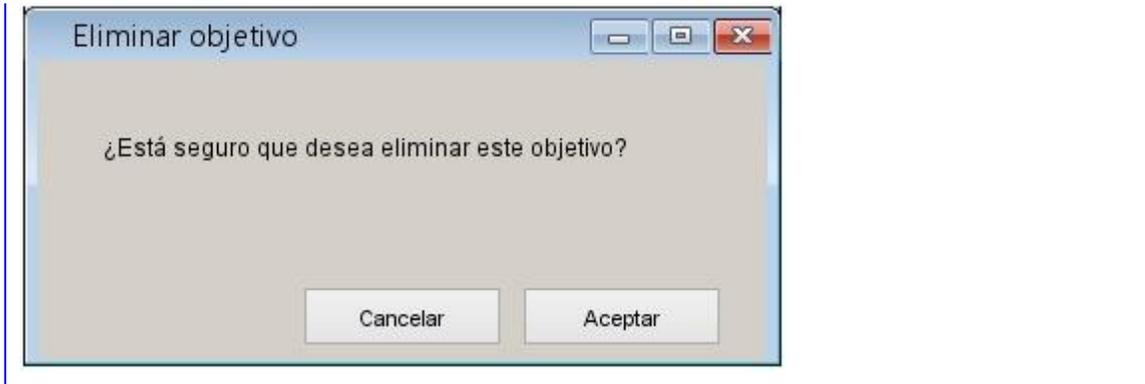
se realizó cambios.

**Observaciones:**

**Prototipo de interfaz:**

**Historia de usuario: Eliminar objetivos**

|                                                                                                                                                                                                           |                                                 |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| <b>Número:</b> 35                                                                                                                                                                                         | <b>Nombre del requisito:</b> Eliminar objetivos |
| <b>Programador:</b> Yaima                                                                                                                                                                                 | <b>Iteración Asignada:</b> 3                    |
| <b>Prioridad:</b> Baja                                                                                                                                                                                    | <b>Tiempo Estimado:</b> 0.5                     |
| <b>Riesgo en Desarrollo:</b> Alto                                                                                                                                                                         | <b>Tiempo Real:</b> 0.14                        |
| <b>Descripción:</b> Esta historia de usuario permite eliminar un objetivo. Se selecciona el objetivo que se desea eliminar y se muestra al usuario un mensaje de si está seguro de eliminar ese objetivo. |                                                 |
| <b>Observaciones:</b>                                                                                                                                                                                     |                                                 |
| <b>Prototipo de interfaz:</b>                                                                                                                                                                             |                                                 |



| Historia de usuario: Listar objetivos                                                       |                                               |
|---------------------------------------------------------------------------------------------|-----------------------------------------------|
| <b>Número:</b> 36                                                                           | <b>Nombre del requisito:</b> Listar objetivos |
| <b>Programador:</b> Yaima                                                                   | <b>Iteración Asignada:</b> 3                  |
| <b>Prioridad:</b> Baja                                                                      | <b>Tiempo Estimado:</b> 0.5                   |
| <b>Riesgo en Desarrollo:</b> Alto                                                           | <b>Tiempo Real:</b> 0.14                      |
| <b>Descripción:</b> Esta historia de usuario permite listar todos los objetivos existentes. |                                               |
| <b>Observaciones:</b>                                                                       |                                               |
| <b>Prototipo de interfaz:</b>                                                               |                                               |

**Anexo 22. Historias de usuario perteneciente a la agrupación por requisitos Gestionar resultado**

| Historia de usuario: Adicionar resultado                                                                                                                                                   |                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| <b>Número:</b> 37                                                                                                                                                                          | <b>Nombre del requisito:</b> Adicionar resultado |
| <b>Programador:</b> Yaima                                                                                                                                                                  | <b>Iteración Asignada:</b> 3                     |
| <b>Prioridad:</b> Baja                                                                                                                                                                     | <b>Tiempo Estimado:</b> 0.5                      |
| <b>Riesgo en Desarrollo:</b> Alto                                                                                                                                                          | <b>Tiempo Real:</b> 0.14                         |
| <b>Descripción:</b> Esta historia de usuario permite adicionar un nuevo resultado. En caso de que los campos se llenen correctamente introduciendo caracteres válidos se adiciona un nuevo |                                                  |

resultado, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos y en caso de que haya campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos.

**Observaciones:**

**Prototipo de interfaz:**

| Historia de usuario: Modificar resultado                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                           |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| Número: 38                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Nombre del requisito: Modificar resultado |
| Programador: Yaima                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Iteración Asignada: 3                     |
| Prioridad: Baja                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Tiempo Estimado: 0.5                      |
| Riesgo en Desarrollo: Alto                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Tiempo Real: 0.14                         |
| <p><b>Descripción:</b> Esta historia de usuario permite modificar un resultado. En caso de que se realicen cambios y los campos se llenen correctamente introduciendo caracteres válidos se modifica el resultado, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos, en caso de que se dejen campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos y en caso de que no se realice ningún cambio se muestra al usuario un mensaje indicándole que no</p> |                                           |

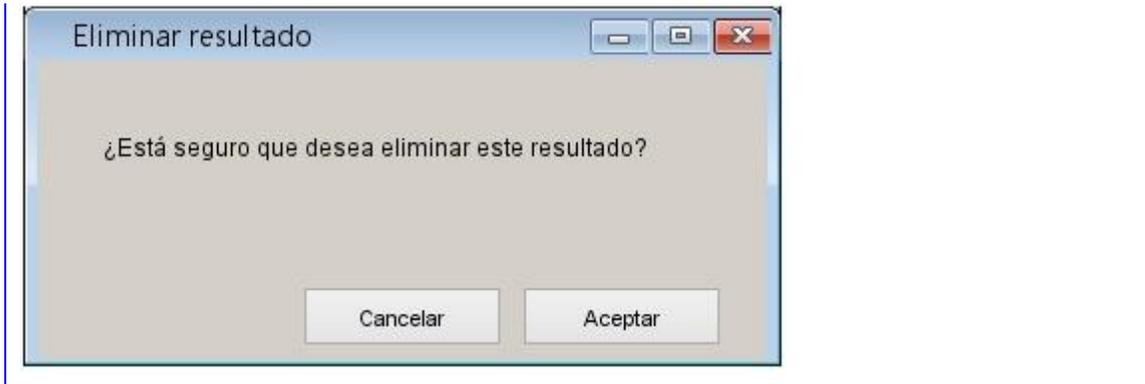
se realizó cambios.

**Observaciones:**

**Prototipo de interfaz:**

**Historia de usuario: Eliminar resultado**

|                                                                                                                                                                                                              |                                                 |                              |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|------------------------------|
| <b>Número:</b> 39                                                                                                                                                                                            | <b>Nombre del requisito:</b> Eliminar resultado |                              |
| <b>Programador:</b> Yaima                                                                                                                                                                                    |                                                 | <b>Iteración Asignada:</b> 3 |
| <b>Prioridad:</b> Baja                                                                                                                                                                                       |                                                 | <b>Tiempo Estimado:</b> 0.5  |
| <b>Riesgo en Desarrollo:</b> Alto                                                                                                                                                                            |                                                 | <b>Tiempo Real:</b> 0.14     |
| <b>Descripción:</b> Esta historia de usuario permite eliminar un resultado. Se selecciona el resultado que se desea eliminar y se muestra al usuario un mensaje de si está seguro de eliminar ese resultado. |                                                 |                              |
| <b>Observaciones:</b>                                                                                                                                                                                        |                                                 |                              |
| <b>Prototipo de interfaz:</b>                                                                                                                                                                                |                                                 |                              |



| Historia de usuario: Listar resultados                                                |                                         |
|---------------------------------------------------------------------------------------|-----------------------------------------|
| Número: 40                                                                            | Nombre del requisito: Listar resultados |
| Programador: Yaima                                                                    | Iteración Asignada: 3                   |
| Prioridad: Baja                                                                       | Tiempo Estimado: 0.5                    |
| Riesgo en Desarrollo: Alto                                                            | Tiempo Real: 0.14                       |
| Descripción: Esta historia de usuario permite listar todos los resultados existentes. |                                         |
| Observaciones:                                                                        |                                         |
| Prototipo de interfaz:                                                                |                                         |

**Anexo 23. Historias de usuario perteneciente a la agrupación por requisitos Gestionar actividad**

| Historia de usuario: Adicionar actividad                                                    |                                           |
|---------------------------------------------------------------------------------------------|-------------------------------------------|
| Número: 41                                                                                  | Nombre del requisito: Adicionar actividad |
| Programador: Yaima                                                                          | Iteración Asignada: 3                     |
| Prioridad: Baja                                                                             | Tiempo Estimado: 0.5                      |
| Riesgo en Desarrollo: Alto                                                                  | Tiempo Real: 0.14                         |
| Descripción: Esta historia de usuario permite adicionar una nueva actividad. En caso de que |                                           |

los campos se llenen correctamente introduciendo caracteres válidos se adiciona una nueva actividad, en caso de que los datos introducidos no sean válidos se muestra un mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos y en caso de que haya campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos.

**Observaciones:**

**Prototipo de interfaz:**

| Historia de usuario: Modificar actividad                                                                                                                                                                                                                                             |                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| <b>Número:</b> 42                                                                                                                                                                                                                                                                    | <b>Nombre del requisito:</b> Modificar actividad |
| <b>Programador:</b> Yaima                                                                                                                                                                                                                                                            | <b>Iteración Asignada:</b> 3                     |
| <b>Prioridad:</b> Baja                                                                                                                                                                                                                                                               | <b>Tiempo Estimado:</b> 0.5                      |
| <b>Riesgo en Desarrollo:</b> Alto                                                                                                                                                                                                                                                    | <b>Tiempo Real:</b> 0.14                         |
| <b>Descripción:</b> Esta historia de usuario permite modificar una actividad. En caso de que se realicen cambios y los campos se llenen correctamente introduciendo caracteres válidos se modifica la actividad, en caso de que los datos introducidos no sean válidos se muestra un |                                                  |

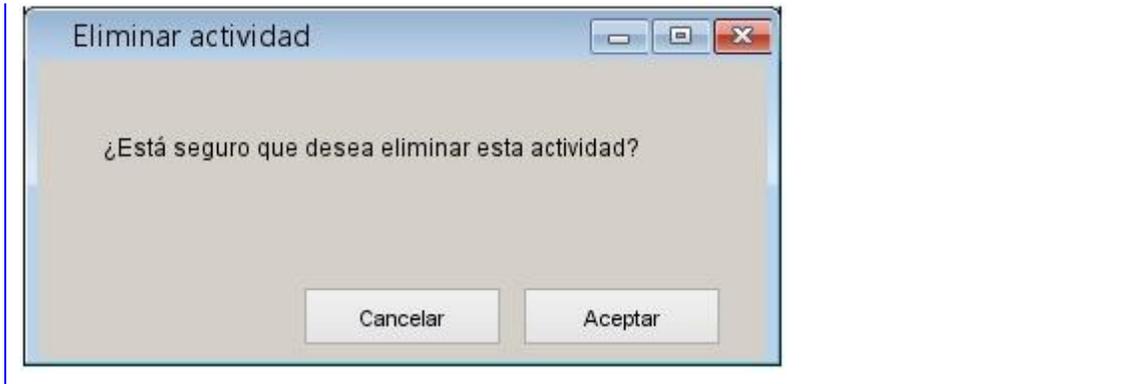
mensaje de aviso indicándole al usuario que los datos introducidos son incorrectos, en caso de que se dejen campos vacíos se muestra un mensaje indicándole al usuario que hay campos en blancos y en caso de que no se realice ningún cambio se muestra al usuario un mensaje indicándole que no se realizó cambios.

**Observaciones:**

**Prototipo de interfaz:**

**Historia de usuario: Eliminar actividad**

|                                                                                                                                                                                                               |                                                 |  |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|--|
| <b>Número:</b> 43                                                                                                                                                                                             | <b>Nombre del requisito:</b> Eliminar actividad |  |
| <b>Programador:</b> Yaima                                                                                                                                                                                     | <b>Iteración Asignada:</b> 3                    |  |
| <b>Prioridad:</b> Baja                                                                                                                                                                                        | <b>Tiempo Estimado:</b> 0.5                     |  |
| <b>Riesgo en Desarrollo:</b> Alto                                                                                                                                                                             | <b>Tiempo Real:</b> 0.14                        |  |
| <b>Descripción:</b> Esta historia de usuario permite eliminar una actividad. Se selecciona la actividad que se desea eliminar y se muestra al usuario un mensaje de si está seguro de eliminar esa actividad. |                                                 |  |
| <b>Observaciones:</b>                                                                                                                                                                                         |                                                 |  |
| <b>Prototipo de interfaz:</b>                                                                                                                                                                                 |                                                 |  |



| Historia de usuario: Listar actividades                                                       |                                                 |
|-----------------------------------------------------------------------------------------------|-------------------------------------------------|
| <b>Número:</b> 44                                                                             | <b>Nombre del requisito:</b> Listar actividades |
| <b>Programador:</b> Yaima                                                                     | <b>Iteración Asignada:</b> 3                    |
| <b>Prioridad:</b> Baja                                                                        | <b>Tiempo Estimado:</b> 0.5                     |
| <b>Riesgo en Desarrollo:</b> Alto                                                             | <b>Tiempo Real:</b> 0.14                        |
| <b>Descripción:</b> Esta historia de usuario permite listar todas las actividades existentes. |                                                 |
| <b>Observaciones:</b>                                                                         |                                                 |
| <b>Prototipo de interfaz:</b>                                                                 |                                                 |

**Anexo 24. Prueba de aceptación**

### Anexo 25. Diagrama de componentes para el sistema

