



**Universidad de las Ciencias Informáticas**

**Facultad 9**

**Subsistema de Departamentos Docentes de la Facultad 9.**

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

**Autores:** Carlos Fidel Montero Hernández.

Yoennis Vega Ricardo.

**Tutor:** Ingeniero Informático, Yoandrys S. Pacheco Jerez.

**Asesor:** Licenciado, José Antonio Peñalver Belancourt.

**Ciudad de La Habana, Julio 2008.**

**Año 50 de la Revolución.**

### AGRADECIMIENTOS

Queremos agradecer a todas las personas que hicieron posible la realización de este trabajo, al Chino, a la figura, al tutor por estar siempre preocupado por nosotros, a los profesores de nuestra facultad. Agradecimiento especial a nuestro Comandante en Jefe por ser el creador y promotor de la Universidad de las Ciencias Informáticas y así permitirnos estudiar en esta magnífica escuela.

#### Carlos

Agradezco de forma muy especial a toda mi familia por siempre darme mucho apoyo, en especial a mi madre, a mi tío Julito, mi tía Ileana, a mis abuelos Antonio y Josefa, a mis primos. Agradecer a los amigos que siempre me ayudaron en la universidad en especial a la chichi, Edgar, Yoennis, a Maye.

#### Yoennis

A toda mi familia que incondicionalmente me apoya, en especial a mi mamá Migdalia lo más grande que tengo en mi vida, que siempre me ha dado su cariño y me ha ayudado en los buenos y malos momentos. A mis abuelos Pablo y Orgelina que son como mis padres, por quererme mucho y por haberme guiado siempre por el camino correcto, por haberme dado fuerza siempre para que continuara adelante, por brindarme todo su amor y ayudarme a ser lo que soy hoy. A mi papá Manolo por haberme aconsejado siempre el camino que debía escoger. A mis tíos Vladimir, Olider y Roberto, por haber sido algo más que mis tíos, han sido como hermanos, por estar siempre atentos y haberme ayudado en todo lo que necesitaba. A mi padrastro Guillermo que ha sido como un padre para mí y siempre ha estado presente cuando he necesitado de su ayuda. A mis hermanos queridos Guillermito, Leodennis, Dailín, Dailé y Dayami por tenerme siempre presente. A mi prima Lidiuska por preocuparse siempre por mí y por todo el cariño que siempre me ha demostrado. A mis tías Yanely, Misleideis y Nani por quererme mucho. A mi abuela Nuris y a mi tía Yaraiza por todo el cariño que han depositado en mí. A los padres de mi novia y su hermano, Mabel, Héctor y Eduardito por su apoyo incondicional en todo este tiempo. A mi novia Bibiana por haberme brindado su amor y haberme acompañado en los buenos y malos momentos. A mi amigo y compañero de tesis Carlos por haberme ayudado en todo momento, a Rodisney, Eliadito, Edgar, Reynier, Angel, Alain, Pedro, Ariel, Yunielkis, Poly, Naná, Marita, Yusleidis, Yadira, Romel y todos aquellos que siempre me han ofrecido su amistad de forma incondicional.

### DEDICATORIA

#### Carlos

A toda mi familia, en especial a mi madre, mi tío Julito, mi tía Ileana, a mis abuelos Antonio y Josefa y a mis primos.

#### Yoennis

A mi novia Bibiana.

A toda mi familia en especial a mi mamá Migdalia, mis abuelos Pablo y Orgelina y a mis hermanos Leodennis y Guillermito.

*“Las que conducen y arrastran al mundo no son las máquinas, sino las ideas.”*

*Victor Hugo*

### RESUMEN

El buen funcionamiento de la gestión de la información en los Departamentos Docentes de la Facultad 9 es de vital importancia para el correcto desempeño docente y educativo de la facultad. En la actualidad la Facultad 9 de la Universidad de las Ciencias Informáticas (UCI) no cuenta con una aplicación que permita gestionar el constante flujo de datos que existe en los Departamentos Docentes de la misma. Por este motivo se hace engorrosa la actualización y obtención de la información necesaria para el buen funcionamiento de los departamentos. Esto ha traído como consecuencia que existan retrasos o pérdida de información por no contar con un medio de almacenamiento de datos. Esta situación dio lugar al desarrollo de una aplicación Web que responde al control y automatización de los procesos que suceden en dichos Departamentos. La puesta en marcha de este sistema, permitirá cambiar la situación existente en los Departamentos Docentes, y así se podrá contar con una mayor confiabilidad, integridad y disponibilidad de la información que allí se maneja. El trabajo está dividido en 4 capítulos que explican como se realiza de forma iterativa el trabajo investigativo. El capítulo 1 trata el tema de la fundamentación teórica del trabajo donde se ven los conceptos asociados al dominio del problema, el objeto de estudio y se analizan otras soluciones existentes. En el capítulo 2 se desarrolla todo lo concerniente a las tendencias tecnológicas actuales en el desarrollo de las aplicaciones Web y se explica porque se usaron el Lenguaje Unificado de Modelado (UML) como soporte de la modelación y el Proceso Unificado de Desarrollo de Software (RUP) como metodología del desarrollo del software. En el capítulo 3 se realiza una presentación de la solución propuesta donde se ve todo lo concerniente al desarrollo del sistema y análisis del producto. El capítulo 4 aborda sobre la construcción de la solución propuesta, donde se ven aspectos relacionados con la implementación del producto.

**PALABRAS CLAVES**

UML.

RUP.

Departamentos Docentes.

Aplicaciones Web.

## ÍNDICE

<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....</b>	<b>5</b>
1.1 INTRODUCCIÓN.....	5
1.2 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA.....	5
1.2.1 Departamentos Docentes.....	5
1.2.2 Plan Metodológico.....	5
1.2.3 Definición de Subsistema.....	6
1.2.4 Procesos.....	6
1.2.5 Programa Analítico (P1).....	6
1.2.6 Gestión de la Información.....	6
1.3 OBJETO DE ESTUDIO.....	7
1.3.1 Descripción General.....	7
1.3.2 Descripción actual del dominio del problema.....	8
1.3.3 Situación Problemática.....	8
1.4 ANÁLISIS DE OTRAS SOLUCIONES EXISTENTES.....	9
1.5 CONCLUSIONES.....	10
<b>CAPÍTULO 2: “TENDENCIAS Y TECNOLOGÍAS ACTUALES PARA EL DESARROLLO DE APLICACIONES WEB”.....</b>	<b>11</b>
2.1 INTRODUCCIÓN.....	11
2.2 CARACTERÍSTICAS DE LAS TECNOLOGÍAS DE APLICACIONES WEB.....	11
2.2.1 Arquitectura Cliente – Servidor.....	11
2.2.2 Tecnologías del lado del Cliente.....	14
2.2.3 Tecnologías del Lado del Servidor.....	15
2.2.4 ¿Por qué PHP?.....	16
2.3 UML (UNIFIED MODELING LANGUAGE).....	17
2.4 METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	18
2.4.1 Características de RUP.....	18
2.4.2 Comparación entre RUP, XP y FDD.....	19
2.4.3 ¿Por qué utilizar RUP como metodología de desarrollo de Software?.....	21

2.5	HERRAMIENTAS CASE.....	22
2.5.1	Características de las Herramientas CASE.....	22
2.5.1.1	Rational Rose.....	22
2.5.1.2	Visual Paradigm.....	22
2.5.1.3	MagicDraw.....	23
2.5.1.4	Umbrello.....	23
2.5.1.5	¿Por qué Rational Rose?.....	23
2.6	DISEÑO DE LA INTERFAZ.....	24
2.6.1	Dreamweaver MX.....	24
2.6.2	¿Qué es Eclipse?.....	25
2.7	FRAMEWORK SYMFONY.....	26
2.7.1	Patrón Modelo Vista Controlador (MVC).....	26
2.8	APACHE COMO SERVIDOR WEB.....	28
2.9	SISTEMA GESTOR DE BASE DE DATOS.....	29
2.9.1	PostgreSQL.....	29
2.9.2	MySQL.....	31
2.9.3	¿Por qué utilizar PostgreSQL?.....	31
2.10	CONCLUSIONES.....	31
<b>CAPÍTULO 3: “PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA”.....</b>		<b>32</b>
3.1	INTRODUCCIÓN.....	32
3.2	ENTORNO DONDE TRABAJARÁ EL SISTEMA INFORMÁTICO.....	32
3.2.1	¿Cuándo aplicar un modelo de Dominio?.....	32
3.2.2	Formas típicas de las clases del dominio.....	32
3.3	¿QUÉ ES MODELO DE DOMINIO?.....	33
3.3.1	Conceptos principales del entorno.....	33
3.3.2	Eventos principales del entorno.....	33
3.3.3	Diagrama de clases del Modelo de Dominio.....	34
3.3.4	Glosario de Términos del Dominio.....	35
3.4	REQUERIMIENTOS.....	35
3.4.1	Requerimientos Funcionales.....	35
3.4.2	Requerimientos No Funcionales.....	37



3.5	DESCRIPCIÓN DEL SISTEMA PROPUESTO.....	38
3.5.1	Descripción de los actores.....	38
3.5.2	Casos de Uso del Sistema.....	40
3.5.2.1	Subsistema Jefe Departamento.....	41
3.5.2.2	Subsistema Jefe Disciplina.....	52
3.5.2.3	Subsistema Jefe Asignatura.....	55
3.5.2.3	Subsistema Profesor Regular.....	60
3.6	CONCLUSIONES.....	63
<b>CAPÍTULO 4 “CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA”.....</b>		<b>64</b>
4.1	INTRODUCCIÓN.....	64
4.2	DIAGRAMA DE CLASES DE DISEÑO WEB.....	65
4.2.1	Subsistema Jefe Departamento.....	65
4.2.2	Subsistema Jefe Disciplina.....	71
4.2.3	Subsistema Jefe de Asignatura.....	72
4.2.4	Subsistema Profesor Regular.....	74
4.3	PRINCIPIOS DE DISEÑO.....	75
4.3.1	Estándares de la interfaz de la aplicación.....	75
4.3.2	Tratamiento de Errores.....	75
4.3.3	Generación de Reportes.....	75
4.3.4	Concepción general de la ayuda.....	76
4.4	DISEÑO DE LA BASE DE DATOS.....	76
4.5	GENERALIDADES DE LA IMPLEMENTACIÓN.....	79
4.6	MODELO DE DESPLIEGUE.....	79
4.7	MODELO DE IMPLEMENTACIÓN.....	81
4.8	CONCLUSIONES.....	82
<b>CONCLUSIONES.....</b>		<b>83</b>
<b>RECOMENDACIONES.....</b>		<b>84</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>		<b>85</b>
<b>BIBLIOGRAFÍA.....</b>		<b>86</b>

<b>GLOSARIO .....</b>	<b>87</b>
<b>ANEXOS .....</b>	<b>89</b>

### INTRODUCCIÓN

La Universidad de las Ciencias Informáticas (UCI) surgida en el 2002 por iniciativa de nuestro Comandante en Jefe Fidel Castro Ruz, es uno de los centros rectores que llevan a cabo la informatización de la sociedad. Este centro desde su surgimiento ha contado con varias facultades que son las encargadas de llevar todo el control docente de los estudiantes y de la universidad en general, la cual cuenta para realizar un mejor trabajo con cinco Departamentos Docentes.

Los Departamentos Docentes son las entidades que controlan todo lo referente al trabajo docente de una facultad, en ellos se organizan los profesores según las asignaturas que imparten en cinco Departamentos Docentes, los cuales son: Departamento de Humanidades, Departamento de Ciencias Básicas, Departamento de Ingeniería de Software y PP, Departamento de Programación y Departamento de Sistemas Digitales, cada departamento está compuesto por el Jefe de Departamento, los Jefes de Disciplina y Asignaturas y los demás profesores que trabajan en cada uno de los departamentos.

El reporte y control diario de la información es uno de los procesos más importantes que se realizan para llevar a cabo el óptimo funcionamiento del trabajo docente en la facultad. En este momento el flujo de la estadística se realiza de forma manual lo que trae consigo atrasos en la actualización de la información y en algunas ocasiones pérdidas irreparables. Por toda la problemática existente, la dirección de la facultad, decidió diseñar e implementar una aplicación que permita la gestión y control de todos los procesos que allí se manejan, como vía para alcanzar mejoras en la eficiencia y control de los Departamentos Docentes. La aplicación forma parte del polo de Informatización de la Facultad 9, que se encarga de gestionar los recursos y la información de la facultad.

La implementación de este sistema facilita la eficiencia en el trabajo hasta tal punto que se haría mucho más rápido y organizado el control de la estadística, puesto que toda la información quedaría almacenada en un historial, ya sea cualitativo o cuantitativo, lo cual posibilitaría una mejor organización del trabajo en la facultad. Como propuesta de uso se tiene esta herramienta para su futura implementación y explotación, logrando así, una vinculación directa y homogénea de estas hacia todos los Departamentos Docentes. De ahí que la principal tarea estaría encaminada a resolver el siguiente **problema científico**:

¿Cómo facilitar la Gestión docente de la estadística, los documentos y flujo de información que en los Departamentos Docentes de la facultad 9 se manejan?

Se esperan, como aportes prácticos, dotar a la facultad 9 de una aplicación informática que permita el

desarrollo de un sistema funcional del subsistema de gestión y control de las actividades de los Departamentos Docentes de la misma y la documentación completa UML del Subsistema de gestión de la actividad docente.

En el trabajo el **objeto de estudio** es el proceso de gestión de la información de los Departamentos Docentes de la Facultad 9. El **campo de acción** es la gestión de la información de los Departamentos Docentes de la Facultad 9, el cual incluye tanto a las personas encargadas de la planificación y el control de la información los cuales serían los Jefes de departamento, ya que tienen el control más estricto de los mismos y controlan toda la actividad relacionada con su departamento; así como los Jefes de disciplina y Jefes de asignatura que a su vez y en menor incidencia tienen que hacer controles e informes de sus determinadas áreas; además se encuentran los profesores pertenecientes al mismo, quienes tienen que entregar constantemente informes y realizar tareas definidas por el departamento. De esta forma el sistema se realizará sobre una plataforma Web, el mismo tendrá una gran importancia desde el punto de vista técnico debido a que agilizará en gran medida la gestión de la información, lo cual traerá consigo que el trabajo en los departamentos sea más eficiente y hará más factible el control de las actividades realizadas en los mismos.

Partiendo de la Hipótesis de que si se realiza un análisis, diseño e implementación para el desarrollo de una aplicación Web sobre la gestión de la información referente a los Departamentos Docentes entonces disminuirán los retrasos y pérdidas de los procesos que se manejan en la Facultad 9; el objetivo general del trabajo es:

### **Objetivo General:**

Desarrollar un sistema informático sobre plataforma Web que permita el control de la estadística, así como los documentos y flujo de información de los departamentos docentes de la facultad 9.

La hipótesis está compuesta por dos tipos de variables, dependientes e independientes en las cuales se establecen la relación causa - efecto.

### **Variable independiente:(Causa)**

Análisis, diseño e implementación para el desarrollo de una aplicación Web.

### **Variable dependiente: (Efecto)**

Disminución en los retrasos de los procesos que se manejan en los Departamentos Docentes de la Facultad 9.

### Objetivos Específicos:

1. Documentar el flujo de información de los Departamentos Docentes en la Facultad 9.
2. Establecer las modificaciones necesarias al flujo de información de los Departamentos Docentes de la Facultad 9 logrando su optimización máxima.
3. Diseñar y poner en explotación un sistema informático que de respuesta al flujo de información de los Departamentos Docentes de la Facultad 9.
4. Validar los resultados obtenidos de la investigación científica.

### Tareas.

1. Estudio de las condiciones actuales en las cuales se desarrolla el proceso de control de las estadísticas, documentos y flujo de información entre los Departamentos Docentes de la Facultad 9.
2. Estudio del arte de las principales y más actuales tecnologías utilizadas para el diseño y desarrollo de aplicaciones Web en Cuba y el mundo.
3. Estudio de estándares nacionales e internacionales para el desarrollo de aplicaciones Web.
4. Diseño del subsistema para la Gestión docente de la estadística, los documentos y flujo de información que en los Departamentos Docentes de la Facultad 9 se manejan.
5. Planteamiento del flujo de información, sus nodos centrales, así como los datos de entrada y reportes para la actividad de control de las estadísticas, documentos y flujo de la información que se maneja en los Departamentos Docentes de la Facultad 9.
6. Desarrollo de un prototipo funcional para Gestionar el control de la estadística, así como los documentos y flujo de información de los Departamentos Docentes de la Facultad 9.
7. Validar los resultados obtenidos de la investigación científica.

Lo que se propone con la implementación del sistema es la realización de la documentación de los procesos que sostienen los Departamentos Docentes en la facultad, sus mejoras potenciales para un funcionamiento óptimo y un Sistema Web para la informatización de la actividad de los Jefes de Departamentos Docentes. Para llevar a cabo estas tareas se emplean métodos empíricos y teóricos de

la investigación científica. Dentro de los métodos empíricos se encuentra la entrevista, la cual posibilita obtener la información referente a como se espera que funcione la aplicación Web. La entrevista será realizada no sólo al cliente, sino también a todo el personal involucrado en el proceso. Dentro del método teórico se utiliza el de modelación, debido a que es necesario modelar el sistema en un lenguaje de modelado para entender de forma más profunda y particular los procesos que se realizan en los Departamentos Docentes. Entre los métodos teóricos también se utiliza el análisis histórico-lógico para analizar a nivel nacional e internacional el empleo de sistemas informáticos similares al que se propone, así como investigaciones realizadas anteriormente sobre las aplicaciones Web.

Para la implementación del sistema sobre plataforma Web, se utiliza como gestor de bases de datos PostgreSQL, como servidor de aplicaciones Web Apache. Para su implementación y puesta en funcionamiento se utiliza el Framework Symfony bajo el lenguaje de programación del lado del Servidor PHP, además de otras herramientas fundamentales para el diseño de la aplicación como Dreamweaver, UML y el Rational Rose.

### **CAPÍTULO 1: Fundamentación Teórica.**

#### **1.1 Introducción.**

En este capítulo se hace una descripción general de los procesos que se realizan en los Departamentos Docentes de la Facultad 9 y se identifican los principales motivos que dieron lugar a la realización de la aplicación Web, se ven algunos conceptos asociados con el problema a resolver, se da una explicación detallada del objeto de estudio de la investigación, se realiza un análisis amplio de la situación problémica existente y además se hace el análisis de otras soluciones existentes.

#### **1.2 Conceptos asociados al dominio del problema.**

##### **1.2.1 Departamentos Docentes.**

Es donde se organizan los trabajadores y profesores los cuales imparten una determinada asignatura en correspondencia con el departamento al cual pertenezcan. Estos departamentos son: Departamento de Humanidades, Departamento de Ciencias Básicas, Departamento de Ingeniería de Software y PP, Departamento de Programación y Departamento de Sistemas Digitales. En dichos departamentos trabajan una serie de personas que garantizan el cumplimiento de las actividades que se realizan en la facultad.

**Jefe de Departamento:** Es el máximo responsable de controlar las actividades de un departamento específico, es el encargado de realizar el plan de trabajo del departamento.

**Jefe de Disciplina:** Controla toda la información de una disciplina docente, se entiende como disciplina un conjunto de asignaturas de la misma rama de estudio. Es el encargado de realizar el plan de trabajo de la disciplina. Participa en las reuniones metodológicas.

**Jefe de Asignatura:** Es el encargado de controlar y revisar los planes de trabajo de los profesores, lleva el control de las evaluaciones de los mismos y desarrolla el plan de trabajo de la asignatura.

**Profesores:** Son los responsables de chequear las actividades asignadas a los alumnos ayudantes que él atiende y realiza su plan de trabajo individual. [1]

##### **1.2.2 Plan Metodológico.**

Documento que recoge todo lo que el profesor debe hacer en varios aspectos relacionados con la docencia, como Preparación Metodológica, atención de Alumnos Ayudantes, Superación, Extensión Universitaria e Investigación.

### 1.2.3 Definición de Subsistema.

Un subsistema es pues una caja negra (que encapsula y oculta), responsable de gestionar información, y de proporcionar un conjunto de servicios, para lo cual, en general, requerirá colaboraciones de otros subsistemas. [2]

Un subsistema es un entorno operativo único y predefinido a través del cual el sistema coordina el flujo de trabajo y la utilización de recursos. El sistema puede contener varios subsistemas, todos operando independientemente de los demás. Los subsistemas gestionan recursos. Las características del tiempo de ejecución de un subsistema se definen en un objeto llamado descripción de subsistema. [3]

### 1.2.4 Procesos.

Acción o sucesión de acciones continuas regulares, que ocurren o se llevan a cabo de una forma definida, y que llevan al cumplimiento de algún resultado; una operación continua o una serie de operaciones. [4]

Los procesos son grupos de actividades y tareas que juntas entregan valor al cliente, involucran muchas personas y departamentos y transforman entradas en salidas. Estos están enfocados al cliente y a los resultados. [4]

Un proceso es una serie de actividades relacionadas y ejecutadas con lógica para alcanzar resultados específicos. [4]

### 1.2.5 Programa Analítico (P1).

Es el documento en el cual se encuentran reflejados todos los objetivos de las asignaturas del semestre, así como la distribución de las mismas en Conferencias., Clases Prácticas, Seminarios, Talleres, Laboratorios y otras actividades docentes.

### 1.2.6 Gestión de la Información.

La gestión de la información se puede definir como el conjunto de actividades realizadas con el fin de controlar, almacenar y, posteriormente recuperar adecuadamente la información producida, recibida o retenida por cualquier organización en el desarrollo de sus actividades. [11]



### Los Objetivos de la Gestión de la Información son:

- Maximizar el valor y los beneficios derivados del uso de la información
- Minimizar el costo de adquisición, procesamiento y uso de la información.
- Determinar responsabilidades para el uso efectivo, eficiente y económico de la información.
- Asegurar un suministro continuo de la información.

### 1.3 Objeto de Estudio.

#### 1.3.1 Descripción General.

La Universidad de las Ciencias Informáticas, surgida al calor de la Batalla de Ideas es una de las instituciones rectoras en la producción y exportación del software del país, estos avances no fueran posibles si no existiera una sólida infraestructura docente en cada una de sus facultades. Los Departamentos Docentes que existen en las facultades son los que llevan el control de todo el trabajo académico de las mismas, por lo que el objeto de estudio constituye: El proceso de gestión de la información de los Departamentos Docentes de la Facultad 9. Los mismos se dividen en cinco Departamentos Docentes, Departamento de Humanidades, en el cual se agrupan una serie de disciplinas como son: Ciencias Sociales, Preparación para la Defensa e Idioma Extranjero, y dentro de ellas se encuentran asignaturas como Inglés 1,2,3,4 y 5, Problemas Sociales, Preparación para Defensa 1y 2, Economía Política 1 y 2 ; el Departamento de Ciencias Básicas, en cual se encuentran las disciplinas de Matemática, Física y Ciencias Empresariales, dentro de ellas se encuentran asignaturas como Física 1 y 2, Contabilidad y Finanzas, Probabilidades Estadísticas, Matemática Discreta, Administración de Empresa, Comercio Electrónico, Álgebra Lineal y Matemática 1,2,3,4; el Departamento de Ingeniería de Software y PP, que recoge las disciplinas de Práctica Profesional e Ingeniería de Software que recoge asignaturas como Práctica Profesional 1,2,3,4,5, Ingeniería de Software1,2, Metodología de la Investigación Científica, Base de Datos y Gestión de Software; el Departamento de Programación, que recoge la disciplina de Programación y asignaturas como Introducción a la Programación, Gráfico por Computadoras, Programación 1,2,3,4 e Inteligencia Artificial; el Departamento de Sistemas Digitales que recoge disciplinas como Teleinformática y Máquinas Computadoras y asignaturas como Teleinformática 1y2, Máquinas Computadoras 1y2, Seguridad Informática, Sistema Operativo y las Operativas correspondientes al perfil de la facultad. Los trabajadores que participan activamente en estos departamentos son: los Jefes de Departamentos, los cuales tienen un control más estricto sobre los procesos que intervienen en su departamento, debido a que controlan todas las actividades que corresponden al mismo. Otros trabajadores que intervienen

son los Jefes de Disciplina y los Jefes de Asignatura, los cuales deben hacer controles e informes de sus determinadas áreas. Además se encuentran los Profesores que también deben entregar informes correspondientes al proceso docente y realizar las tareas definidas por el departamento.

### **1.3.2 Descripción actual del dominio del problema.**

Para la realización del sistema es necesario indagar y conocer las interioridades del entorno al cual le va a servir la aplicación. Los Departamentos Docentes están compuestos por los Jefes de Departamentos que son los encargados de realizar y llevar el control del plan metodológico que llevará a cabo cada uno de los departamentos donde trabaja. En este documento se organiza el funcionamiento del departamento durante el curso, en este se recogen las deficiencias del curso anterior relacionadas con las disciplinas y las asignaturas del departamento en cuestión, de estas insuficiencias se trazan una serie de medidas que ayudan al correcto funcionamiento del departamento durante el siguiente curso, aquí surgen además las nuevas líneas de trabajo y se discuten los objetivos fundamentales del presente curso escolar. Los Jefes de Departamento son los encargados de realizar el plan de trabajo del departamento. Los Jefes de Disciplina son los encargados de realizar los planes de trabajo de las diferentes disciplinas, además se encargan de la evaluación de los Jefes de Asignatura y las tareas de la disciplina. El Jefe de Asignatura se rige por el plan de trabajo de la disciplina para elaborar el plan de trabajo de la asignatura donde se tienen en cuenta el programa y las características de la misma, así como el desglose de los contenidos de la asignatura (P1). El Profesor es otro de los que intervienen en el flujo de procesos el cual se guía por los documentos que elabora el Jefe de la asignatura para elaborar su plan de trabajo individual, por el cual se guiará el Jefe de Asignatura para asignarle una evaluación en dependencia a las tareas y actividades planificadas en su plan de trabajo. Además de estas tareas el profesor se encarga de tutorear al alumno ayudante que imparte su asignatura dándole una evaluación en dependencia del comportamiento del estudiante según las tareas asignadas. El plan de trabajo del departamento incluye los controles a clases, los cuales reportan los informes de los chequeos que realizan los profesores en los turnos de clases, estos chequeos pueden ser de dos tipos:

1. De comprobación.
2. De inspección: dentro de este se encuentra la planificada y la sorpresiva.

### **1.3.3 Situación Problemática.**

El flujo constante de la información es un proceso que se lleva a cabo de forma diaria y sistemática en los Departamentos Docentes de la Facultad 9, planificando, organizando y controlando una serie de

actividades. Los problemas que existen en estos departamentos son: los procesos se realizan de forma manual, lo que trae como consecuencia diversos inconvenientes para todas las personas que intervienen en la realización de dicho proceso, algunos de estos problemas son: Pérdida de la información al no lograr la centralización y almacenamiento de la mayoría de las actividades y eventos; demora excesiva en la entrega de datos e informaciones importantes para llevar a cabo el buen funcionamiento del proceso docente de la facultad; no se conocen los resultados de muchas actividades que se han orientado a realizar con anterioridad; no existe una vía para informar de todos los eventos y procesos que ocurren en el departamento de forma diaria, así como la obtención de información relacionada con los profesores y alumnos ayudantes que integran dichos departamentos. En la actualidad la Facultad 9 de la Universidad de las Ciencias Informáticas no cuenta con un sistema capaz de gestionar la información que fluye por sus Departamentos Docentes, por lo que el principal problema a resolver es la creación de un sistema informático que permita gestionar el control de la estadística, los documentos y flujo de información de los Departamentos Docentes de la Facultad 9.

### **1.4 Análisis de otras soluciones existentes.**

En cursos anteriores se le trató de dar solución al problema de la gestión y automatización de la información de los principales procesos que se producen en los Departamentos Docentes de la Facultad 9 por medio de un prototipo funcional creado por las ingenieras Anay Iyenis Chapman Hernández y Dailien García Pérez. El objetivo de este sistema consistía en controlar el flujo de datos y estadísticas que se manejaban en los Departamentos Docentes, pero no se llegó a aplicar debido a que no se llegaron a automatizar todos los requisitos funcionales requeridos para el buen funcionamiento de la aplicación, este prototipo funcional fue realizado sobre plataforma .NET utilizando el lenguaje de programación C# y además se escogió como sistema gestor de Bases de Datos SQLServer 2000. A pesar de que las herramientas utilizadas ofrecen un ambiente de desarrollo muy seguro y práctico, la aplicación no llegó a su desarrollo óptimo debido a problemas de coordinación y de tiempo. Además se realizó el análisis de una tesis que aborda el tema de un Sistema de Gestión de Información para el Departamento Central de Ingeniería de Software realizada por las ingenieras Yailín Estrada Herrera y Lenna Carballo Muñoz, la misma permitió crear un sistema automatizado para controlar la gestión de la información en ese departamento docente central. Se indagó porqué se utilizaron tecnologías de desarrollo Web, como por ejemplo PHP como lenguaje de programación y PostgreSQL como sistema gestor de Base de Datos. Esta aplicación se ejecutó de forma exitosa y le permitió contar al Departamento Decente Central de Ingeniería de Software con un software capaz de gestionar todo el flujo de información que allí se sucede. De las dos soluciones analizadas se hizo más

énfasis en la segunda debido a que fue desarrollada bajo las políticas de desarrollo de software libre que es uno de los objetivos que persigue la investigación.

### **1.5 Conclusiones.**

Al no contar la Facultad 9 con un sistema que gestione la información de los Departamentos Docentes, se hizo necesario analizar las causas que imposibilitaban un mejor tratamiento de la información en sus departamentos. En este capítulo se han descrito los conceptos asociados al dominio del problema, para poder entender mejor el objetivo del trabajo, se enfatiza en los conceptos de Departamentos Docentes, Procesos, Plan Metodológico, Subsistema, Programa Analítico y Gestión de Información. Además se aborda sobre el objeto de estudio, donde se hizo una descripción general de la infraestructura en la cual se va a trabajar. Se hace referencia al actual dominio del problema para indagar sobre las interioridades del entorno al cual le va a servir la aplicación. En este capítulo se estudia la situación problemática para ver las causas de porque es necesario un subsistema que gestione la información en los Departamentos Docentes de la Facultad 9. Además de esto se analizan otras soluciones existentes que tratan de dar solución a la problemática existente.

### **CAPÍTULO 2: “Tendencias y tecnologías actuales para el desarrollo de aplicaciones Web”.**

#### **2.1 Introducción.**

En este capítulo se ven las principales tendencias y tecnologías usadas en las aplicaciones Web. Se abordan temas relacionados con las tecnologías del lado del cliente y del servidor. Se explica porque se utilizó PHP como lenguaje de programación, RUP como metodología de desarrollo de software y UML como lenguaje de modelado. Además se describe a PostgreSQL como sistema gestor de base de datos y se hace referencia al framework Symfony.

#### **2.2 Características de las tecnologías de Aplicaciones Web.**

##### **2.2.1 Arquitectura Cliente – Servidor.**

La arquitectura Cliente-Servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

En este modelo las aplicaciones se dividen de forma tal que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Los principales componentes del modelo Cliente-Servidor son: los clientes, los servidores y la infraestructura de comunicación.

Un cliente es el que inicia un requerimiento de servicio. El requerimiento inicial puede convertirse en múltiples requerimientos de trabajo. La ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente. Los clientes frecuentemente se comunican con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad.

**Los clientes realizan generalmente funciones como:**

- Manejo de la interfaz de usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.

**Por su parte los servidores realizan, entre otras, las siguientes funciones:**

- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local.

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y este le responde proporcionándolo, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

Los servidores proporcionan un servicio al cliente y devuelven los resultados, deben manejar servicios como administración de la red, mensajes, control y administración de la entrada al sistema, auditoría, recuperación y contabilidad.

Para que los clientes y los servidores puedan comunicarse se requiere de una infraestructura de comunicación o protocolo de comunicación, la cual proporciona los mecanismos básicos de direccionamiento y transporte.

Se denomina protocolo al conjunto de reglas que controlan la secuencia de mensajes que ocurren durante una comunicación entre entidades que forman una red. En este contexto, las entidades de las cuales se habla son programas de computadora o autómatas de otro tipo, tales como dispositivos electrónicos capaces de interactuar en una red. Existen muchos tipos de protocolos cada uno con sus reglas bien definidas, como por ejemplo: HTTP, FTP, POP3, SMTP, ICMP.

**Entre las principales características de la arquitectura Cliente-Servidor, se destacan las siguientes:**

- El cliente y el servidor pueden actuar como una sola entidad y también como entidades separadas, realizando actividades o tareas independientes.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.
- Las funciones de cliente y servidor pueden estar en plataformas separadas o en la misma plataforma.

### **Ventajas:**

- Aumento de la productividad: Los usuarios pueden utilizar herramientas que le son familiares, como hojas de cálculo y herramientas de acceso a bases de datos. Mediante la integración de las aplicaciones cliente-servidor con las aplicaciones personales de uso habitual, los usuarios pueden construir soluciones particularizadas que se ajusten a sus necesidades cambiantes con menores costes de operación.
- Permiten un mejor aprovechamiento de los sistemas existentes, protegiendo la inversión. Por ejemplo, la compartición de servidores habitualmente caros y dispositivos periféricos como impresoras entre máquinas clientes permite un mejor rendimiento del conjunto.
- Proporcionan un mejor acceso a los datos. La interfaz de usuario ofrece una forma homogénea de ver el sistema, independientemente de los cambios o actualizaciones que se produzcan en él y de la ubicación de la información.
- El movimiento de funciones desde un ordenador central hacia servidores o clientes locales origina el desplazamiento de los costes de ese proceso hacia máquinas más pequeñas y por tanto, más baratas.
- Mejoras en el rendimiento de la red.
- Las arquitecturas cliente-servidor eliminan la necesidad de mover grandes bloques de información por la red hacia los ordenadores personales o estaciones de trabajo para su proceso. Los servidores controlan los datos, procesan peticiones y después transfieren sólo los datos requeridos a la máquina cliente. Entonces, la máquina cliente presenta los datos al usuario mediante interfaces amigables. Todo esto reduce el tráfico de la red, lo que facilita que pueda soportar un mayor número de usuarios.
- Tanto el cliente como el servidor pueden escalarse para ajustarse a las necesidades de las aplicaciones.
- En una arquitectura como esta, los clientes y los servidores son independientes los unos de los otros con lo que pueden renovarse para aumentar sus funciones y capacidad de forma independiente, sin afectar al resto del sistema.

### **Inconvenientes:**

- Hay una alta complejidad tecnológica al tener que integrar una gran variedad de productos.
- Requiere un fuerte rediseño de todos los elementos involucrados en los sistemas de información (modelos de datos, procesos, interfaces, comunicaciones y almacenamiento de datos). Además, en la actualidad existen pocas herramientas que ayuden a determinar la mejor forma de dividir las aplicaciones entre la parte cliente y la parte servidor.
- Es más difícil asegurar un elevado grado de seguridad en una red de clientes y servidores que en un sistema con un único ordenador centralizado.

### **2.2.2 Tecnologías del lado del Cliente.**

#### **JavaScript:**

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. [5]

Con JavaScript se puede dar respuesta a eventos iniciados por el usuario, eventos tales como la entrada de una forma o algún enlace. Esto sucede sin ningún tipo de transmisión. De tal forma que cuando un usuario escribe algo en una forma, no es necesario que sea transmitido, verificado y devuelto hacia el servidor. Las entradas son verificadas por la aplicación cliente y pueden ser transmitidas después de esto.

#### **Ventajas:**

- Fácil de aprender, rápido y potente: permite realizar ciertas funciones rápidas en una página Web sólo con crear el código y cargarlo sin necesidad de crear una máquina virtual para compilar su código.
- Es un lenguaje de alto nivel, siendo capaz de aprovechar las propiedades de los exploradores Web, incluso puede realizar algunas acciones sobre el sistema en que se está ejecutando.
- JavaScript al contar con la habilidad de ejecutarse en el cliente y tener tantas funcionalidades, se ha podido ganar la atención de la mayoría de los desarrolladores Web pues ayuda a reducir la carga de trabajo del servidor, ejemplo de esto son las validaciones que posibilita.



### **Cascade Style Sheets (CSS):**

CSS es una tecnología que permite controlar la presentación de los documentos en la Web. Usando CSS se pueden especificar muchos atributos de los elementos que conforman una página Web, como por ejemplo el color del texto, márgenes, el tipo de letras y tamaño, además permite posicionar elementos de forma arbitraria dentro del documento.

### **HTML:**

HTML es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque si le indica como desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. [6]

HTML, no es un lenguaje de programación, es un lenguaje de especificación de contenidos para un tipo específico de documentos.

El HTML es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como lenguajes. Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

### **2.2.3 Tecnologías del Lado del Servidor.**

Entre los lenguajes que trabajan del lado del servidor se pueden citar algunos, que se destacan por ser los más utilizados por los programadores como son: PERL, ASP, PHP, Java, JSP, entre otros. Estos lenguajes desarrollan la lógica de negocio dentro del servidor, además se encargan de los accesos a los distintos Sistemas de Gestión de Bases de Datos. La aplicación Web estará programada en PHP.

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. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

### **PHP (Personal Home Page):**

PHP es una tecnología del lado del servidor, que funciona dentro del código HTML de una página, dándole mayor dinamismo a la misma y con acceso a bases de datos. Es un lenguaje de programación de estilo clásico, lo que quiere decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones. No es un lenguaje de marcas como podría ser HTML, XML o WML.

### **Ventajas:**

- Soporta en cierta medida la orientación a objeto, clases y herencia.
- El análisis léxico para recoger las variables que se pasan en la dirección lo hace PHP de forma automática. Librándose el usuario de tener que separar las variables y sus valores.
- Tiene soporte para diferentes bases de datos: MySQL, PostgreSQL, Oracle,
- Se puede incrustar código PHP con etiquetas HTML.

### **Desventajas:**

- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- Todo el trabajo lo realiza el servidor. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.

### **Active Server Page (ASP).**

ASP, Páginas Activas en el Servidor, es una tecnología creada por Microsoft, destinada a la creación de sitios Web. Añade otra alternativa en sus posibles opciones para el desarrollo de las funcionalidades del lado del servidor. Le permite combinar HTML y código Script en el servidor para crear páginas Web dinámicas y altamente interactivas. Con ASP, todos los Scripts son procesados en el servidor y los resultados son retornados al cliente en formato HTML estándar, reconocible por cualquier navegador.

#### **2.2.4 ¿Por qué PHP?**

Después de hacer el análisis entre el PHP y el ASP, se decide utilizar el PHP debido a que está soportado en la mayoría de las plataformas de Sistemas Operativos, no ocurre lo mismo con ASP por ser propiedad de Microsoft. El lenguaje PHP no tiene costo oculto, o sea que cuando se adquiere incluye un sinnúmero de bibliotecas que proporcionan el soporte para la mayoría de las aplicaciones Web. En el caso de ASP, forma parte del Internet Information Server que viene integrado en Windows NT-2000 Server con un elevado costo de adquisición. PHP y ASP son parecidos en cuanto a la forma de utilización, pero PHP es más rápido, gratuito y multiplataforma.

Uno de los aspectos que se tienen en cuenta a la hora de seleccionar PHP es que es libre, lo que implica menores costos y servidores más baratos que otras alternativas, a la vez que el tiempo entre el hallazgo de un fallo y su resolución es más corto. Además, el volumen de código PHP libre es mucho mayor que en otras tecnologías. Es multiplataforma. Funciona en toda máquina que sea capaz de

compilar su código, entre ellas diversos sistemas operativos. El código escrito en PHP en cualquier plataforma funciona exactamente igual en otra. El acceso a las bases de datos de PHP es muy heterogéneo, pues dispone de un juego de funciones distinto por cada gestor. Es suficientemente versátil y potente como para hacer aplicaciones de gran tamaño que necesiten acceder a recursos a bajo nivel del sistema, como pequeños scripts que envíen por correo electrónico un formulario relleno por el usuario. Puede interactuar con muchos motores de bases de datos tales como MySQL, Oracle, Informix, PostgreSQL, y otros muchos.

### **2.3 UML (Unified Modeling Language).**

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema de software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. [7]

Uno de los objetivos principales de la creación de UML es posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado.

Esta notación ha sido ampliamente aceptada debido al prestigio de sus creadores y debido a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa principalmente en OMT y OOSE.

Se ha escogido UML debido a que es un lenguaje más expresivo, claro y uniforme que los anteriores definidos para el diseño Orientado a Objetos, que no garantiza el éxito de los proyectos pero si mejora sustancialmente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios.

#### **Presenta las siguientes características:**

- Tecnología de orientación a objetos.
- Viabilidad en la corrección de errores.
- Desarrollo incremental e iterativo.
- Participación del cliente en todas las etapas del proyecto.
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).

- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones y otros).
- Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.

### **2.4 Metodologías de Desarrollo de Software.**

El Proceso Unificado de Desarrollo de Software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. RUP es más que un simple proceso, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. [7]

#### **2.4.1 Características de RUP.**

##### **Proceso dirigido por Casos de Uso:**

En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema, también guían su diseño, implementación y prueba. Constituyen un elemento integrador y una guía del trabajo. No sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo. Basándose en los Casos de Uso, se crean los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que efectivamente el producto implemente adecuadamente cada Caso de Uso.

##### **Proceso centrado en la arquitectura:**

La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los Casos de Uso relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML, como son:

las Vista Lógica, Vista de Procesos, Vista de Despliegue, Vista de Implementación y la Vista de Casos de Uso, esta última controlando a las cuatro primeras.

### **Iterativo e Incremental:**

En RUP el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto. Cada mini proyecto se puede ver como una iteración que no es más que un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales del cual se obtiene un incremento que produce un crecimiento en el producto. El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor énfasis en las distintas actividades.

### **2.4.2 Comparación entre RUP, XP y FDD.**

Aunque todos estos procesos de desarrollo comparten muchas características, ya que todos se centran en la producción de software, y que el proceso se implantará para aumentar la calidad del software producido y la eficiencia de los desarrolladores, también existen diferencias significativas. Aquí se tendrán en cuenta sólo algunos puntos para establecer una comparación entre estos procesos, que son considerados uno de los más utilizados para la producción de software. "RUP, XP y FDD tienen pocas similitudes entre sí, aunque XP y FDD poseen algunas más al ser ambos ligeros, orientados al cliente y de iteraciones cortas y rápidas. También se puede decir que debido al carácter general de RUP, algunos autores consideran todos los demás procesos de desarrollo como casos particulares de este". [15]

**Tamaño de los equipos:** RUP está pensado para proyectos y equipos grandes, en cuanto a tamaño y duración, mientras que FDD y XP se implementan mejor para proyectos cortos y equipos más pequeños, siendo quizás FDD más escalable que XP, ya que a mayor tamaño de código y/o equipo, mayor es la necesidad de cierta organización.

**Obtención de requisitos:** RUP y XP describen los requerimientos de la aplicación desde el punto de vista del usuario. Ambos definen los requisitos técnicos sin meterse con detalles de implementación. FDD por el contrario no define explícitamente esa parte del proyecto sobre la adquisición de requisitos, y sólo define el proceder a partir del momento en que ya se han recogido dichos requisitos, de la forma que se quiera, dividiendo los requisitos en las tres primeras fases del proyecto.

**Carga de trabajo:** RUP es un proceso muy basado en la documentación, en la que no son deseables todos los cambios volátiles. Además define en cada momento del ciclo de vida del proyecto, qué artefactos, con qué nivel de detalle, y por qué rol, se deben crear. Se definen que artefactos son necesarios para poder realizar una actividad y que artefactos se deberán crear durante dicha actividad. En el desarrollo de un proyecto con XP es más importante la entrega al cliente del software que necesita, que las funcionalidades que quedan por implementar. FDD es por su parte un proceso intermedio, en el sentido de que genera más documentación que XP; pero menos que RUP que intenta documentar todo.

**Relación con el cliente:** Con RUP se presentarán al cliente los artefactos al final de una fase y se valorarán las precondiciones para la siguiente, y sólo después de que el cliente acepte los artefactos generados se pasará a la siguiente fase. Para asegurar la calidad, en XP y FDD no se basan en formalismos en la documentación, sino en controles propios y una comunicación fluida con el cliente. En ambos casos el cliente recibe después de cada iteración una parte funcional del programa. A través de un ciclo de iteración corto el cliente está informado constantemente sobre la situación del proyecto y puede intervenir rápidamente si el desarrollo se aleja de sus necesidades.

**Desarrollo:** Estos procesos se basan en un proceso iterativo. Esto permite acercarse poco a poco a la solución sin entrar demasiado rápido en detalles, aunque las iteraciones de XP y FDD tienen por lo general una duración menor que en RUP, puesto que la carga a llevar por los programadores a parte del desarrollo del propio software es menor. XP está diseñado con los programadores en mente, facilitando su trabajo, y es por ello que define casi todo el proceso de desarrollo completo, incluido el de pruebas e integración.

RUP y FDD se centran más en la organización global, y muchas de esas actividades, como ejecución de pruebas, las asumen como obligatorias aunque sin definir las completamente, dejando libertad a las distintas subunidades del proyecto para implementarlas a su manera, aunque las directrices de la empresa suelen marcar el camino a seguir.

**Código fuente:** XP es el único que presenta la compartición del código, mayormente debido a que está pensado para equipos pequeños que trabajan conjuntamente y con comunicación constante e inmediata, mientras que RUP y FDD optan por la propiedad del código, aunque definen grupos y sesiones de trabajo conjuntos.

**Evaluación del estado del proyecto:** FDD es posiblemente el proceso más adecuado para definir métricas que definan el estado del proyecto, ya que al dividirlos en unidades pequeñas, es bastante sencillo hacer un seguimiento de las mismas. XP también define esos componentes pequeños; pero la

tarea de informar recae sólo en los jefes de proyecto, mientras que en FDD esta más distribuida en la jerarquía. RUP por su parte, es tan grande y complejo en este sentido como en el resto, por lo que manejar el volumen de información que puede generar, requiere mucho tiempo.

**Algunos puntos débiles:** Ninguno de estos procesos puede ser considerado perfecto, ni ser aplicado en su totalidad, por lo que también es necesario saber donde están sus puntos débiles para corregirlos. XP es un proceso muy orientado a la implementación. Debido al bajo número de documentos a generar, se ofrece al desarrollador un escenario ideal para participar en el proyecto. Este proceso es aceptado con el mejor grado por desarrolladores menos experimentados ya que pueden sacar provecho directo de los compañeros más experimentados. También el cliente se mantiene contento porque recibe un software que se adapta exactamente a sus deseos, mientras disponga de tiempo y dinero. En este proceso la funcionalidad exacta del software final nunca se definió formal y contractualmente, es por eso que este método de desarrollo es quizás más aplicable para desarrollos internos. FDD presenta su talón de Aquiles en la necesidad de tener en el equipo miembros con experiencia que marquen el camino a seguir desde el principio, con la elaboración del modelo global, puesto que no es tan ágil como podría serlo XP. Aunque su punto intermedio entre la libertad de XP y la rigurosidad de RUP, lo hacen sin duda un proceso interesante. El problema de usar RUP está en otro campo completamente distinto. Para el desarrollo de software por medio de equipos pequeños es RUP definitivamente muy grande y prácticamente inalcanzable, esto supone que antes de implantar RUP se debe adaptar hasta el punto de hacerlo parecer otro proceso, lo que también requiere de tiempo y coste.

### 2.4.3 ¿Por qué utilizar RUP como metodología de desarrollo de Software?

Lo primero que se debe tener en cuenta cuando se va a escoger una metodología de desarrollo de software para la realización de un proyecto es definir bien las características que va a tener el mismo. Se decidió escoger RUP porque es un proceso pesado, esta basado en componentes, concebido para proyectos y equipos grandes en cuanto a tamaño y duración. Uno de los requisitos de la aplicación es que debe quedar documentado todo el flujo de información, esta fue otra de las razones por la que se eligió RUP, porque se basa en la documentación, característica que no comparten otras metodologías analizadas como XP y FDD. Además estas metodologías manejan con poco rigor el diseño y el análisis, lo que puede llevar a proyectos eternos que no son viables ni económicamente factibles, ni para el usuario ni para los desarrolladores del sistema.

### 2.5 Herramientas CASE.

#### 2.5.1 Características de las Herramientas CASE.

##### 2.5.1.1 Rational Rose.

Existen varias herramientas creadas para el desarrollo de la ingeniería de software, que tienen el fin de desarrollar programas utilizando técnicas de diseño y metodologías bien definidas, soportadas por herramientas automatizadas. Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas que automatizan una parte del ciclo de desarrollo de software. Ejemplos de herramientas CASE, que se utilizan para el modelado de artefactos son: Umbrello, MagicDraw, Visual Paradigm y Rational Rose.

Rational Rose es una herramienta para el modelado visual, que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida del desarrollo de software. Permite completar una gran parte de las disciplinas de RUP, e incluye un conjunto de herramientas de ingeniería inversa y generación de código que facilitan el camino hasta el producto final. Es una herramienta para traducir requisitos de alto nivel a una arquitectura flexible basada en componentes. Rational Rose se basa en UML para realizar el modelado. [8]

##### 2.5.1.2 Visual Paradigm.

Es una herramienta CASE que utiliza UML como lenguaje de modelado.

##### Características:

- Soporta los últimos estándares de notaciones de JAVA y UML.
- Posee soporte para la generación de código y la ingeniería inversa para Java.
- Es multiplataforma y portable.
- Producto de calidad.
- Soporta aplicaciones web.
- Las imágenes y reportes generados, no son de muy buena calidad.
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.



### 2.5.1.3 MagicDraw.

Posee como características fundamentales la facilitación del análisis y del diseño de los sistemas orientados a objetos y bases de datos, permite a los desarrolladores trabajar en paralelo en el mismo modelo, además entre otras funciones tiene las de generación de código en Java, C++, C#, XML y muchas funciones de ingeniería inversa.

Es una herramienta de modelado que facilita el análisis y diseño de los sistemas orientados a objeto y bases de datos, que utiliza UML y que ha sido implementada totalmente en Java. MagicDraw proporciona una visualización de modelos rápida, eficiente y permite a los desarrolladores trabajar en paralelo en el mismo modelo. Además genera código fuente de un modelo UML y elimina la preparación de documentos tediosos con la generación de informes automáticamente. Incluye también herramientas para ingeniería inversa.

MagicDraw también tiene desventajas, no es software libre y sus requisitos son relativamente altos. Además de que el trabajo en paralelo es limitado.

### 2.5.1.4 Umbrello.

Se caracteriza básicamente por su facilidad de uso, permitir la generación de códigos a partir de él, ayuda en el proceso del desarrollo de software facilitando la creación de un producto de alta calidad, especialmente durante fases de análisis y diseño del proyecto.

- Gran facilidad de uso y cuidado de la interfaz.
- Buen soporte para la generación de código, desde el mismo programa es posible generar todo o parte de este.
- Es de gran ayuda en el proceso de desarrollo de software, principalmente en las fases de análisis y diseño.

### 2.5.1.5 ¿Por qué Rational Rose?

Se decidió utilizar Rational Rose porque es una de las herramientas que esta en la avanzada en el análisis, modelamiento, diseño y construcción orientada a objetos, pues cuenta con las especificaciones y necesidades que los analistas, desarrolladores y arquitectos exigen. Una de las ventajas de Rational Rose es que utiliza la notación estándar en la arquitectura de software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto.

Basado en la investigación realizada sobre algunas de las herramientas CASE más conocidas y utilizadas en el proceso de desarrollo de software, se llegó a la conclusión de utilizar Rational Rose como herramienta de modelado, por ser la más apropiada para el desarrollo de este trabajo debido a su magnitud y porque permite modelar todos los artefactos que realiza el analista y no requiere de condiciones o medios de trabajo específicos. Las demás herramientas CASE presentan limitantes que impiden su uso en el desarrollo de este trabajo: Umbrello no comprende el modelado del negocio, MagicDraw y Visual Paradigm necesitan una gran capacidad de memoria RAM y las computadoras del equipo de desarrollo del proyecto no cuentan con esas características de hardware para su uso.

### **2.6 Diseño de la Interfaz.**

#### **2.6.1 Dreamweaver MX.**

Dreamweaver MX es un software fácil de usar que permite crear páginas Web profesionales. Las funciones de edición visual de Dreamweaver MX permiten agregar rápidamente diseño y funcionalidad a las páginas, sin la necesidad de programar manualmente el código HTML. Se pueden crear tablas, editar marcos, trabajar con capas, insertar comportamientos JavaScript, etc., de una forma muy sencilla y visual. Además incluye un software de cliente FTP completo, permitiendo entre otras cosas trabajar con mapas visuales de los sitios Web. [9]

#### **Características de Dreamweaver MX:**

- Ofrece un único entorno integrado para la creación de sitios Web y aplicaciones de Internet para las principales tecnologías Web.
- Se integra con las demás herramientas de Macromedia como Fireworks y Flash.
- Combina facilidad y potencia en un entorno de desarrollo integrado para los sitios Web PHP, HTML, ASP o JSP. El producto permite un control completo sobre el código y el diseño.
- Macromedia Dreamweaver MX proporciona un entorno accesible y productivo para una variedad de usuarios que va desde los creadores Web hasta los desarrolladores experimentados de aplicaciones Web.
- Dreamweaver permite al usuario utilizar la mayoría de los navegadores Web instalados en su ordenador para previsualizar las páginas Web. Es una herramienta de desarrollo muy cómoda a la hora de trabajar con HTML dinámico.

Se escogió esta herramienta por el potencial del software en cuanto a la capacidad de programar bajo los lenguajes ASP, CSS, PHP, SQL, JSP, y XML, permitiendo la creación de aplicaciones y diseños Web dinámicos y complejos.

### 2.6.2 ¿Qué es Eclipse?

Eclipse es una plataforma universal para integrar herramientas de desarrollo, con una arquitectura abierta y basada en plugins. Además, da soporte a todo tipo de proyectos que abarcan desde el ciclo de vida del desarrollo de aplicaciones, incluyendo soporte para modelado.

**Eclipse es soportado por los principales sistemas operativos:**

- Linux.
- Windows.
- Solaris 8 (SPARC/GTK 2).
- Mac: OSX –Mac/Carbon.

**Beneficios:**

- Es una herramienta open-source.
- Soporta la construcción de una variedad de herramientas para el desarrollo de aplicaciones.
- Soporta el desarrollo de aplicaciones basadas en GUI.
- Soporta herramientas que manipulan diferentes tipos de archivos como por ejemplo Java, C, C++, HTML, GIF.
- Corre en una gran cantidad de sistemas operativos incluyendo Windows y Linux.
- Provee a los desarrolladores, herramientas que facilitan la creación de plugins.

**Desventajas:**

- Si bien Eclipse es multiplataforma, los plugins no tienen que serlo.
- Existen plugins que sólo corren en una plataforma, o que aún no han sido desarrollados para más de una.
- Al ser una herramienta open-source, se desarrollan plugins que no tienen todas las funcionalidades que tienen en otras herramientas comerciales.

### 2.7 Framework Symfony.

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones Web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas Linux, como en plataformas Windows.

#### Características de Symfony:

- Fácil de instalar y configurar en la mayoría de las plataformas, funciona correctamente en los sistemas Windows y Linux.
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de los casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Sigue las mejores prácticas y patrones de diseño para la Web.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.

Se decidió utilizar Symfony como framework de desarrollo debido a las bondades que ofrece en la organización, desarrollo e implementación de las aplicaciones Web. Uno de los factores que incidió en la selección de Symfony es que ofrece un grupo de funcionalidades y helpers que garantizan la integridad y seguridad de los datos.

#### 2.7.1 Patrón Modelo Vista Controlador (MVC)

El Modelo-Vista-Controlador fue introducido inicialmente en la comunidad de desarrolladores de Smalltalk-80. El MVC divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones: modelo vista y controlador.[10]

Este es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El patrón MVC es muy usado en aplicaciones Web.

### **Modelo:**

Es la representación específica del dominio de la información sobre la cual funciona la aplicación. La lógica del dominio añade significado a los datos.

### **El modelo es el responsable de:**

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema).
- Lleva un registro de las vistas y controladores del sistema.

### **Vista:**

Presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.

### **Las vistas son responsables de:**

- Recibir datos del modelo y mostrárselos al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

### **Controlador:**

Responde a eventos usualmente a acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

### **El controlador es responsable de:**

- Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos.

### **Ventajas:**

- Soporte de vistas múltiples. Dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas de una aplicación Web pueden utilizar el mismo modelo de objetos, mostrado de maneras diferentes.
- Adaptación al cambio. Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación, o requerir soporte para nuevos dispositivos como teléfonos celulares. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

### **Desventajas:**

- Complejidad. El patrón introduce nuevos niveles de indirección y por lo tanto aumenta ligeramente la complejidad de la solución. También se profundiza la orientación a eventos del código de la interfaz de usuario, que puede llegar a ser difícil de depurar.
- Costo de actualizaciones frecuentes. Desacoplar el modelo de la vista no significa que los desarrolladores del modelo puedan ignorar la naturaleza de las vistas.

Se decide utilizar el Patrón Arquitectónico Modelo-Vista-Controlador debido a que facilita la programación en diferentes capas de manera paralela e independiente, ventaja que no presenta el Patrón de Tres Capas. Tiene una clara separación entre interfaz, lógica de negocio y de presentación, que además ofrece sencillez para crear distintas representaciones de los datos, presenta facilidades para la realización de pruebas unitarias de los componentes, reutilización de los componentes, simplicidad en el mantenimiento de los sistemas y facilidad para desarrollar prototipos rápidos.

### **2.8 Apache como Servidor Web.**

Un servidor es una computadora que entrega a otras computadoras (los clientes), una información que ellos requieren bajo un lenguaje común, denominado protocolo. Por lo tanto al ver una página Web es porque el servidor les entrega una página HTML vía protocolo HTTP (HyperText Transport Protocol) o protocolo para la transmisión de hipertexto, a través de una conexión TCP/IP por el puerto 80. [13]

Apache es el servidor Web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. [14]

La licencia Apache es una descendiente de la licencias BSD, no es GPL. Esta licencia te permite hacer lo que quieras con el código fuente (incluso forks y productos propietarios) siempre que les reconozcas su trabajo.

### **Ventajas:**

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Apache es una tecnología gratuita de código fuente abierto. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esto le da una transparencia a este software de manera que si se quiere ver que es lo que se está instalando como servidor, se puede saber, sin ningún secreto y sin ningún problema.
- Apache 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.
- Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor.

Se decide utilizar Apache como Servidor Web debido a su alta configurabilidad, robustez y modularidad que posee, además de ser un software gratuito y de código abierto.

## **2.9 Sistema Gestor de Base de Datos.**

### **2.9.1 PostgreSQL.**

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS). Es el sistema de administración de bases de datos relacionales de código abierto que utiliza un subconjunto de instrucciones del lenguaje SQL. Código abierto significa que usted tiene acceso real a números y estadísticas de rendimiento, datos que otras compañías como Oracle no facilitan. Código abierto también significa que usted es libre de modificar PostgreSQL para adaptarlo a sus necesidades particulares.

### Características:

- PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia y arreglos.
- Instalación ilimitada: Es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. Algunos proveedores comerciales consideran a esto la principal fuente de incumplimiento de licencia. Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.
- Altamente Extensible: PostgreSQL soporta operadores funcionales, métodos de acceso y tipos de datos definidos por el usuario.
- Integridad Referencial: PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- Diseñado para ambientes de alto volumen: PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC (Acceso concurrente multiversión) para conseguir una mejor respuesta en ambientes de grandes volúmenes.
- Cliente-Servidor: PostgreSQL usa una arquitectura cliente-servidor. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.
- Protección por medio de usuarios y grupos de usuarios asignando privilegios sobre los objetos de un cluster de bases de datos.
- Asignación de privilegios para los diferentes objetos (base de datos, tabla, vista, función, esquema).
- Expresiones condicionales dentro de sentencias SELECT.



### 2.9.2 MySQL.

En el caso del software MySQL es un sistema para la administración de bases de datos relacional (RDBMS) rápido y sólido. El servidor de MySQL controla el acceso a los datos para garantizar el uso simultáneo de varios usuarios, proporcionar acceso a dichos datos y asegurarse de que sólo obtienen acceso a ellos los usuarios con autorización. Proporciona un servidor de base de datos SQL (Structured Query Language) muy rápido, multiusuario y robusto, que está diseñado para entornos de producción críticos, con alta carga de trabajo, así como para integrarse en software para ser distribuido. Además trabaja en entornos cliente-servidor con diferentes programas y bibliotecas cliente, herramientas administrativas y un amplio abanico de interfaces de programación para aplicaciones, pudiendo utilizarse en una gran cantidad de sistemas Unix diferentes así como bajo Microsoft Windows.

### 2.9.3 ¿Por qué utilizar PostgreSQL?

Se decidió utilizar PostgreSQL debido a que está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo y soporta la arquitectura cliente-servidor, que se utilizará en la aplicación. Además está diseñado para proyectos de gran volumen y cuenta con varias funcionalidades que garantizan la integridad y seguridad de los datos, aspecto este de suma importancia para la construcción de cualquier software.

### 2.10 Conclusiones.

Las aplicaciones Web se han hecho indispensables para la correcta gestión de la información de cualquier entidad u organización. En este capítulo se abordaron las características principales de las tecnologías Web que va a soportar la aplicación, se abordan características sobre las tecnologías del lado del cliente y del lado del servidor como PHP y JavaScript. Se hace referencia al Lenguaje Unificado de Modelado, se explican sus características y ventajas. Se explican las características fundamentales de Symfony, plataforma que se utiliza para un mejor desarrollo de la aplicación Web. Se explica porque escoger el proceso unificado de desarrollo (RUP) como metodología de desarrollo de software. Se describe PostgreSQL como sistema gestor de base de datos a utilizar, además se describen una serie de herramientas que ayudan en el desarrollo de la aplicación como: Dreamweaver y el Rational Rose.

### **CAPÍTULO 3: “Presentación de la solución propuesta”.**

#### **3.1 Introducción.**

Para darle solución a la situación problemática se decide realizar un modelo de dominio debido a que no existen procesos bien definidos, y los flujos de información que se suceden en los Departamentos Docentes son difusos, por lo que fue necesario encontrar los conceptos, personas y eventos principales del entorno. En este capítulo se define el entorno organizacional, se realiza una breve explicación de cuando es necesario aplicar un modelo de dominio y sus formas típicas de representación. Además se identifican los principales eventos y conceptos relacionados con el dominio del problema. Se realiza la representación del diagrama de clases del dominio, se hace una descripción del sistema identificando los actores y los requerimientos funcionales y no funcionales de la aplicación.

#### **3.2 Entorno donde trabajará el sistema Informático.**

El entorno donde se pondrá en funcionamiento la aplicación Web, está comprendido en los Departamentos Docentes de la Facultad 9, debido a que son estos los encargados de procesar toda la información concerniente a los profesores, los mismos necesitan de esta aplicación Web puesto que existen problemas con la gestión de la información que se manejan en estas instituciones.

##### **3.2.1 ¿Cuándo aplicar un modelo de Dominio?**

- Dificil establecimiento de reglas de funcionamiento.
- Imposibilidad de determinar subsistemas (exceso de interconexiones).
- Los flujos de información son difusos (múltiples orígenes, sólo eventos, sucesos).
- Solapamiento de responsabilidades.
- Múltiples responsabilidades.

##### **3.2.2 Formas típicas de las clases del dominio.**

- Objetos del entorno que representan cosas que se manipulan.
- Objetos del mundo real y conceptos de los que el futuro sistema debe de hacer seguimiento.
- Sucesos que ocurrirán o han ocurrido.

### 3.3 ¿Qué es Modelo de Dominio?

Un modelo de dominio describe los conceptos importantes del contexto como objetos del dominio, y enlaza estos objetos unos con otros, además captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las cosas que existen o eventos que suceden en el entorno en el que trabaja el sistema.

#### 3.3.1 Conceptos principales del entorno.

**Facultad:** Es la entidad docente encargada de organizar y controlar a los estudiantes y profesores de una Universidad.

**Departamento Docente:** Es la entidad organizativa que controla todo lo referente a los profesores que laboran en la facultad, las disciplinas y las asignaturas.

**Jefe de Departamento:** Es la máxima autoridad que existe en un Departamento Docente, controla y gestiona los recursos del departamento.

**Jefe Disciplina:** Es la persona que controla una disciplina docente, vela por el correcto funcionamiento de la disciplina y las asignaturas correspondientes a la misma.

**Jefe de Asignatura:** Es el máximo responsable de una asignatura, es el encargado de controlar y tutorear a los profesores que imparten esa asignatura.

**Profesor:** Es la persona que imparte una asignatura determinada y pertenece a un Departamento Docente.

#### 3.3.2 Eventos principales del entorno.

**Plan de Trabajo:** Documento que recoge una serie de actividades, sus responsables, el tiempo de cumplimiento y la puntuación que tiene la realización de cada actividad.

**Programa Analítico:** Es el documento en el cual están reflejados todos los objetivos de las asignaturas del semestre, así como la distribución de las mismas en conferencias, clases prácticas, seminarios, talleres, laboratorios y otras actividades docentes.

**Reportes:** Informe detallado que genera el Jefe de Departamento sobre la realización de los planes de trabajos y las actividades del departamento.

3.3.3 Diagrama de clases del Modelo de Dominio.

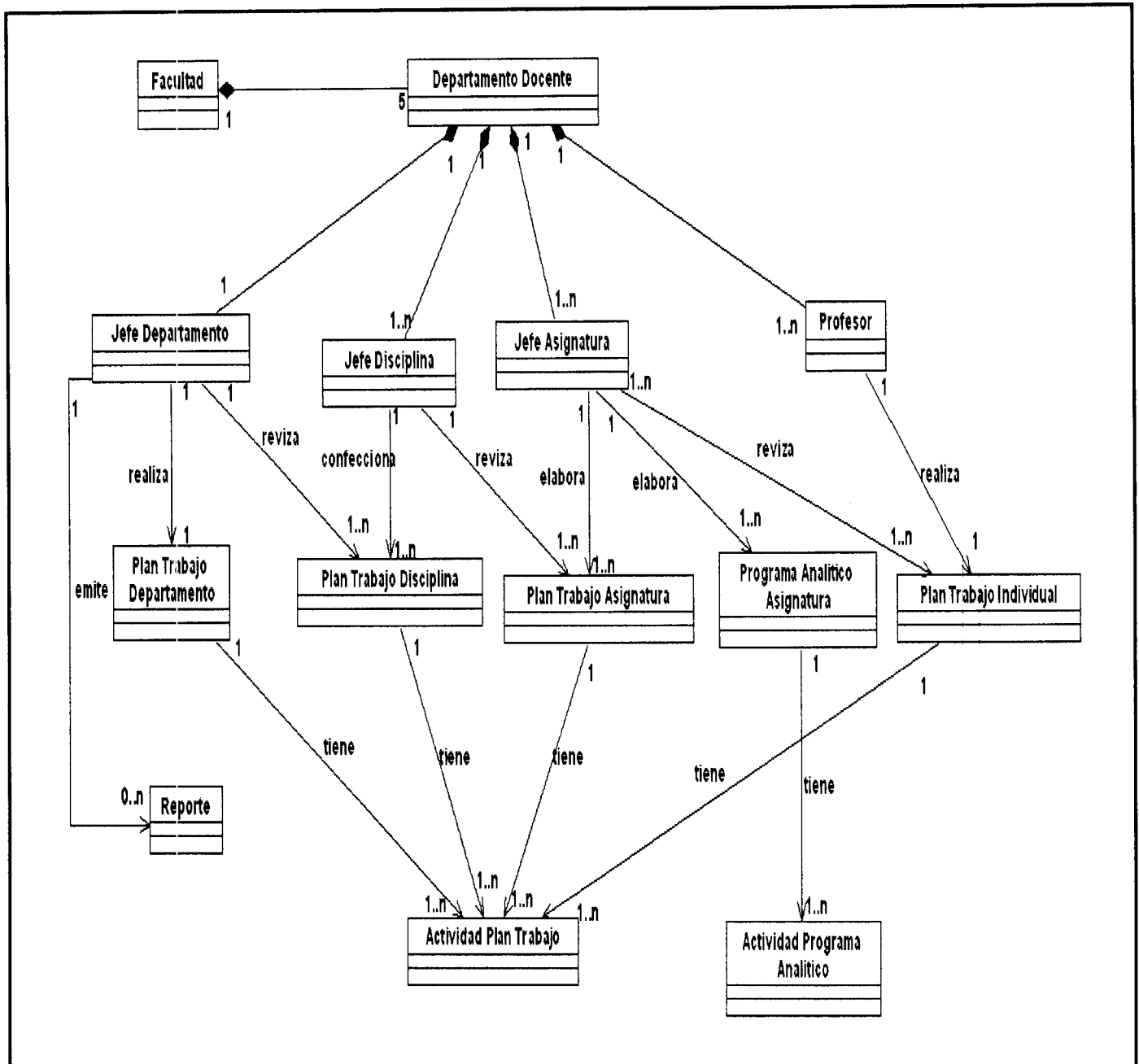


Figura # 1: Diagrama de Clases del Dominio.

### 3.3.4 Glosario de Términos del Dominio.

**Disciplina Docente:** Conjunto de asignaturas de la misma rama de estudio.

**Plan de Adiestramiento:** Documento que recoge una serie de actividades para el mejor desempeño de los profesores.

**Asignatura:** Materia que imparte un profesor o maestro a sus alumnos y que forma parte del plan de estudios.

**Evaluación Docente:** Documento que recoge la evaluación de los profesores de la facultad en un intervalo de tiempo determinado.

### 3.4 Requerimientos.

Un Requisito es la condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Es la condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer las necesidades del cliente.

#### 3.4.1 Requerimientos Funcionales.

Son capacidades o condiciones que el sistema debe cumplir. Definen que debe hacer el software. Se mantienen invariables sin importar con que propiedades o cualidades se relacionen.

1. Gestionar Plan Trabajo Departamento.
  - 1.1 - Crear Plan Trabajo Departamento.
  - 1.2 - Adicionar Actividades Plan Trabajo Departamento.
  - 1.3 - Eliminar Actividades Plan Trabajo Departamento.
  - 1.4 - Modificar Actividades Plan Trabajo Departamento.
2. Modificar Evaluación.
  - 2.1 - Modificar Evaluación Jefe Disciplina.
  - 2.2 - Modificar Evaluación Jefe Asignatura.
  - 2.3 - Modificar Evaluación Profesor.
3. Evaluar Actividades Plan Trabajo Disciplina.
4. Aprobar Plan Trabajo Disciplina.
5. Evaluar Jefe Disciplina.

6. Generar % Cumplimiento Planes Trabajo.
  - 6.1 - Generar % Cumplimiento Plan de Trabajo Departamento.
  - 6.2 - Generar % Cumplimiento Plan de Trabajo Disciplina.
  - 6.3 - Generar % Cumplimiento Plan de Trabajo Asignatura.
  - 6.4 - Generar % Cumplimiento Plan de Trabajo Individual.
7. Gestionar Plan de Trabajo Disciplina.
  - 7.1 - Crear Plan de Trabajo Disciplina.
  - 7.2 - Eliminar Actividades Plan de Trabajo Disciplina.
  - 7.3 - Modificar Actividades Plan de Trabajo Disciplina.
  - 7.4 - Adicionar Actividades Plan de Trabajo Disciplina.
  - 7.5 - Mostrar Plan Trabajo Departamento.
8. Evaluar Jefe Asignatura.
9. Evaluar Actividades Plan Trabajo Asignatura.
10. Aprobar Plan Trabajo Asignatura.
11. Gestionar Plan Trabajo Asignatura.
  - 11.1 - Crear Plan Trabajo Asignatura.
  - 11.2 - Eliminar Actividades Plan Trabajo Asignatura.
  - 11.3 - Modificar Actividades Plan Trabajo Asignatura.
  - 11.4 - Adicionar Actividades Plan Trabajo Asignatura.
  - 11.5 - Mostrar Plan Trabajo Disciplina.
12. Evaluar Profesor.
13. Evaluar Actividades Plan Trabajo Individual.
14. Gestionar Plan de Trabajo Individual.
  - 14.1 - Crear Plan Trabajo Individual.
  - 14.2 - Eliminar Actividades Plan Trabajo Individual.
  - 14.3 - Modificar Actividades Plan Trabajo Individual.

- 14.4 - Adicionar Actividades Plan Trabajo Individual.
- 14.5 - Mostrar Plan Trabajo Asignatura.
- 15. Gestionar Programa Analítico.
  - 15.1 - Crear Programa Analítico.
  - 15.2 - Eliminar Actividades Programa Analítico.
  - 15.3 - Modificar Actividades Programa Analítico.
  - 15.4 - Adicionar Actividades Programa Analítico.
- 16. Aprobar Plan Trabajo Individual.

### **3.4.2 Requerimientos No Funcionales.**

Son las propiedades o cualidades que el producto debe tener. Son fundamentales en el éxito del producto y normalmente están vinculados a los requerimientos funcionales.

#### **Usabilidad:**

- El sistema podrá ser utilizado sólo por las personas autorizadas, por el administrador.
- Facilidad de uso.
- Además garantizará una conexión rápida y segura con la base de datos que contendrá la información, lo que permitirá facilidades de actualización y mantenimiento desde cualquier lugar.

#### **Rendimiento:**

- La eficiencia del producto estará determinada por la velocidad de transmisión de la red y su nivel de congestión, además de la capacidad de procesamiento del servidor de datos.
- El sistema debe ser capaz de brindar la información necesaria y mostrar la información con el criterio de búsqueda seleccionado por el usuario.

#### **Seguridad:**

- Las personas que utilizarán la aplicación accederán, según el rol que desempeñan, para garantizar que la información sea vista y actualizada solamente por quien tiene derecho ella.
- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos, los profesores que no están registrados en el sistema no podrán acceder al mismo.

### **Disponibilidad:**

- El sistema estará disponible las 24 horas del día.

### **Soporte:**

- El sistema permitirá su extensibilidad, pudiéndose agregar nuevas funcionalidades o modificar las existentes para lograr mejores prestaciones en el momento que se requieran realizar cambios.
- La información deberá estar disponible a los usuarios en todo momento, limitada solamente por las restricciones que estos tengan de acuerdo a la política de seguridad del sistema.

### **Software:**

- Navegador compatible o superior con Internet Explorer 4.
- PostgreSQL 8.2.x
- Apache 2.0.x
- PHP 5.0.x
- Framework Symfony.

### **Hardware:**

El Servidor de Aplicaciones y el Servidor de Base de Datos deben poseer:

- Una memoria RAM de 512 MB o superior.
- Ambas máquinas deben ser Pentium IV o superior.

## **3.5 Descripción del Sistema Propuesto.**

### **3.5.1 Descripción de los actores.**



Tabla # 1. Descripción de los actores del sistema.

Actores del Sistema	Justificación
<b>Jefe Departamento</b>	Es la máxima autoridad en un Departamento Docente. Realiza la operación de Gestionar el Plan de Trabajo del Departamento, es el encargado de evaluar a los Jefes de Disciplina de su departamento, evalúa las actividades de los Planes de Trabajo de las Disciplinas correspondientes a su departamento, además puede generar el porcentaje de cumplimiento de cualquier plan de trabajo del Departamento. Aprueba los Planes de Trabajos de las Disciplinas correspondientes a su Departamento.
<b>Jefe Disciplina</b>	Realiza la operación de Gestionar el Plan de Trabajo de la Disciplina, es el encargado de evaluar a los Jefes Asignaturas y de evaluar las actividades de los Planes de Trabajo de las Asignaturas. Aprueba los Planes de Trabajo de las asignaturas que pertenecen a su Disciplina.
<b>Jefe Asignatura</b>	Realiza la operación de Gestionar el Plan de Trabajo de la Asignatura, es el encargado de evaluar a los Profesores que imparten su Asignatura y evalúa las actividades de los Planes de Trabajo Individual de los Profesores de su Asignatura. Además se encarga de Gestionar el Programa Analítico de la Asignatura.
<b>Profesor</b>	Se encarga Gestionar su Plan de Trabajo Individual.

### **3.5.2 Casos de Uso del Sistema.**

Los Casos de Uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario.

El sistema esta dividido en cuatro subsistemas para poder hacer más legible y comprensible el modelado del sistema, definidos por actores, los cuales ejecutan una serie de funcionalidades fundamentales para el funcionamiento óptimo de la aplicación. Ellos son: Subsistema Jefe Departamento, Subsistema Jefe Disciplina, Subsistema Jefe Asignatura y el Subsistema Profesor Regular.

3.5.2.1 Subsistema Jefe Departamento.

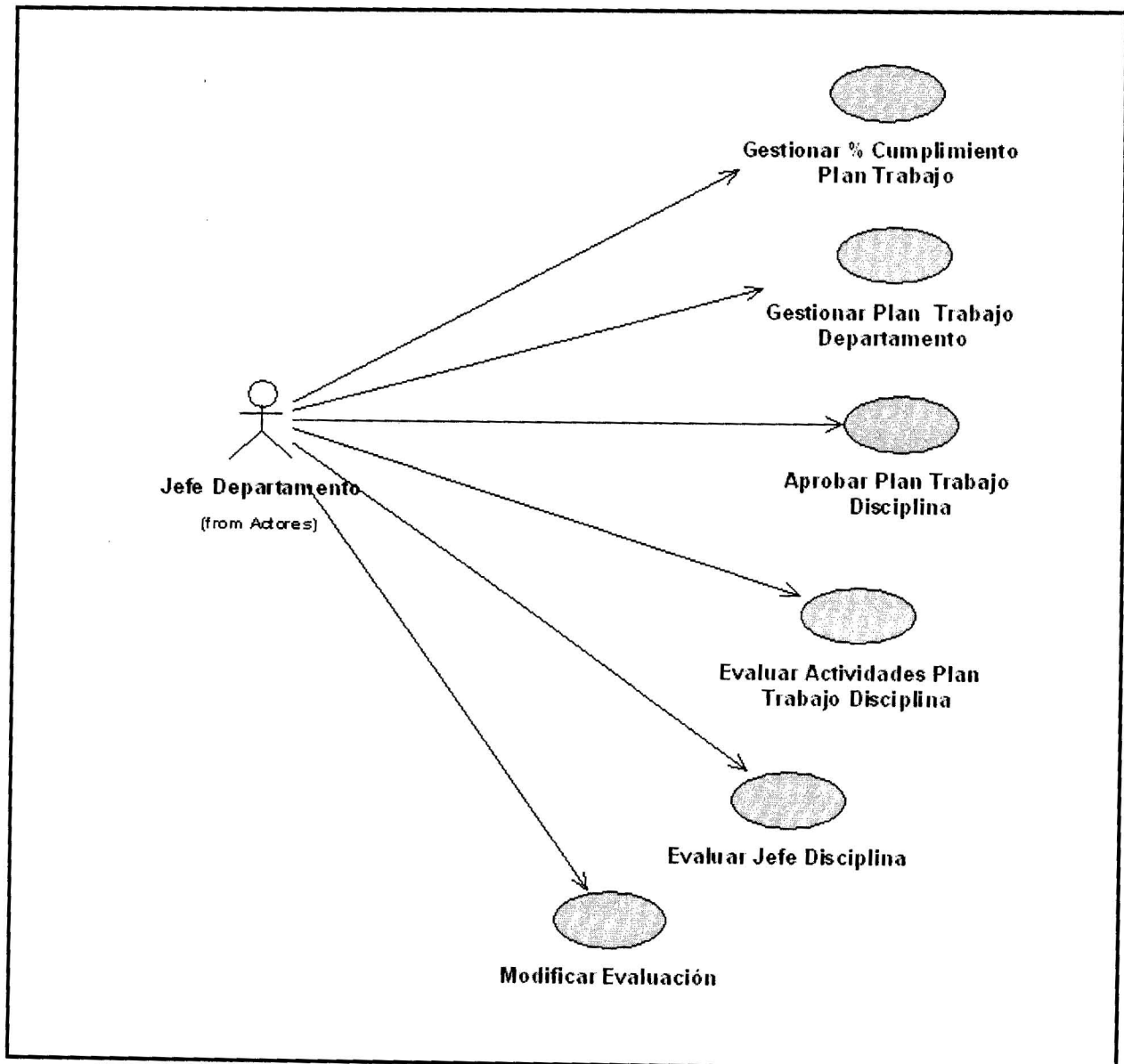


Figura # 2: Diagrama de Caso de Uso Subsistema Jefe Departamento.

Tabla # 2. CUS Gestionar Plan Trabajo Departamento.

<b>Caso de Uso:</b>		<b>Gestionar Plan Trabajo Departamento.</b>
<b>Actor:</b>	Jefe Departamento (inicia)	
<b>Propósito:</b>	Permite al Jefe del Departamento gestionar (Modificar, Eliminar y Adicionar Actividades al PT Departamento, además de crear un plan de Trabajo del Departamento).	
<b>Resumen:</b>	El CUS se inicia cuando el Jefe del Departamento selecciona la opción de Gestionar Plan Trabajo Departamento, luego selecciona el tipo de gestión, introduce los datos necesarios, el sistema ejecuta la acción seleccionada por el Jefe del Departamento y termina el CUS.	
<b>Referencias:</b>	RF 1, RF 1.1, RF 1.2, RF1.3, RF1.4	
<b>Precondiciones:</b>	El Jefe del Departamento debe pertenecer al departamento.	
<b>Poscondiciones:</b>	<ol style="list-style-type: none"> <li>1. Información de las Actividades del Plan de Trabajo del Departamento adicionada a la Base de Datos.</li> <li>2. Información de las Actividades del Plan de Trabajo del Departamento modificadas a la Base de Datos.</li> <li>3. Información de las Actividades del Plan de Trabajo del departamento eliminadas de la Base de Datos.</li> <li>4. Información del Plan de Trabajo del Departamento creado en la Base de Datos.</li> </ol>	
<b>Curso Normal de Eventos:</b>		
<b>Acción del Actor:</b>	<b>Respuesta del Sistema:</b>	
1. El Jefe Departamento selecciona la opción de Gestionar Plan Trabajo Departamento.	1.1- El sistema verifica que sea el Jefe de Departamento.  1.2- El sistema muestra las opciones: Crear Plan Trabajo Departamento, Adicionar Actividades Plan Trabajo Departamento, Modificar Actividades Plan Trabajo Departamento y Eliminar Actividades Plan Trabajo Departamento.	

<b>Escenario 1: Crear Plan Trabajo Departamento.</b>	
1. El Jefe Departamento selecciona la opción de Crear Plan Trabajo Departamento.	1.1 - El sistema muestra el formulario a completar para la creación del nuevo Plan de Trabajo del Departamento.
2. El Jefe Departamento introduce los datos solicitados por el sistema.	2.1- El sistema verifica los datos introducidos por el Jefe Departamento.  2.2- Si los datos introducidos son correctos el sistema adiciona dicha información en la Base de Datos correspondiente y termina el CUS.
<b>Curso alternativo de los eventos:</b>	
<b>Acción 2.2:</b>	Si los datos introducidos por el Jefe Departamento son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al usuario retornar a la acción 2.
<b>Escenario 2: Adicionar Actividades Plan Trabajo Departamento.</b>	
1. El Jefe Departamento selecciona la opción de Adicionar Actividades Plan Trabajo Departamento.	1.1- El sistema muestra el formulario a completar para adicionar actividades al Plan de Trabajo del Departamento.
2. El Jefe Departamento introduce los datos solicitados por el sistema.	2.1- El sistema verifica los datos introducidos por el Jefe Departamento.  2.2 - Si los datos introducidos son correctos el sistema adiciona dicha información en la Base de Datos correspondiente y termina el CUS.
<b>Curso alternativo de los eventos:</b>	
<b>Acción 2.1:</b>	Si los datos introducidos por el Jefe Departamento son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al usuario retornar a la acción 2.

<b>Escenario 3: Modificar Actividades Plan Trabajo Departamento.</b>	
<p>1. El Jefe Departamento selecciona la opción de Modificar Actividades Plan Trabajo Departamento.</p> <p>2. El Jefe Departamento realiza las modificaciones a las Actividades del Plan Trabajo Departamento.</p>	<p>1.1- El sistema muestra el Plan de Trabajo del Departamento con sus Actividades al cual el Jefe Departamento tiene acceso.</p> <p>2.1- El sistema verifica los datos modificados por el Jefe Departamento.</p> <p>2.2- Si los datos modificados son correctos el sistema adiciona dicha información en la Base de Datos correspondiente y termina el CUS.</p>
<b>Curso alterno de los eventos:</b>	
<b>Acción 2.1:</b>	Si los datos modificados por el Jefe Departamento son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al usuario retornar a la acción 2.
<b>Escenario 4: Eliminar Actividades Plan Trabajo Departamento.</b>	
<p>1. El Jefe Departamento selecciona la opción de Eliminar Actividades Plan Trabajo Departamento.</p> <p>2. El Jefe Departamento selecciona la actividad que desea eliminar.</p> <p>3. El Jefe Departamento confirma si quiere o no eliminar la actividad seleccionada.</p>	<p>1.1- El sistema muestra las actividades del Plan de Trabajo del Departamento a las cuales el Jefe Departamento tiene acceso.</p> <p>2.1- El sistema muestra un mensaje de advertencia para la acción a realizar.</p> <p>3.1- Si el Jefe Departamento acepta el sistema elimina la actividad del Plan de Trabajo del Departamento y culmina el CUS.</p>
<b>Curso alterno de los eventos:</b>	

<b>Acción 3.1:</b>	Si el Jefe Departamento cancela la acción se culmina el CUS sin ejecutar ninguna acción.
<b>Prioridad:</b>	Crítico.

Tabla # 3 CUS Modificar Evaluación.

<b>Caso de Uso:</b>		<b>Modificar Evaluación.</b>
<b>Actor:</b>	Jefe Departamento (inicia)	
<b>Propósito:</b>	Permite al Jefe del Departamento modificar la evaluación de cualquier Profesor que pertenezca a su Departamento.	
<b>Resumen:</b>	El CUS se inicia cuando el Jefe del Departamento selecciona la opción Modificar Evaluación, luego selecciona el Profesor al cual le va a modificar la evaluación modificando su evaluación, el sistema realiza la acción ejecutada por el Jefe del Departamento y termina el CUS.	
<b>Referencias:</b>	RF 2, RF 2.1, RF 2.2, RF2.3.	
<b>Precondiciones:</b>	El Jefe del Departamento debe pertenecer al departamento. El Profesor ya debe estar evaluado.	
<b>Poscondiciones:</b>		
<b>Curso Normal de Eventos:</b>		
<b>Acción del Actor:</b>	<b>Respuesta del Sistema:</b>	
1. El Jefe Departamento selecciona la opción de Modificar Evaluación.	1.1- El sistema verifica que sea el Jefe de Departamento. 1.2- El sistema muestra un listado con los profesores que pertenecen al Departamento.	
2. El Jefe Departamento selecciona el Profesor al cual le va a modificar la evaluación.	2.1- El sistema verifica que el Profesor pertenezca al Departamento Docente. 2.2- El sistema muestra al Profesor con sus	

3. El Jefe Departamento modifica la evaluación.	evaluaciones. 3.1- El sistema actualiza la evaluación del profesor finalizando el CUS.
<b>Curso alterno de los eventos:</b>	
<b>Acción 1.1:</b>	Si no es Jefe Departamento el sistema le muestra un mensaje de error al usuario indicándole que sólo el Jefe Departamento tiene acceso a esa página.
<b>Prioridad:</b>	Crítico.

Tabla # 4. CUS Evaluar Actividades Plan Trabajo Disciplina.

<b>Caso de Uso:</b>		<b>Evaluar Actividades Plan Trabajo Disciplina.</b>
<b>Actor:</b>	Jefe Departamento (inicia)	
<b>Propósito:</b>	Permite al Jefe del Departamento evaluar las actividades de los Planes de Trabajo de las Disciplinas.	
<b>Resumen:</b>	El CUS se inicia cuando el Jefe del Departamento selecciona la opción Evaluar Actividades Plan Trabajo de la Disciplina, el sistema le muestra los Planes de Trabajo de las Disciplinas con sus respectivas actividades, luego el Jefe Departamento evalúa cada actividad y el sistema adiciona cada evaluación en las actividades del Plan de Trabajo de la Disciplina y termina el CUS.	
<b>Referencias:</b>	RF 3.	
<b>Precondiciones:</b>	El Plan de Trabajo de la Disciplina debe estar creado.	
<b>Poscondiciones:</b>		
<b>Curso Normal de Eventos:</b>		
<b>Acción del Actor:</b>	<b>Respuesta del Sistema:</b>	
1. El Jefe Departamento selecciona la opción de Evaluar Actividades Plan Trabajo Disciplina.	1.1- El sistema verifica que sea el Jefe de Departamento.  1.2- El sistema muestra todos los Planes de	



<p>2. El Jefe Departamento selecciona el Plan de Trabajo de la Disciplina.</p> <p>3. El Jefe Departamento evalúa las Actividades.</p>	<p>Trabajo de Disciplina que pertenecen al departamento del Jefe Departamento.</p> <p>2.1- El sistema verifica que el Plan de Trabajo de la Disciplina seleccionado cuente con al menos una actividad no evaluada.</p> <p>2.2- El sistema muestra el Plan de Trabajo de la Disciplina con sus actividades.</p> <p>3.1- El sistema Adiciona las evaluaciones finalizando el CUS.</p>
<b>Curso alterno de los eventos:</b>	
<b>Acción 2.1:</b>	Si el Plan de Trabajo de la Disciplina seleccionado no tiene actividades sin evaluar el sistema muestra un mensaje de error indicando que el Plan de Trabajo no presenta actividades para evaluar y le indica al usuario retomar la acción 2.
<b>Prioridad:</b>	<b>Crítico.</b>

**Tabla # 5. CUS Aprobar Plan Trabajo Disciplina.**

<b>Caso de Uso:</b>		<b>Aprobar Plan Trabajo Disciplina.</b>
<b>Actor:</b>	Jefe Departamento (inicia)	
<b>Propósito:</b>	Permite al Jefe del Departamento aprobar los Planes Trabajo de la Disciplina que pertenezca a su Departamento.	
<b>Resumen:</b>	El CUS se inicia cuando el Jefe del Departamento selecciona la opción Aprobar Plan Trabajo Disciplina, el sistema le pide al Jefe Departamento que seleccione el Plan de Trabajo de la Disciplina que desea aprobar, el Jefe Departamento selecciona el Plan de Trabajo de la Disciplina, el sistema le muestra el Plan de Trabajo de la Disciplina, luego el Jefe Departamento aprueba el Plan de Trabajo de la Disciplina y el sistema actualiza el Plan de Trabajo de la Disciplina y termina el CUS.	
<b>Referencias:</b>	RF 4.	

<b>Precondiciones:</b>	El plan Trabajo de la Disciplina seleccionado debe estar creado.
<b>Poscondiciones:</b>	
<b>Curso Normal de Eventos:</b>	
<b>Acción del Actor:</b>	<b>Respuesta del Sistema:</b>
<p>1. El Jefe Departamento selecciona la opción Aprobar Plan Trabajo Disciplina.</p> <p>2. El Jefe Departamento selecciona el Plan de Trabajo de la Disciplina.</p> <p>3. El Jefe Departamento aprueba el Plan de Trabajo de la Disciplina.</p>	<p>1.1- El sistema verifica que sea el Jefe de Departamento.</p> <p>1.2- El sistema le pide al Jefe Departamento que seleccione el Plan de Trabajo de la Disciplina que desea aprobar.</p> <p>2.1- El sistema verifica que el Plan de Trabajo de de la Disciplina seleccionado no esté aprobado.</p> <p>2.2- El sistema muestra el Plan de Trabajo de la Disciplina con sus actividades.</p> <p>3.1- El sistema actualiza el Plan de Trabajo de la Disciplina finalizando el CUS.</p>
<b>Curso alterno de los eventos:</b>	
<b>Acción 2.1:</b>	Si el Plan de Trabajo de la Disciplina seleccionado ya se encuentra aprobado el sistema muestra un mensaje de error indicando que el Plan de Trabajo de la Disciplina ya esta aprobado y le indica al usuario retomar la acción 2.
<b>Prioridad:</b>	Crítico.

Tabla # 6. CUS Evaluar Jefe Disciplina.

<b>Caso de Uso:</b>		<b>Evaluar Jefe Disciplina.</b>
<b>Actor:</b>	Jefe Departamento (inicia)	
<b>Propósito:</b>	Permite al Jefe del Departamento evaluar a los Jefes de Disciplina que pertenezcan a su departamento.	
<b>Resumen:</b>	El CUS se inicia cuando el Jefe del Departamento selecciona la opción Evaluar Jefe Disciplina, el sistema le pide que seleccione el usuario del	

	profesor que desea evaluar, el Jefe Departamento selecciona el usuario, el sistema verifica el porcentaje de cumplimiento de sus actividades y le asigna una evaluación y termina el CUS.	
<b>Referencias:</b>	RF 5.	
<b>Precondiciones:</b>	El Plan Trabajo de la Disciplina que debe realizar el Jefe de Disciplina debe estar creado y que al menos tenga una actividad que esté evaluada.	
<b>Poscondiciones:</b>		
<b>Curso Normal de Eventos:</b>		
<b>Acción del Actor:</b>	<b>Respuesta del Sistema:</b>	
<p>1. El Jefe Departamento selecciona la opción Evaluar Jefe Disciplina.</p> <p>2. El Jefe Departamento selecciona el Jefe Disciplina con el intervalo de fecha que desea evaluar.</p>	<p>1.1- El sistema verifica que sea el Jefe de Departamento.</p> <p>1.2- El sistema le muestra todos los Jefes de Disciplina que pertenezcan a ese Departamento y el intervalo de fecha que desea evaluar.</p> <p>2.1- El sistema verifica los datos introducidos por el Jefe Departamento.</p> <p>2.2- El sistema verifica el porcentaje del cumplimiento de las actividades del Plan de Trabajo de la Disciplina correspondiente al Jefe de Disciplina que escogió.</p> <p>2.3-El sistema le asigna una evaluación al Jefe de Disciplina finalizando el CUS.</p>	
<b>Curso alterno de los eventos:</b>		
<b>Acción 2.1:</b>	Si los datos de selección del Jefe Departamento son incorrectos se muestra un mensaje de error y retorna a la acción 2.	
<b>Prioridad:</b>	Crítico.	

Tabla # 7. CUS Generar % Cumplimiento Planes Trabajo.

Caso de Uso:		Generar % Cumplimiento Planes Trabajo.
<b>Actor:</b>	Jefe Departamento (inicia)	
<b>Propósito:</b>	Permite al Jefe del Departamento generar un porcentaje del cumplimiento de las actividades de cualquier Plan de Trabajo que pertenezca a su Departamento.	
<b>Resumen:</b>	El CUS se inicia cuando el Jefe de Departamento selecciona la opción Generar % del Cumplimiento Plan de Trabajo, el sistema le muestra dos criterios para generar el porcentaje de cumplimiento, el Jefe del Departamento selecciona el criterio deseado y el sistema muestra el porcentaje de cumplimiento de las actividades de del Plan de Trabajo correspondiente al criterio que escogió y termina el CUS.	
<b>Referencias:</b>	RF 6, RF 6.1, RF 6.2, RF 6.3, RF 6.4	
<b>Precondiciones:</b>		
<b>Poscondiciones:</b>		
<b>Curso Normal de Eventos:</b>		
<b>Acción del Actor:</b>	<b>Respuesta del Sistema:</b>	
1. El Jefe Departamento selecciona la opción Generar % Cumplimiento Plan Trabajo.	1.1- El sistema verifica que sea el Jefe de Departamento.  1.2- El sistema muestra las opciones: Generar porcentaje de cumplimiento por usuario y Generar porcentaje de cumplimiento por el tipo de plan de trabajo.	

<b>Escenario 1: Generar porciento de cumplimiento por usuario.</b>	
1. El Jefe Departamento selecciona el usuario del profesor al cual desea gestionarle el %Cumplimiento del plan de trabajo.	1.1- El sistema verifica el usuario seleccionado por el Jefe de Departamento.  2.1- El sistema muestra el reporte con el nombre del Plan de Trabajo, su responsable y el porciento de cumplimiento finalizan el CUS.
<b>Curso alterno de los eventos:</b>	
<b>Acción 1.1:</b>	Si el Jefe Departamento no seleccionó ningún usuario, se muestra un mensaje de error indicando que debe seleccionar el usuario del profesor enviándolo para la acción 1.
<b>Escenario 2: Generar porciento de cumplimiento por el criterio de búsqueda.</b>	
1. El Jefe Departamento selecciona el criterio de búsqueda para el cual desea gestionarle el %Cumplimiento del plan de trabajo.	1.1- El sistema verifica el criterio seleccionado por el Jefe de Departamento.  2.1- El sistema muestra el reporte con el tipo de plan de trabajo seleccionado y el porciento de cumplimiento finalizando el CUS.
<b>Curso alterno de los eventos:</b>	
<b>Acción 1.1:</b>	Si el Jefe Departamento no seleccionó ningún criterio, se muestra un mensaje de error indicando que debe seleccionar un criterio, enviándolo para la acción 1.
<b>Prioridad:</b>	Crítico.

3.5.2.2 Subsistema Jefe Disciplina.

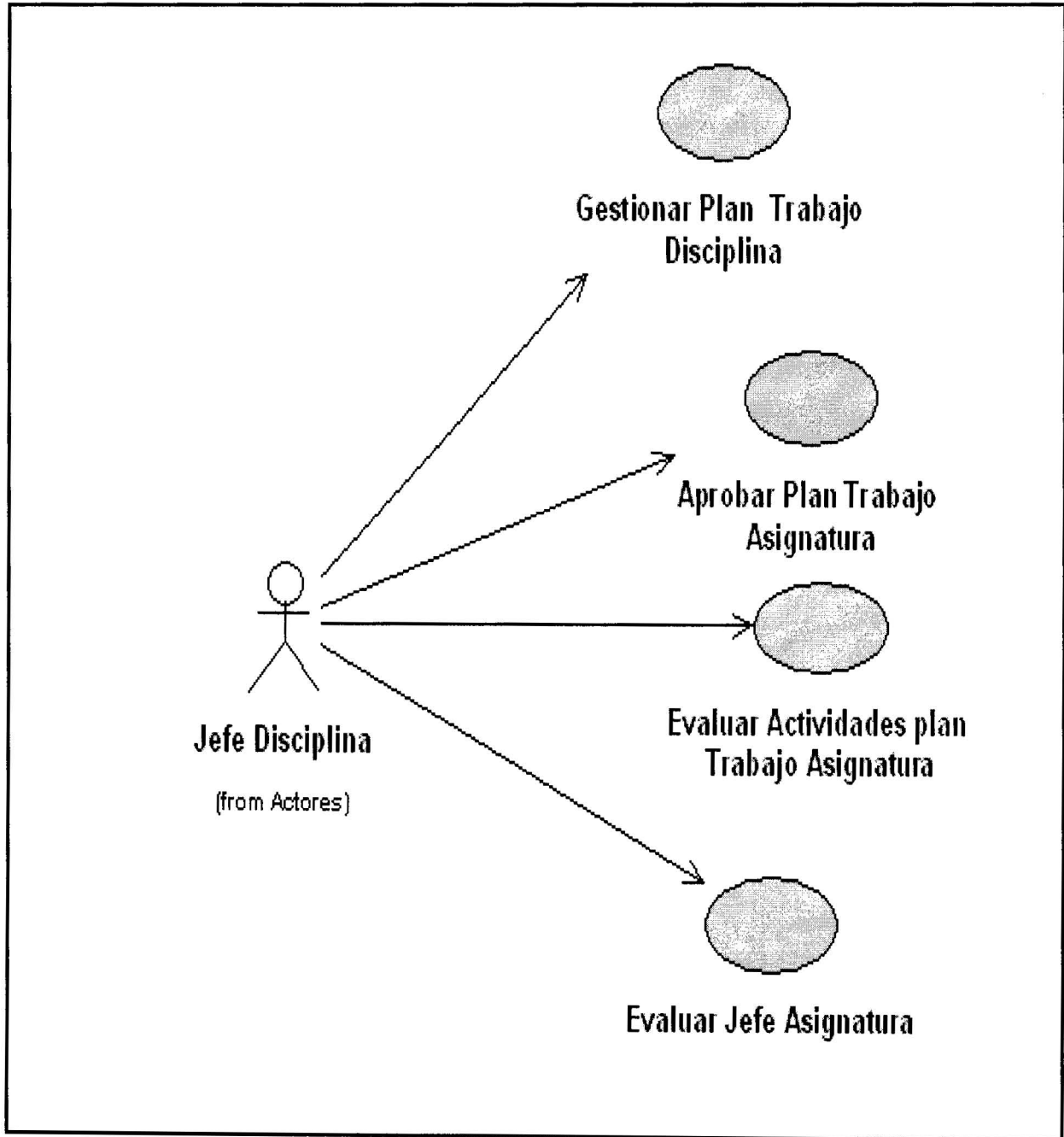


Figura # 3: Diagrama de Caso de Uso Subsistema Jefe Disciplina.

Tabla # 8. CUS Aprobar Plan Trabajo Asignatura.

Caso de Uso:		Aprobar Plan Trabajo Asignatura.
<b>Actor:</b>	Jefe Disciplina (inicia)	
<b>Propósito:</b>	Permite al Jefe de Disciplina aprobar el Plan Trabajo de la Asignatura, de cualquier Asignatura que pertenezca a su Disciplina.	
<b>Resumen:</b>	El CUS se inicia cuando el Jefe de Disciplina selecciona la opción Aprobar Plan Trabajo Asignatura, el sistema le pide que seleccione el Plan de Trabajo de la Asignatura que desea aprobar, el Jefe Disciplina selecciona el Plan, el sistema le muestra el Plan de Trabajo de la Asignatura con sus actividades, luego el Jefe Disciplina aprueba el Plan y el sistema actualiza el Plan de Trabajo de la Asignatura y termina el CUS.	
<b>Referencias:</b>	RF10.	
<b>Precondiciones:</b>	El Plan Trabajo de la Asignatura debe estar creado.	
<b>Poscondiciones:</b>		
<b>Curso Normal de Eventos:</b>		
<b>Acción del Actor:</b>	<b>Respuesta del Sistema:</b>	
1. El Jefe Disciplina selecciona la opción Aprobar Plan Trabajo Asignatura.	1.1- El sistema verifica que sea el Jefe de Disciplina.	
	1.2- El sistema le pide al Jefe Disciplina que seleccione el Plan de Trabajo de la Asignatura que desea aprobar.	
2. El Jefe Disciplina selecciona el Plan de Trabajo de la Asignatura.	2.1- El sistema verifica que el Plan de Trabajo de la Asignatura seleccionado no este aprobado.	
	2.2- El sistema muestra el Plan de Trabajo del Asignatura con sus actividades.	
3. El Jefe Disciplina aprueba el Plan de Trabajo de la Asignatura.	3.1- El sistema actualiza el Plan de Trabajo de la	

	Asignatura finalizando el CUS.
<b>Curso alterno de los eventos:</b>	
<b>Acción 2.1:</b>	Si el Plan de Trabajo de la Asignatura seleccionado ya se encuentra aprobado el sistema muestra un mensaje de error indicando que el Plan de Trabajo de la Asignatura ya se encuentra aprobado y le indica al usuario retornar la acción 2.
<b>Prioridad:</b>	Crítico.



3.5.2.3 Subsistema Jefe Asignatura.

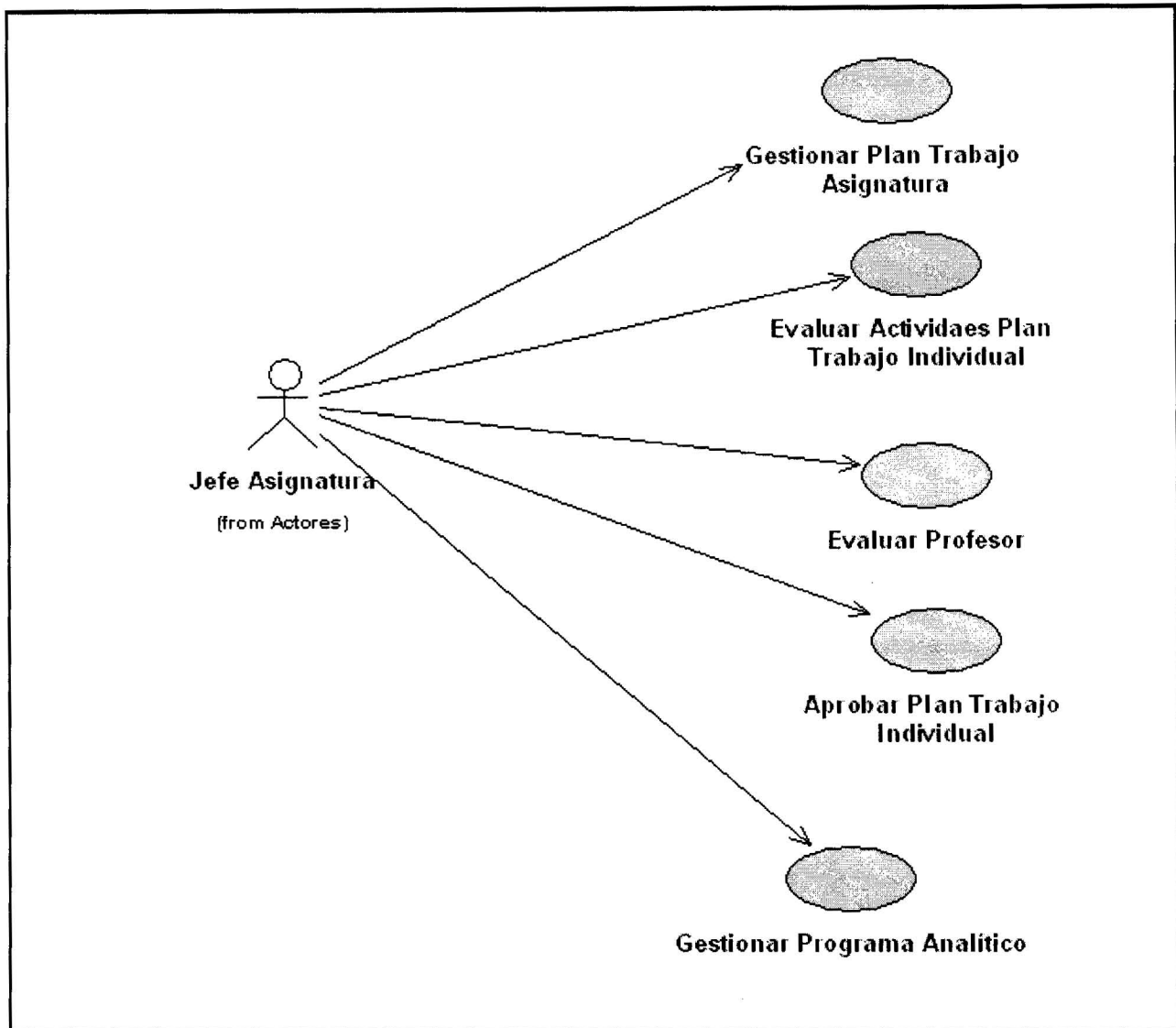


Figura # 4: Diagrama de Caso de Uso Subsistema Jefe Asignatura.

Tabla # 9. CUS Evaluar Profesor.

<b>Caso de Uso:</b>		<b>Evaluar Profesor.</b>
<b>Actor:</b>	Jefe Asignatura (inicia)	
<b>Propósito:</b>	Permite al Jefe de Asignatura evaluar a los Profesores que pertenezcan a su Asignatura.	
<b>Resumen:</b>	El CUS se inicia cuando el Jefe de Asignatura selecciona la opción Evaluar Profesor, el sistema le pide que inserte el usuario del Profesor que desea evaluar, el Jefe de Asignatura inserta el usuario del profesor, el sistema verifica el porcentaje de cumplimiento de sus actividades y le asigna una evaluación y termina el CUS.	
<b>Referencias:</b>	RF 12.	
<b>Precondiciones:</b>	El Plan Trabajo Individual que debe realizar el Profesor debe estar creado y que al menos tenga una actividad que esté evaluada.	
<b>Poscondiciones:</b>		
<b>Curso Normal de Eventos:</b>		
<b>Acción del Actor:</b>	<b>Respuesta del Sistema:</b>	
<p>1. El Jefe Asignatura selecciona la opción Evaluar Profesor.</p> <p>2. El Jefe Asignatura selecciona el Profesor, con el intervalo de fecha que desea evaluar.</p>	<p>1.1- El sistema verifica que sea el Jefe de Asignatura.</p> <p>1.2- El sistema le muestra todos los profesores que pertenezcan a esa Asignatura y el intervalo de fecha que desea evaluar.</p> <p>2.1- El sistema verifica los datos introducidos por el Jefe de Asignatura.</p> <p>2.2-El sistema verifica el porcentaje del cumplimiento de las actividades del Plan de Trabajo Individual correspondiente al Profesor que seleccionó.</p>	

	2.3-El sistema le asigna una evaluación al Profesor finalizando el CUS.
<b>Curso alterno de los eventos:</b>	
<b>Acción 2.1:</b>	Si los datos de selección del Jefe de Asignatura son incorrectos se muestra un mensaje de error y retornando al Jefe de Asignatura a la acción 2.
<b>Prioridad:</b>	Crítico.

Tabla # 10. CUS Gestionar Programa Analítico.

<b>Caso de Uso:</b>		<b>Gestionar Programa Analítico.</b>
<b>Actor:</b>	Jefe Asignatura (inicia)	
<b>Propósito:</b>	Permite al Jefe Asignatura gestionar (Modificar, Eliminar y Adicionar Actividades al Programa Analítico de la Asignatura, además de crear el Programa Analítico de la Asignatura).	
<b>Resumen:</b>	El CUS se inicia cuando el Jefe Asignatura selecciona la opción de Gestionar Programa Analítico, luego selecciona el tipo de gestión, introduce los datos necesarios, el sistema realiza la acción seleccionada por el Jefe de Asignatura y termina el CUS.	
<b>Referencias:</b>	RF 15, RF 15.1, RF 15.2, RF 15.3, RF 15.4	
<b>Precondiciones:</b>	El Jefe de Asignatura debe pertenecer al Departamento.	
<b>Poscondiciones:</b>	<ol style="list-style-type: none"> <li>1 Información de las Actividades del Programa Analítico de la Asignatura adicionada a la Base de Datos.</li> <li>2 Información de las Actividades del Programa Analítico de la Asignatura modificadas a la Base de Datos.</li> <li>3 Información de las Actividades del Programa Analítico de la Asignatura eliminadas de la Base de Datos.</li> <li>4 Información del Programa Analítico de la Asignatura creado en la Base de Datos.</li> </ol>	
<b>Curso Normal de Eventos:</b>		

Acción del Actor:	Respuesta del Sistema:
<p>1. El Jefe Asignatura selecciona la opción de Gestionar Programa Analítico.</p>	<p>1.1- El sistema verifica que sea el Jefe de Asignatura.</p> <p>1.2- El sistema muestra las opciones: Crear Programa Analítico, Adicionar Actividades Programa Analítico, Modificar Actividades Programa Analítico y Eliminar Actividades Programa Analítico.</p>
<p><b>Escenario 1: Crear Programa Analítico.</b></p>	
<p>1. El Jefe Asignatura selecciona la opción de Crear Programa Analítico.</p> <p>2. El Jefe Asignatura introduce los datos solicitados por el sistema.</p>	<p>1.1- El sistema muestra el formulario a completar para la creación del nuevo Programa Analítico.</p> <p>2.1- El sistema verifica los datos introducidos por el Jefe Asignatura.</p> <p>2.2- Si los datos introducidos por el Jefe Asignatura son correctos el sistema adiciona dicha información en la Base de Datos correspondiente y termina el CUS.</p>
<p><b>Curso alternativo de los eventos:</b></p>	
<p><b>Acción 2.2:</b></p>	<p>Si los datos introducidos por el Jefe Asignatura son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al usuario retornar a la acción 2.</p>
<p><b>Escenario 2: Adicionar Actividades Programa Analítico.</b></p>	
<p>1. El Jefe Asignatura selecciona la opción de Adicionar Actividades al Programa Analítico.</p> <p>2. El Jefe Asignatura introduce los datos</p>	<p>1.1- El sistema muestra el formulario a completar para adicionar actividades al Programa Analítico.</p> <p>2.1- El sistema verifica los datos introducidos por el</p>

solicitados por el sistema.	Jefe Asignatura.  2.2- Si los datos introducidos por el Jefe Asignatura son correctos el sistema adiciona dicha información en la Base de Datos correspondiente y termina el CUS.
<b>Curso alterno de los eventos:</b>	
<b>Acción 2.1:</b>	Si los datos introducidos por el Jefe Asignatura son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al usuario retornar a la acción 2.
<b>Escenario 3: Modificar Actividades Programa Analítico.</b>	
1. El Jefe Asignatura selecciona la opción de Modificar Actividades Programa Analítico.  2. El Jefe Asignatura realiza las modificaciones a las Actividades del Programa Analítico.	1.1- El sistema muestra el Programa Analítico con sus Actividades al cual el Jefe Asignatura tiene acceso.  2.1- El sistema verifica los datos modificados por el Jefe Asignatura.  2.2- Si los datos modificados por el Jefe Asignatura son correctos el sistema adiciona dicha información en la Base de Datos correspondiente y termina el CUS.
<b>Curso alterno de los eventos:</b>	
<b>Acción 2.1:</b>	Si los datos modificados por el Jefe Asignatura son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al usuario retornar a la acción 2.
<b>Escenario 4: Eliminar Actividades Programa Analítico.</b>	
1. El Jefe Asignatura selecciona la opción de Eliminar Actividades del Programa Analítico  2. El Jefe Asignatura selecciona la actividad que desea eliminar.	1.1- El sistema muestra las actividades del Programa Analítico a las cuales el Jefe Asignatura tiene acceso.  2.1- El sistema muestra un mensaje de advertencia para la acción a realizar.  3.1- Si el Jefe Asignatura acepta el sistema elimina la actividad del Programa Analítico y culmina el

3. El Jefe Asignatura confirma si quiere eliminar o no la actividad seleccionada.	CUS.
<b>Curso alterno de los eventos:</b>	
<b>Acción 3.1:</b>	Si el Jefe Asignatura cancela la acción se culmina el CUS sin ejecutar ninguna acción.
<b>Prioridad:</b>	Crítico.

3.5.2.3 Subsistema Profesor Regular.

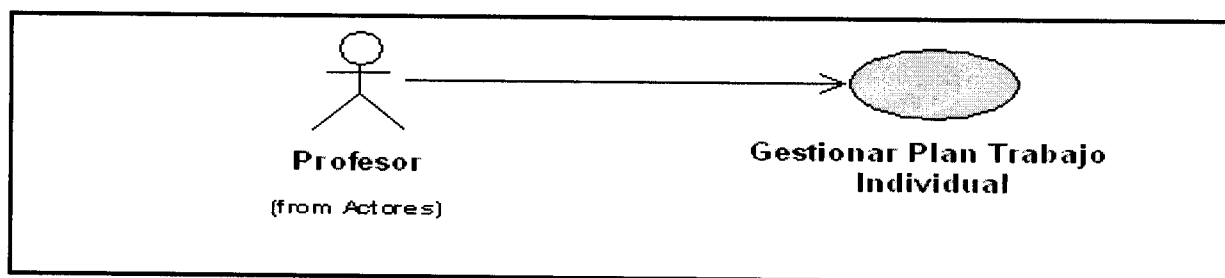


Figura # 5: Diagrama de Caso de Uso Subsistema Profesor.

Tabla # 11. CUS Gestionar Plan Trabajo Individual.

Caso de Uso:		Gestionar Plan de Trabajo Individual.
<b>Actor:</b>	Profesor (inicia)	
<b>Propósito:</b>	Permite al Profesor gestionar (Modificar, Eliminar y Adicionar Actividades a su Plan de Trabajo Individual, además de crear su Plan de Trabajo Individual).	
<b>Resumen:</b>	El CUS se inicia cuando el Profesor selecciona la opción de Gestionar Plan Trabajo Individual, luego selecciona el tipo de gestión, introduce los datos necesarios, el sistema realiza la acción seleccionada por el Profesor y termina el CUS.	
<b>Referencias:</b>	RF 14, RF 14.1, RF 14.2, RF 14.3, RF 14.4, RF 14.5	
<b>Precondiciones:</b>	El Profesor debe pertenecer al Departamento.	
<b>Poscondiciones:</b>	1 Información de las Actividades del Plan de Trabajo Individual adicionada a la Base de Datos.	

	<p>2 Información de las Actividades del Plan de Trabajo Individual modificadas a la Base de Datos.</p> <p>3 Información de las Actividades del Plan de Trabajo Individual eliminadas de la Base de Datos.</p> <p>4 Información del Plan de Trabajo Individual creado en la Base de Datos.</p>
<b>Curso Normal de Eventos:</b>	
<b>Acción del Actor:</b>	<b>Respuesta del Sistema:</b>
<p>1. El Profesor selecciona la opción de Gestionar Plan de Trabajo Individual.</p>	<p>1.1- El sistema verifica que exista el Profesor en el Departamento.</p> <p>1.2- El sistema muestra las opciones: Crear Plan Trabajo Individual, Adicionar Actividades Plan Trabajo Individual, Modificar Actividades Plan Trabajo Individual y Eliminar Actividades Plan de Trabajo Individual.</p>
<b>Escenario 1: Crear Plan Trabajo Individual.</b>	
<p>1. El Profesor selecciona la opción de Crear Plan de Trabajo Individual.</p> <p>2. El Profesor introduce los datos solicitados por el sistema.</p>	<p>1.1-El sistema muestra el formulario a completar para la creación del nuevo Plan de Trabajo Individual.</p> <p>2.1-El sistema verifica los datos introducidos por el Profesor.</p> <p>2.2- Si los datos introducidos por el Profesor son correctos el sistema adiciona dicha información en la Base de Datos correspondiente y termina el CUS.</p>
<b>Curso alternativo de los eventos:</b>	

<p><b>Acción 2.2:</b></p>	<p>Si los datos introducidos por el Profesor son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al usuario retornar a la acción 2.</p>
<p><b>Escenario 2: Adicionar Actividades Plan Trabajo Individual.</b></p>	
<p>1. El Profesor selecciona la opción de Adicionar Actividades al Plan Trabajo Individual.</p> <p>2. El Profesor introduce los datos solicitados por el sistema.</p>	<p>1.1- El sistema muestra el formulario a completar para adicionar actividades al Plan Trabajo Individual.</p> <p>2.1- El sistema verifica los datos introducidos por el Profesor.</p> <p>2.2- Si los datos introducidos por el Profesor son correctos el sistema adiciona dicha información en la Base de Datos correspondiente y termina el CUS.</p>
<p><b>Curso alternativo de los eventos:</b></p>	
<p><b>Acción 2.1:</b></p>	<p>Si los datos introducidos por el Profesor son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al usuario retornar a la acción 2.</p>
<p><b>Escenario 3: Modificar Actividades Plan Trabajo Individual.</b></p>	
<p>1. El Profesor selecciona la opción de Modificar Actividades del Plan Trabajo Individual.</p> <p>2. El Profesor realiza las modificaciones a las Actividades del Plan Trabajo Individual.</p>	<p>1.1- El sistema muestra el Plan Trabajo Individual con sus Actividades al cual el Profesor tiene acceso.</p> <p>2.1- El sistema verifica los datos modificados por el Profesor.</p> <p>2.2- Si los datos modificados por el Profesor son correctos el sistema adiciona dicha información en la Base de Datos correspondiente y termina el CUS.</p>
<p><b>Curso alternativo de los eventos:</b></p>	
<p><b>Acción 2.1:</b></p>	<p>Si los datos modificados por el Profesor son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al</p>



	usuario retornar a la acción 2.
<b>Escenario 4: Eliminar Actividades Plan Trabajo Individual.</b>	
1. El Profesor selecciona la opción de Eliminar Actividades al Plan Trabajo Individual.	1.1- El sistema muestra las actividades Plan Trabajo Individual a las cuales el Profesor tiene acceso.
2. El Profesor selecciona la actividad que desea eliminar.	2.1- El sistema muestra un mensaje de advertencia para la acción a realizar.
3. El Profesor confirma si desea o no eliminar la actividad seleccionada.	3.1- Si el Profesor acepta, el sistema elimina la actividad del Plan de Trabajo Individual y culmina el CUS.
<b>Curso alternativo de los eventos:</b>	
<b>Acción 3.1:</b>	Si el Profesor cancela la acción se culmina el CUS sin ejecutar ninguna acción.
<b>Prioridad:</b>	Crítico.

### 3.6 Conclusiones.

El correcto proceso de obtención de los requerimientos del sistema es de suma importancia en todo el ciclo de desarrollo de un producto, de aquí que los esfuerzos en este capítulo estén encaminados en su correcta extracción. En este capítulo se comenzó a desarrollar la propuesta de solución a partir del estudio realizado del modelo de Dominio. Se hace una breve explicación de cuando es necesario aplicar un modelo de Dominio, además se explica cual es el entorno organizacional del trabajo. Se definieron los requisitos funcionales y no funcionales que debe cumplir el sistema en toda su totalidad, así como los actores del mismo, incluyendo la descripción de cada uno de de los casos de usos encontrados en el sistema. Con todas estas especificaciones se está en condiciones de comenzar a construir el sistema siguiendo a cabalidad los requisitos encontrados en este capítulo.

### Capítulo 4 “Construcción de la solución propuesta”

#### 4.1 Introducción.

En este capítulo se ven los resultados de la etapa de diseño del sistema. Se describe la puesta en práctica de la construcción de la solución propuesta. Se describe la forma en que se realiza la implementación a través del diagrama de clases persistentes y el modelo de datos. Se hace referencia a los principios de diseño, haciendo referencia a los estándares aplicados a la interfaz de la aplicación, se hace referencia al tratamiento de errores y a la generación de reportes, además se ve el modelo de implementación mediante el diagrama de despliegue y de componentes.

4.2 Diagrama de Clases de Diseño Web.

4.2.1 Subsistema Jefe Departamento.

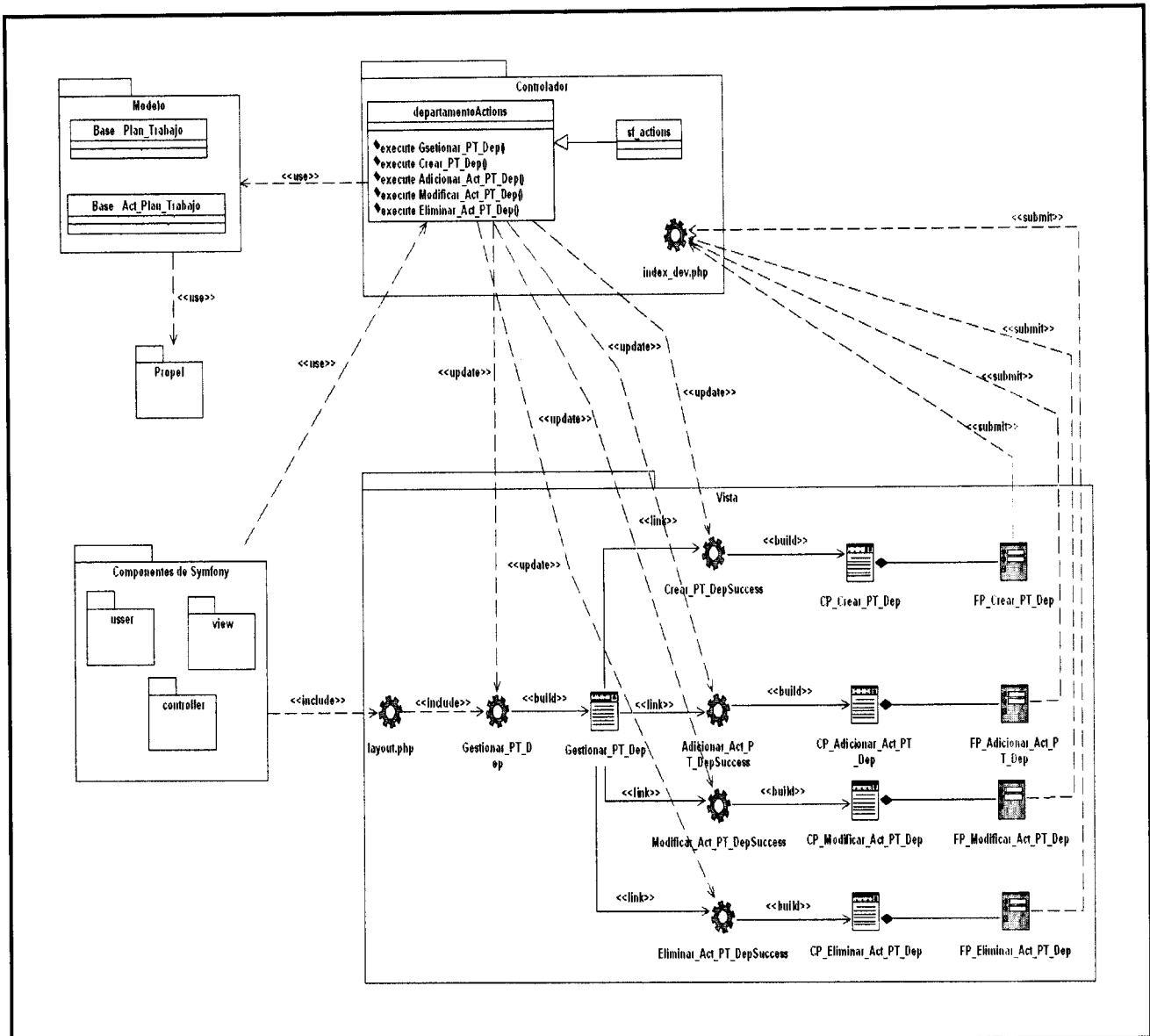


Figura # 6: Gestionar Plan de Trabajo Departamento.

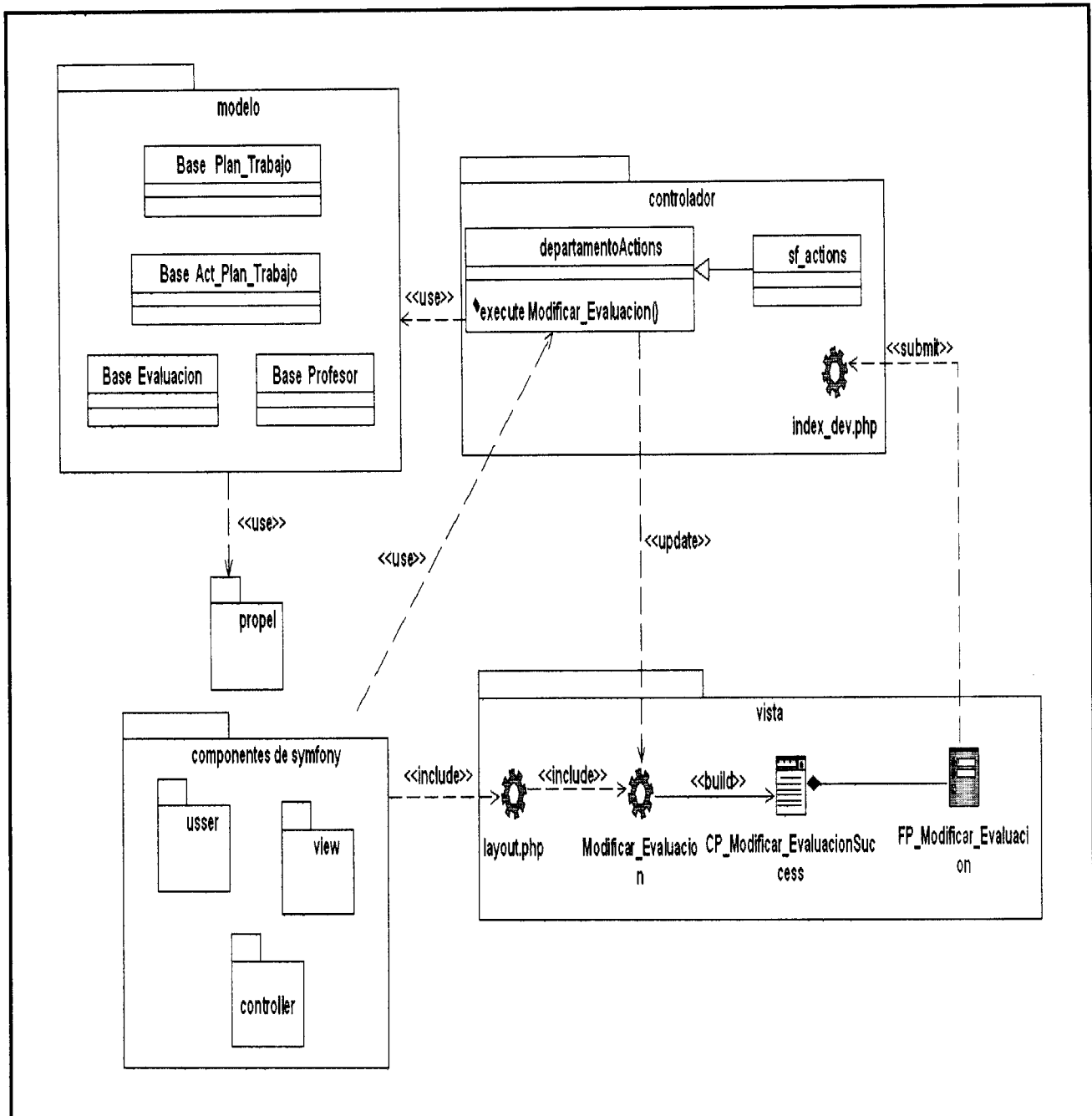


Figura # 7: Modificar Evaluación.

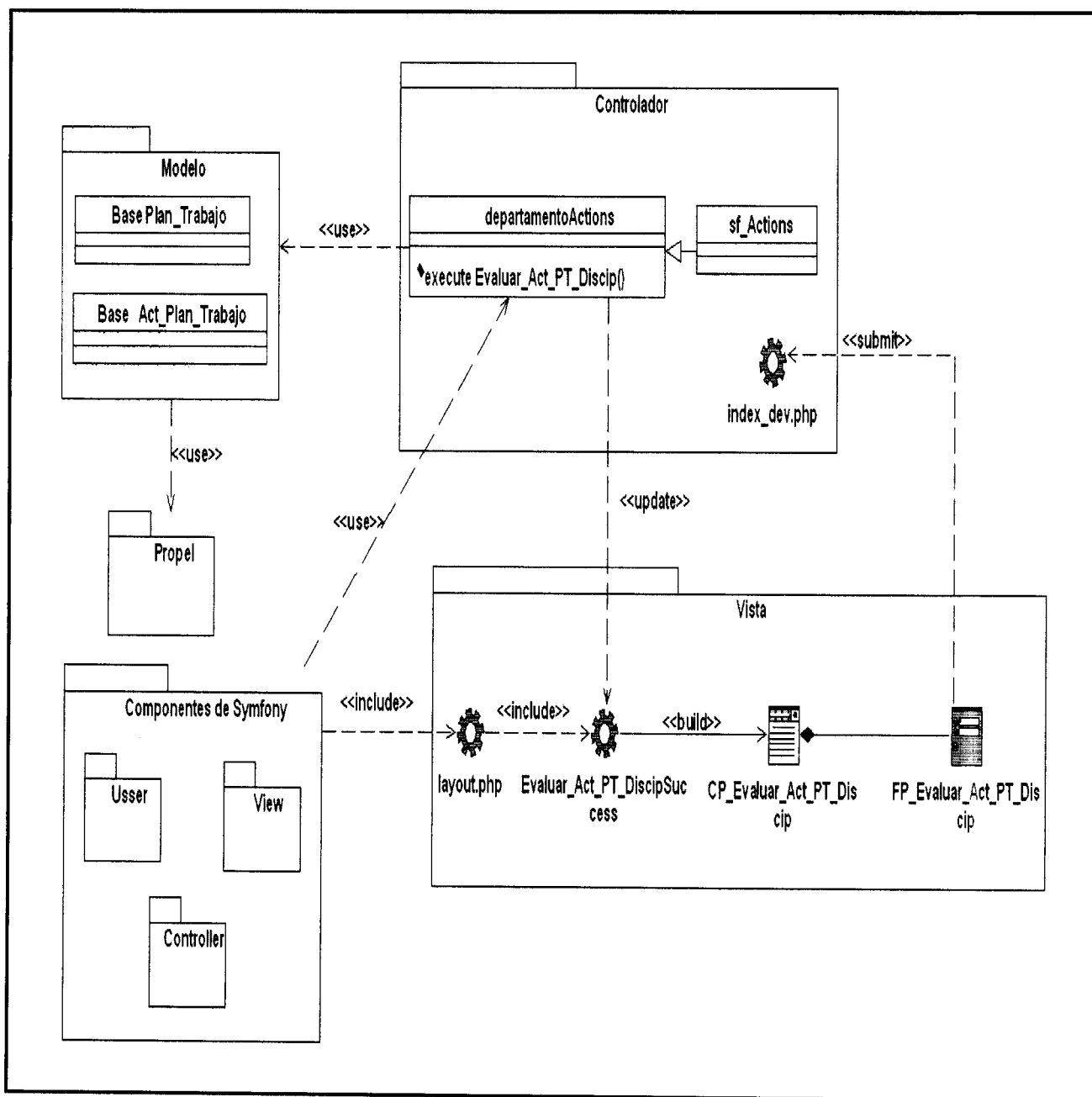


Figura # 8: Evaluar Actividades Plan de Trabajo Disciplina.

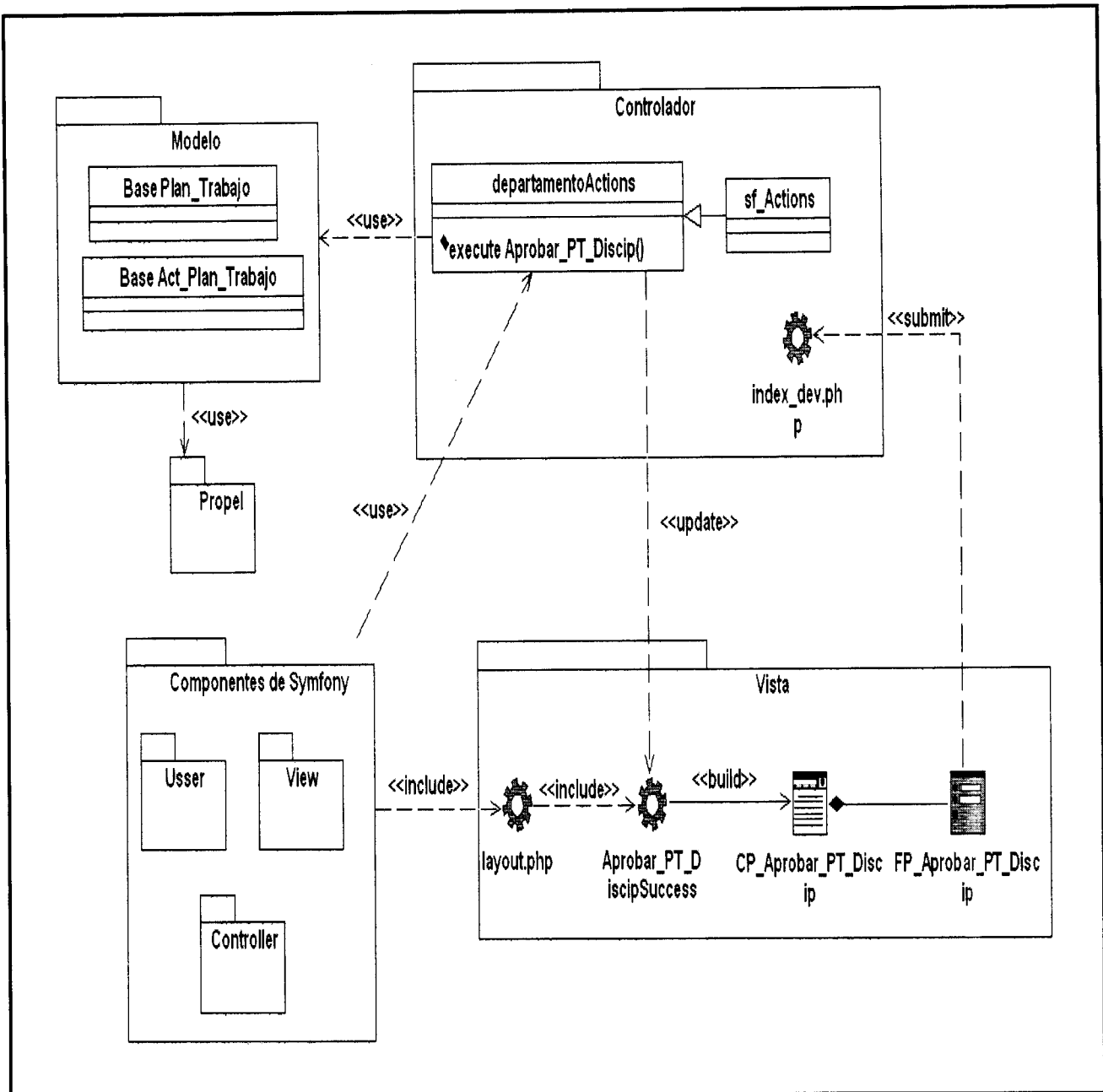


Figura # 9: Aprobar Plan de Trabajo Disciplina.

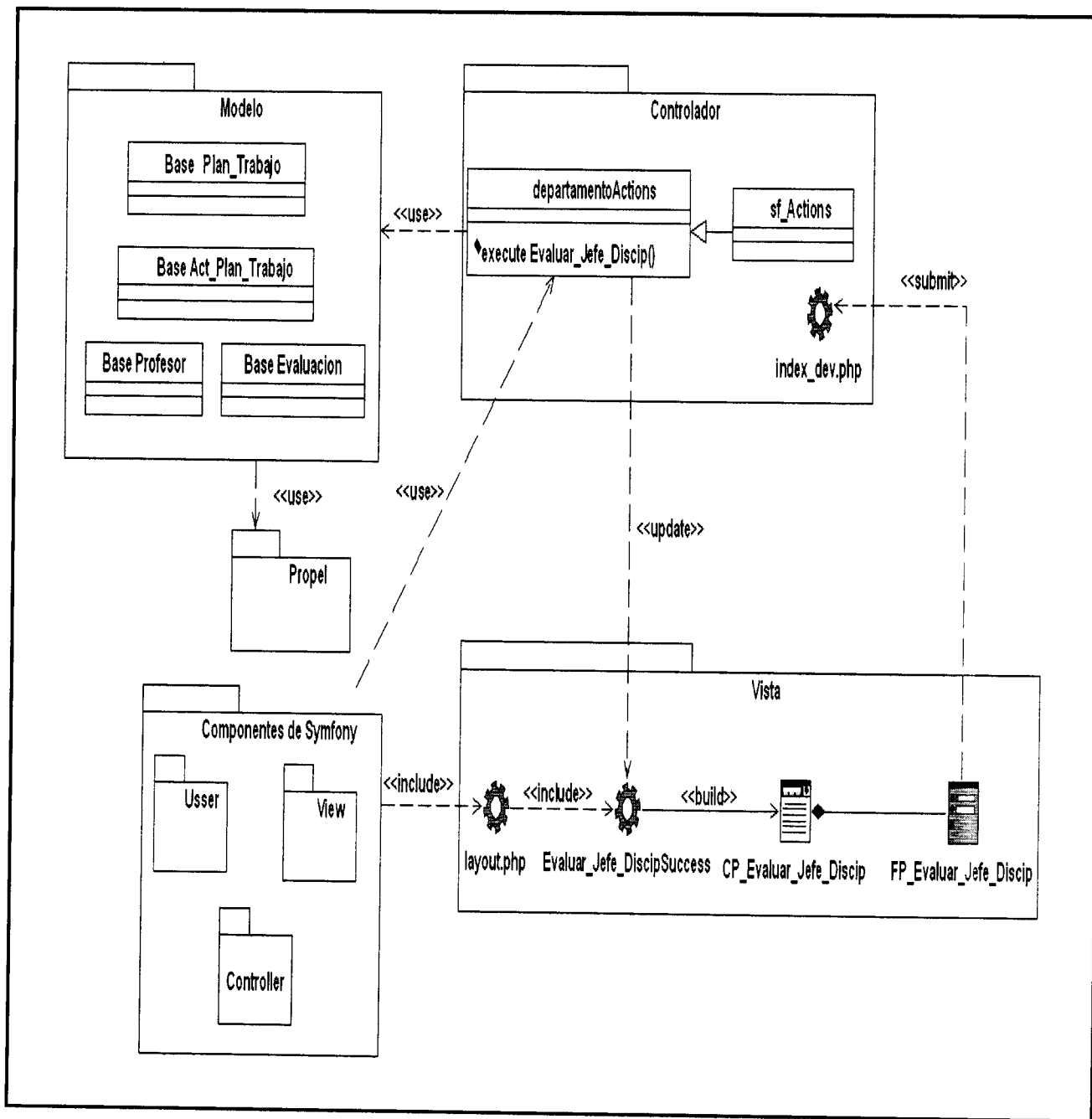


Figura # 10: Evaluar Jefe Disciplina.

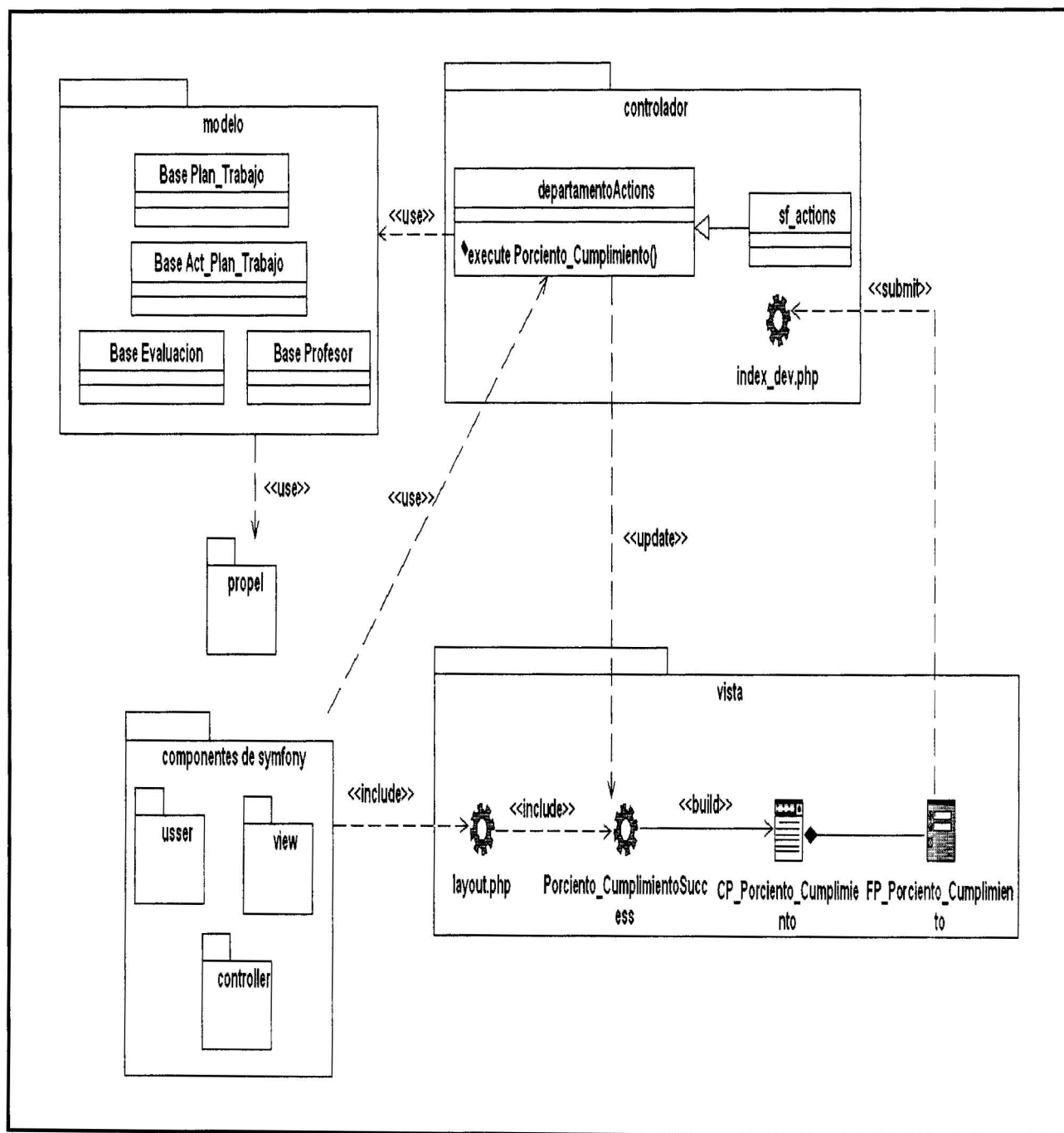


Figura # 11: Generar Porciento Cumplimiento.



4.2.2 Subsistema Jefe Disciplina.

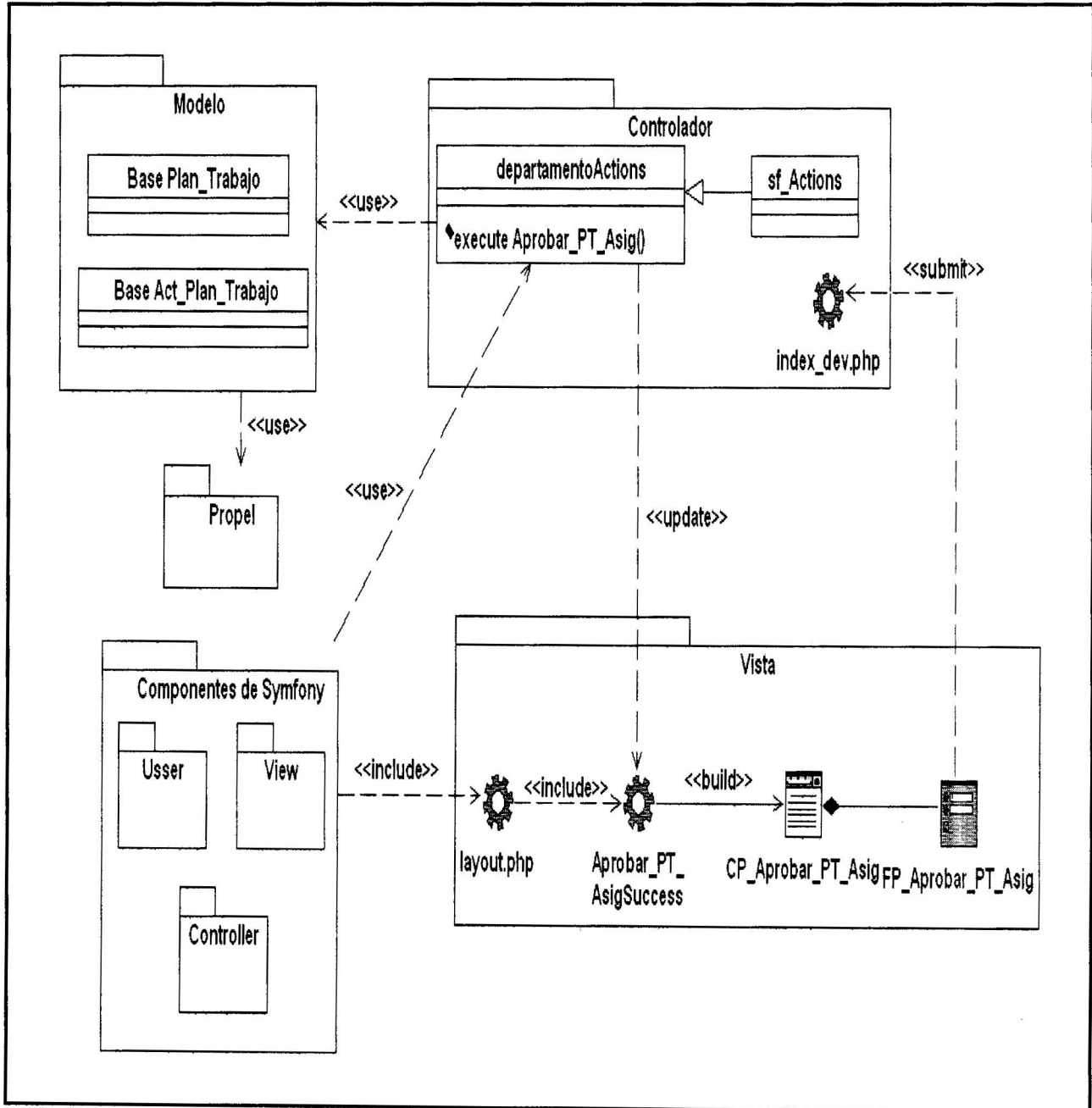


Figura # 12: Aprobar Plan de Trabajo Asignatura.

4.2.3 Subsistema Jefe de Asignatura.

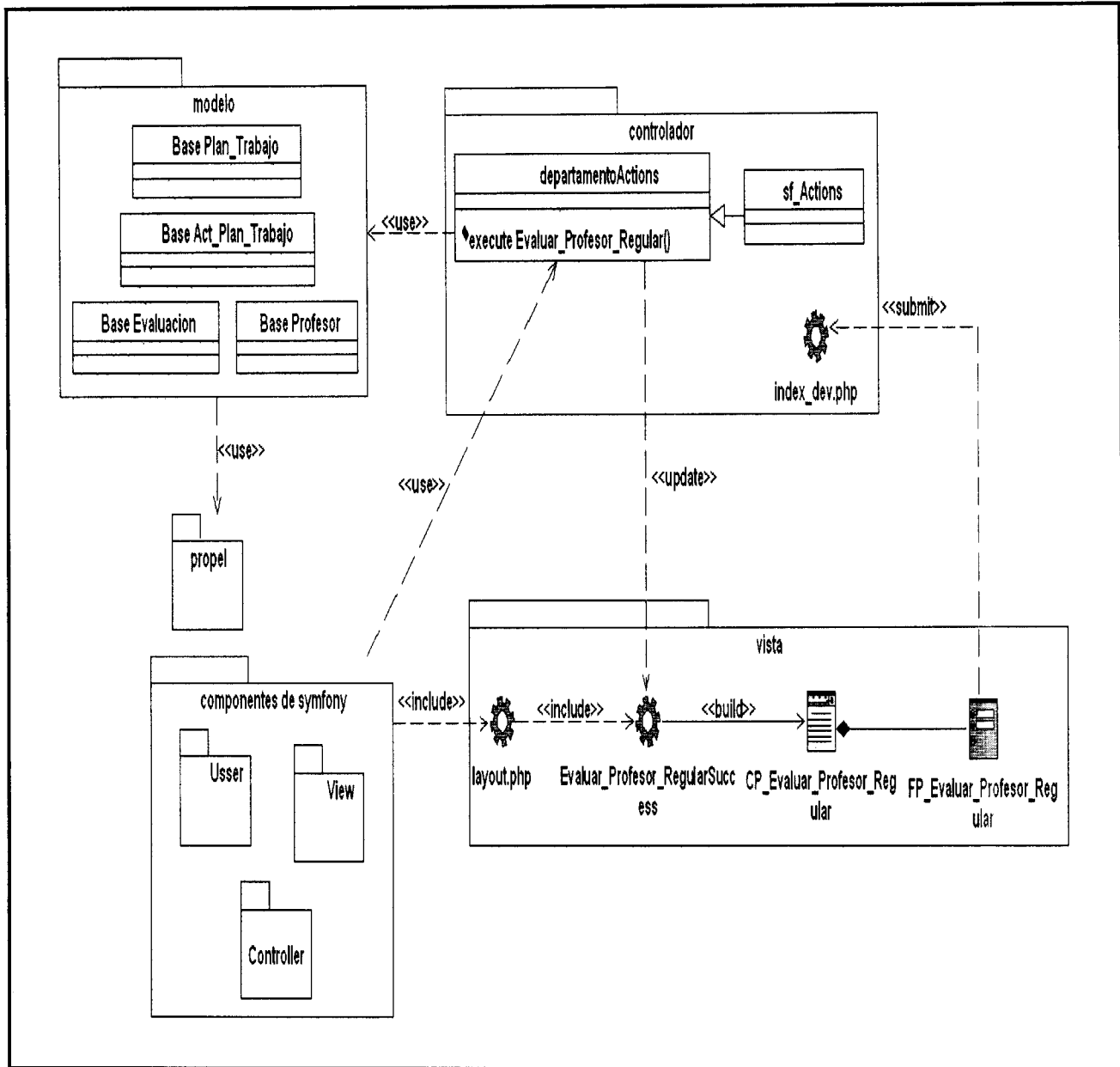


Figura # 13: Evaluador Profesor.

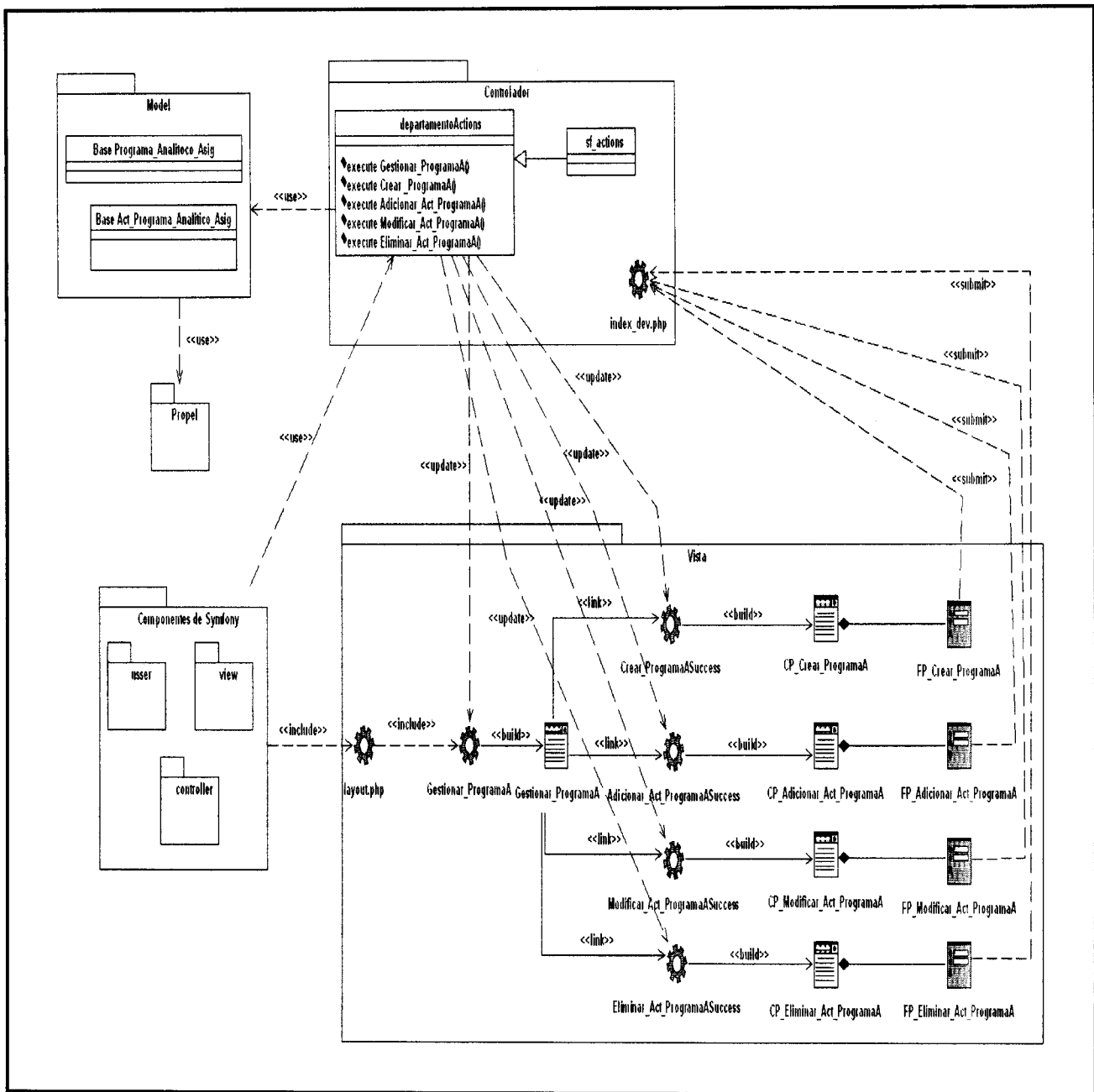


Figura # 14: Gestionar Programa Analítico.

4.2.4 Subsistema Profesor Regular.

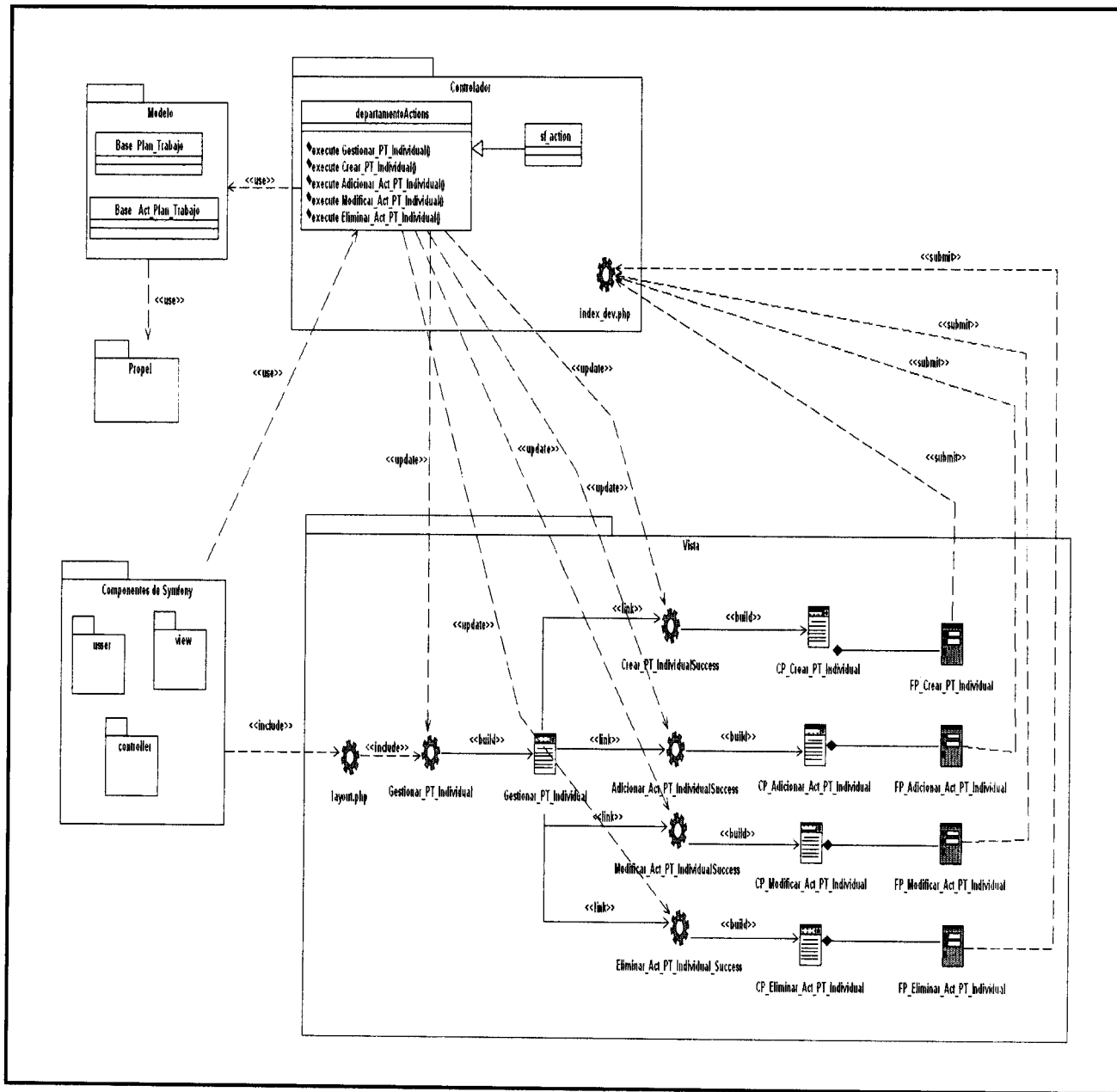


Figura # 15: Gestionar Plan de Trabajo Individual.

### **4.3 Principios de diseño.**

#### **4.3.1 Estándares de la interfaz de la aplicación.**

El diseño de interfaces de usuario es una tarea que ha adquirido relevancia en el desarrollo de un sistema. La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso, ya que mediante la misma el cliente o usuario obtiene la primera impresión del sistema, es por eso que uno de los aspectos más relevantes de la usabilidad de un sistema es la consistencia de su interfaz de usuario. El producto es legible y con colores agradables, preferentemente claros dándole a la página una sensación de seguridad. El diseño de la interfaz es sencillo ya que el sistema no requiere de grandes cargas en formularios, además debe cumplir con el patrón de diseño establecido por los restantes módulos que conforman el Sistema de Informatización. Para lograr una mayor visualización de la información, se utilizó un mismo color en todas las páginas con textos claros. El envío de la información debe ser lo más rápida posible por lo que no se utilizaron muchas imágenes ni funciones que atenten contra esto y que posibiliten también una navegación rápida y eficiente.

#### **4.3.2 Tratamiento de Errores.**

Se cuenta con un sistema de tratamiento de errores para disminuir la posibilidad de cometerlos. Para esto se cuenta con la validación de la información introducida en el sistema por la validación de los formularios mediante funciones JavaScript y mediante funciones definidas por el framework Symfony. Una vez que es detectado un error, ya sea un dato mal escrito, o algún campo que se dejó vacío y era necesario llenarlo, la aplicación lanzará un mensaje de error, indicándole cual es el dato que está mal o que falta y dirigirá al usuario al campo en el cual está el error; existe un sistema de menú que indica al usuario donde debe entrar según el Departamento al que pertenece, si el mismo trata de acceder a uno donde no pertenece, el sistema muestra un mensaje informándole que no pertenece a dicho Departamento.

#### **4.3.3 Generación de Reportes.**

La aplicación tiene dentro de sus principales funcionalidades, además de mantener un control al registrar la información referente a los Departamentos Docentes, permitir generar reportes que brindan la información al Jefe de Departamento de cómo va el porcentaje de cumplimiento de la realización de las actividades de los planes de trabajo de su departamento, reportes que posteriormente podrán darle un margen al Jefe Departamento para la evaluación de los profesores. Los reportes se obtendrán mediante el promedio de la puntuación dada a un Plan de Trabajo, que no es más que la suma de

todos los puntos que se dan por actividades entre la cantidad de actividades que conforman el Plan de Trabajo.

#### **4.3.4 Concepción general de la ayuda.**

El sistema cuenta con una ayuda que le facilitará al usuario que interactuó con la aplicación la navegabilidad dentro de la misma.

#### **4.4 Diseño de la Base de Datos.**

Para diseñar la base de datos del sistema, se utilizó el diagrama de clases persistentes y el modelo de datos, estas clases representan los datos que se obtienen y almacenan durante los procesos de la aplicación, estos son los que pueden modelarse a través de un diagrama de clases persistentes, lo que permitirá ver la relación entre los datos, y completará la modelación de la lógica de negocio de la aplicación.

Diagrama de Clases Persistentes.

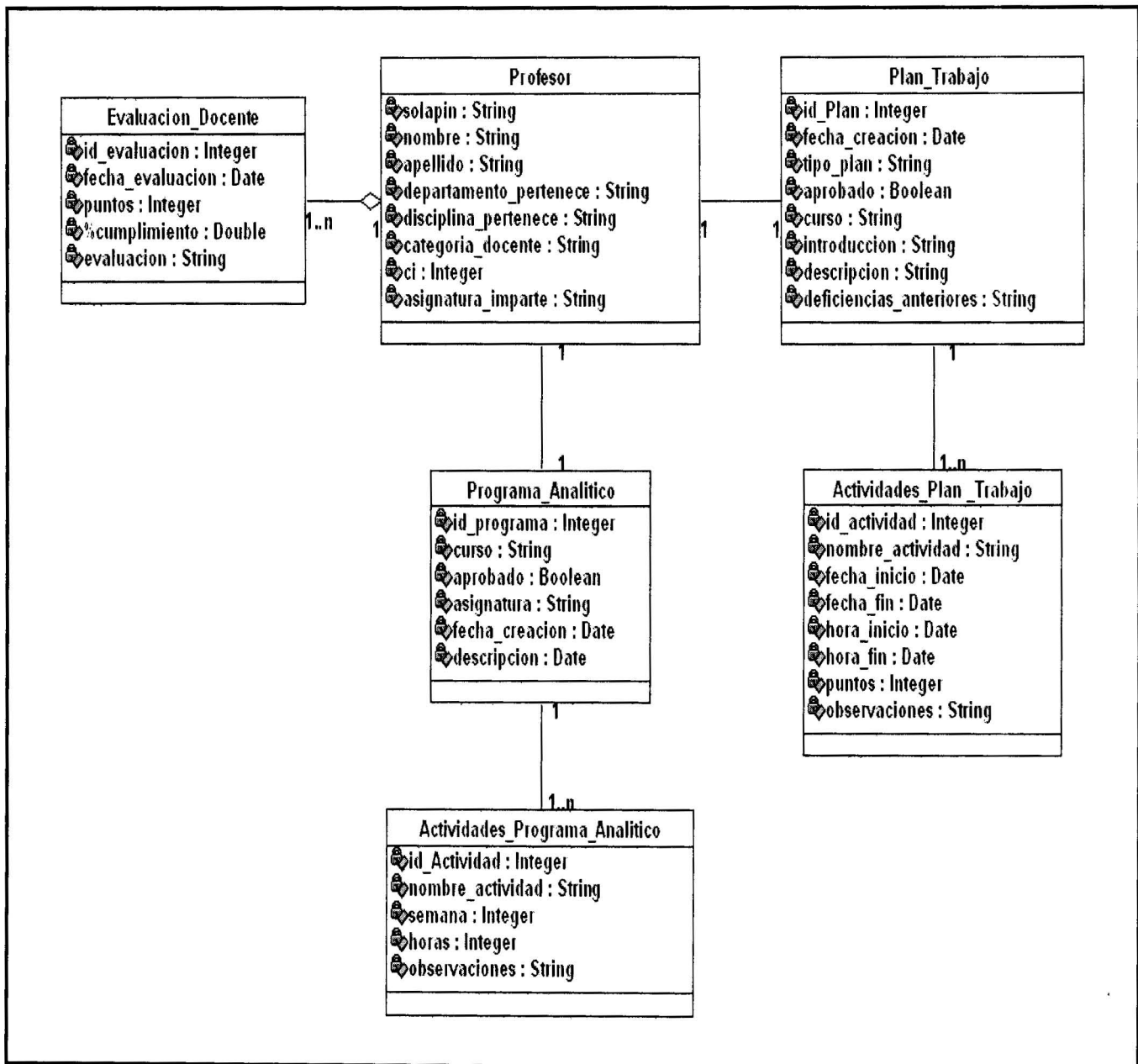


Figura # 16: Diagrama de Clases Persistentes.

Diagrama de Clases del Modelo de Datos.

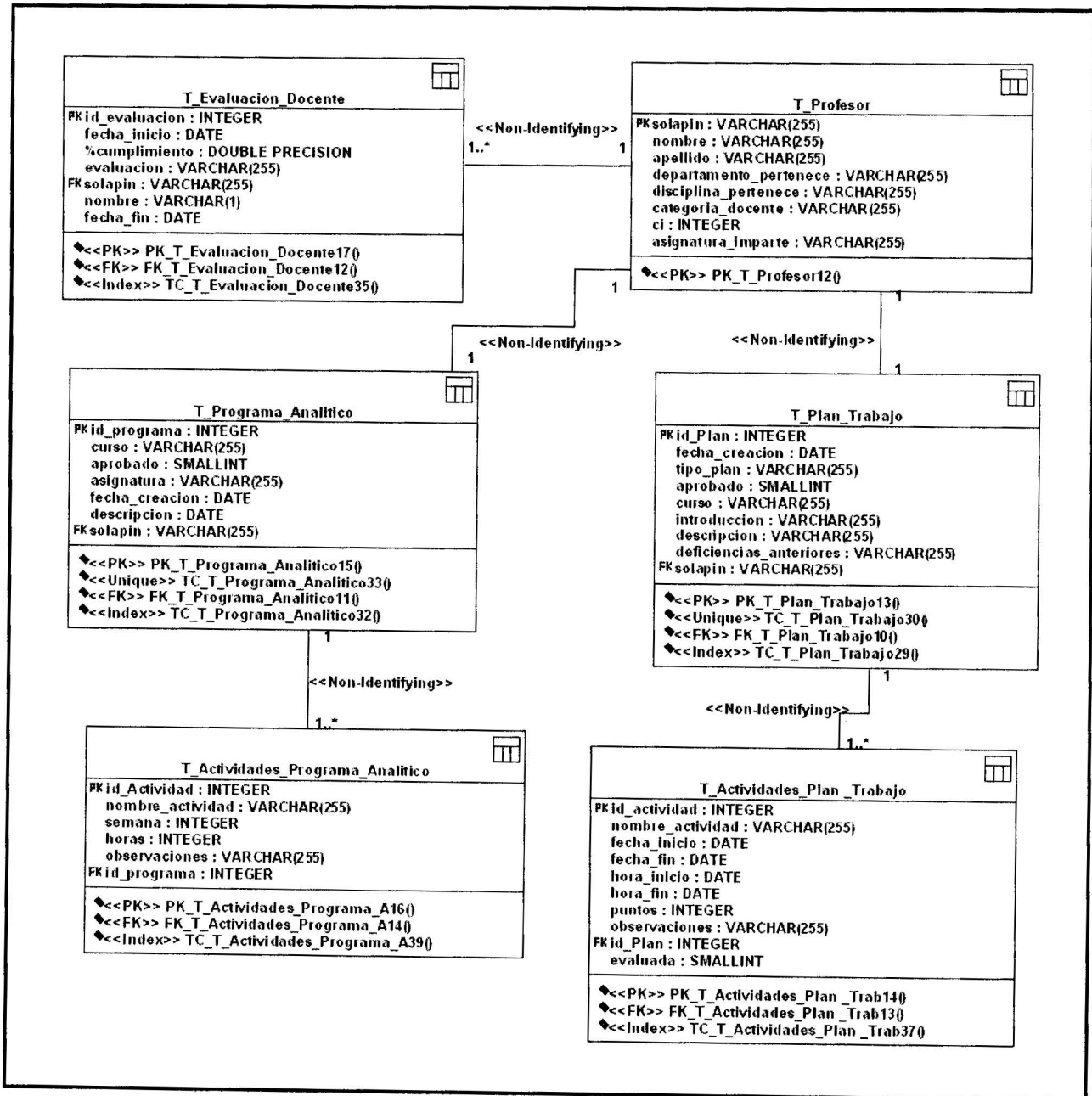


Figura # 17: Diagrama de Clases del Modelo de Datos.



### **4.5 Generalidades de la Implementación.**

La seguridad en el sistema está implementada por parte del servidor de base de datos PostgreSQL y el uso de variables de sesión para restringir el acceso de los usuarios a determinadas páginas. Además está implementado un Sub Módulo de autenticación que estará encargado de verificar que sólo puedan acceder a la aplicación usuarios registrados. Para garantizar la seguridad de la información almacenada se crearon varios niveles de seguridad, definidos como tipos de usuarios cada uno con los privilegios con los que puede contar, que pueden ser: Jefe Departamento, que es el único que puede generar el porcentaje de cumplimiento de los planes de trabajo del departamento, existe también el Jefe de Disciplina, Jefe de Asignatura y el Profesor Regular, todos ellos con un nivel de privilegio asignado según su categoría docente. Para esto se trabaja con variables de sesión de forma tal que siempre se sabe que usuario intenta visitar dichas páginas y estas se muestran sólo para aquellos que pueden tener acceso a ellas. Se usa el patrón de diseño Decorator el cual ayuda a una mejor estructura e implementación del sistema.

### **4.6 Modelo de Despliegue.**

El diagrama de despliegue permite apreciar de forma visual como se encuentran relacionados físicamente los componentes de la aplicación. En este caso el usuario accede al sistema desde un navegador Web por medio del protocolo HTTP. La aplicación se encuentra hospedada en un Servidor Web, el cual se conecta al servidor de base de datos (PostgreSQL) mediante el protocolo TCP/IP, además el servidor de aplicaciones se conecta al servidor donde se encuentra el Directorio de Servicios Web de la UCI mediante el protocolo SOAP y al Servidor de Dominio UCI mediante el protocolo LDAP. En cada PC cliente está conectada una impresora.

Diagrama de Despliegue.

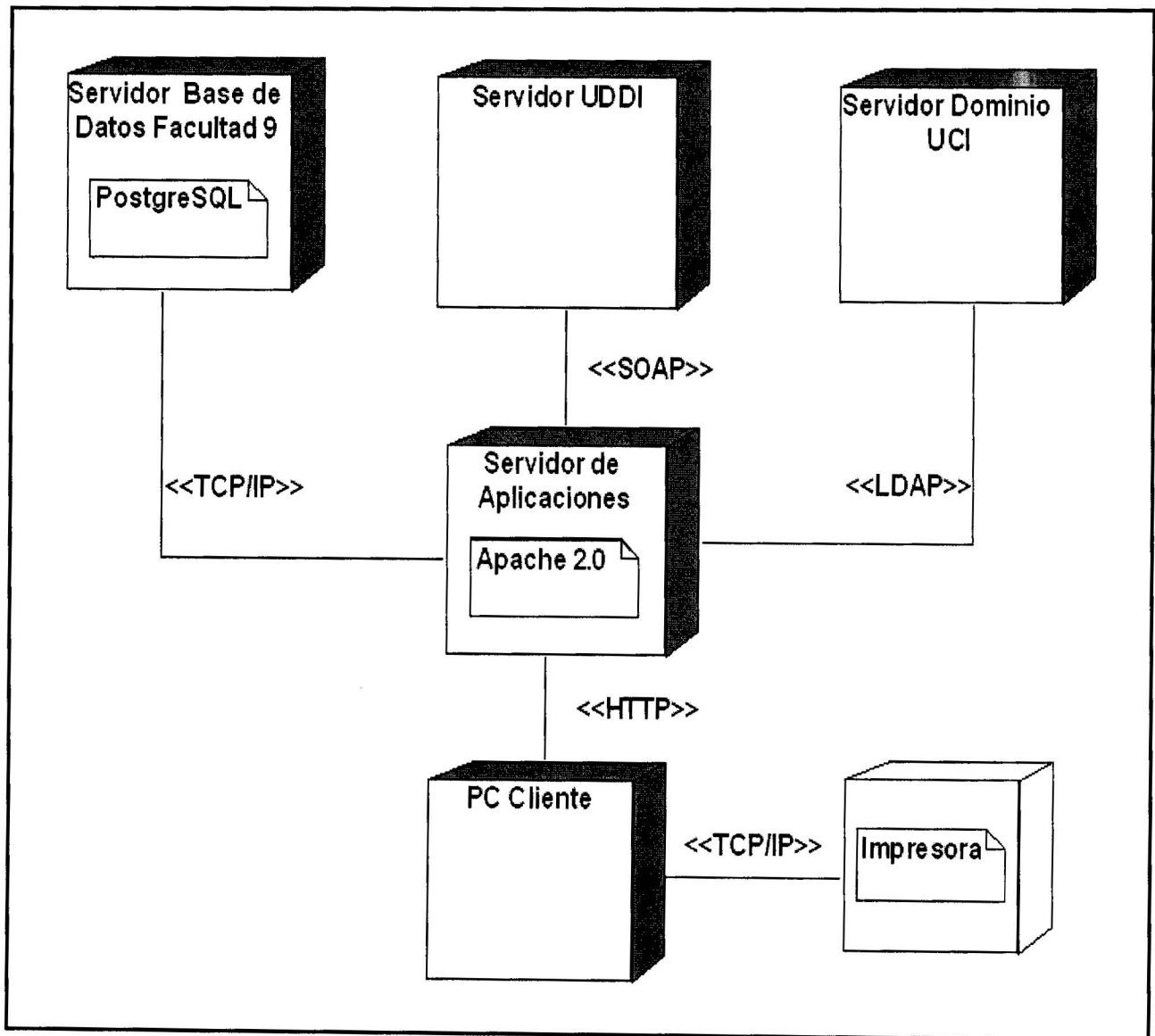


Figura # 18: Diagrama de Despliegue.

4.7 Modelo de Implementación.

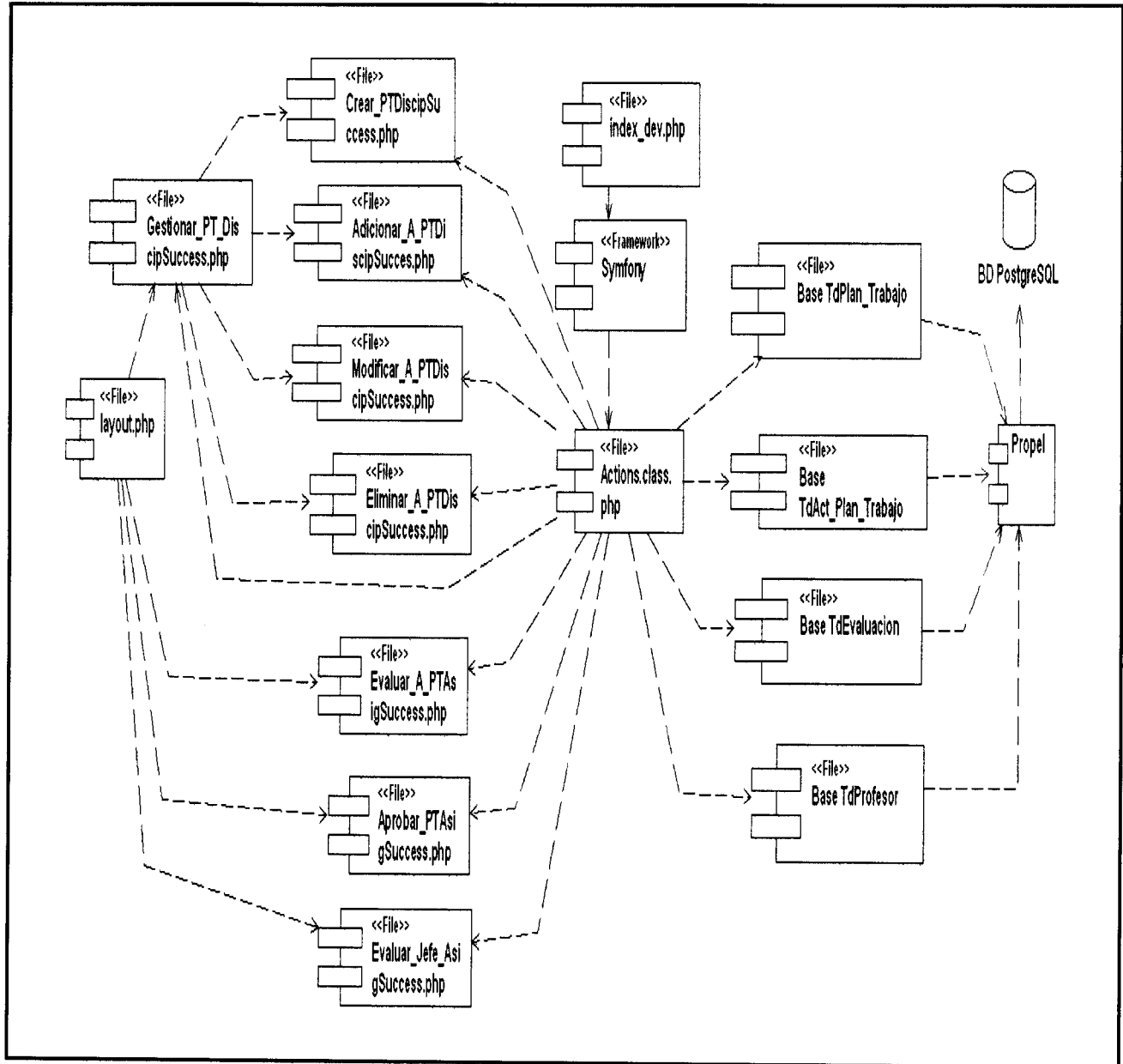


Figura # 19: Diagrama de Componentes Subsistema Jefe Disciplina.

#### **4.8 Conclusiones.**

En este capítulo se ven aspectos relacionados con la construcción e implementación del producto. Se definen los diagramas de clases del diseño, se estructura el modelo de datos y se describe el diagrama de despliegue, fundamental en la comprensión de la distribución física del sistema, además de construir el modelo de implementación mediante el diagrama de componentes.

### CONCLUSIONES

Con la realización del presente Trabajo de Diploma se dota a los Departamentos Docentes de la Facultad 9 de la Universidad de las Ciencias Informáticas de un prototipo funcional para la gestión de sus Departamentos Docentes, lo que posibilitará un mejor control de los procesos que se manejan en dichos departamentos.

El desarrollo exitoso del sistema fue posible gracias al estudio y utilización de las principales tendencias y tecnologías par el desarrollo de aplicaciones Web llegando las siguientes conclusiones:

- El estudio de las condiciones actuales en las cuales se desarrolla el proceso de control de las estadísticas, documentos y flujo de información de los Departamentos Docentes de la Facultad 9 permitió entender mejor la problemática que llevó a la realización de la investigación científica.
- El estudio del arte de las principales y más actuales tecnologías utilizadas para el diseño y desarrollo de aplicaciones Web en Cuba y el mundo dio la posibilidad de construir un sistema fiable y seguro.
- Los objetivos planteados para el correcto desarrollo del sistema se cumplieron, quedando planteado el flujo de información, sus nodos centrales, así como los datos de entrada y reportes para la actividad de control de las estadísticas, documentos y flujo de la información que se maneja en los departamentos docentes de la facultad.
- Se logró la implementación de una aplicación, que brinda una serie de funcionalidades importantes para el correcto funcionamiento de los Departamentos Docentes.
- La utilización de la aplicación asegura que los profesores de la facultad puedan gestionar y tener un mejor control de la información referente a los Departamentos Docentes.

### RECOMENDACIONES

Después de la culminación de este trabajo se impone dar una serie de recomendaciones para un mejor perfeccionamiento y aprovechamiento de la aplicación, estas son:

- Continuar la investigación para lograr nuevas mejoras en posteriores versiones del subsistema.
- Se recomienda el perfeccionamiento del diseño de la aplicación.
- Continuar el desarrollo de este sistema, adicionándole nuevas funcionalidades y dándole seguimiento a las actividades más importantes de los Departamentos Docentes.
- Darle mantenimiento a la aplicación.

### Referencias Bibliográficas

1. Hernández, A.I.C. and D.G. Pérez, SUBSISTEMA DE GESTIÓN DE LOS DEPARTAMENTOS DOCENTES DE LA FACULTAD 9. 2007, Universidad de las Ciencias Informáticas: Ciudad de la Habana. p. 132.
2. José, V.S. La construcción de Modelos de Objetos con técnicas de Servicios y Subsistemas
3. 1996 [cited; Available from: <http://www.ati.es/qt/LATIGOO/OOp96/Ponen10/atio6p10.html>].
4. Cotilla., D.J.M.P.R.D.Z.R. CONSIDERACIONES TEORICAS Y EXPERIENCIAS EN EL ANÁLISIS Y MEJORA DE LOS PROCESOS. Volume, 18
5. [cited; Available from: <http://www.desarrolloweb.com/articulos/25.php>].
6. Kenedy, C.M.y.B., HTML La Guía Completa.
7. Ivar Jacobson, G.B. and J. Rumbaugh, El Proceso Unificado de Desarrollo de Software, Addison Wesley
8. 2007[cited; Available from:  
<http://www-306.ibm.com/software/awdtools/developer/rose/enterprise/index.html>].
9. Conceptos básicos de DREAMWEAVER MX 2003 [cited; Available from:  
[http://www.aulacli.com/dreamweaverMX/t\\_1\\_1.htm](http://www.aulacli.com/dreamweaverMX/t_1_1.htm)].
10. Welicki, L., Patrones y Antipatrones: una Introducción.
11. Iglesias, C.B.R.R.A. GESTIÓN DEL CONOCIMIENTO Y GESTIÓN DE LA INFORMACIÓN. 2001 [cited; Available from: <http://www.inforarea.es/Documentos/GC.pdf>].
12. Cumbre Mundial sobre la Sociedad de la Información. 2005 [cited; Available from:  
<http://www.cubaminrex.cu/Actualidad/2005/>].
13. 2005 [cited; Available from:  
<http://www.linuxparatodos.net/portal/staticpages/index.php?page=servidor-web>].
14. Una Introducción al APACHE. 2006, [cited; Available from:  
[http://linux.ciberaula.com/articulo/linux\\_apache\\_intro/](http://linux.ciberaula.com/articulo/linux_apache_intro/)
15. Molpeceres, Alberto. Procesos de desarrollo: RUP, XP, y FDD, 2003. [Disponible en:  
<http://www.javahispano.org/articles.article.action?id=76>]

### Bibliografía

- JACOBSON, I.; G. BOOCH, et al. El Proceso Unificado de Desarrollo de Software. Madrid, Pearson Educación, 2000.
- LARMAN, C. UML y Patrones. Introducción al análisis y diseño orientado a objetos. México, Prentice Hall, 1999.
- PRESSMAN, R. S. Ingeniería de Software. Un enfoque práctico. Madrid, 1997.
- Sitio Web de los productos de Rational.
- RUMBAUGH, J.; I. JACOBSON, et al. El Lenguaje Unificado del Modelado. Manual de Referencia. California, Addison Wesley, 1999.



### Glosario

**HTTP:** Protocolo de transferencia de hipertexto, es el método más común de intercambio de información en la world wide web, el método mediante el cual se transfieren las páginas web a un ordenador.

**FTP:** Protocolo de Transferencia de Archivos, utiliza los protocolos de Internet TCP/IP para permitir la transferencia de datos, un protocolo para transferir archivos hacia y desde computadoras remotas.

**SMTP:** Protocolo Simple de Traslferencia de Correo. Protocolo que se usa para transmitir correo electrónico entre servidores.

**ICMP:** Protocolo de control de mensajes de internet. Protocolo usado por el IP para informar de errores y excepciones.

**GUI:** Interfaz Gráfica de Usuario, una pantalla gráfica que permite al usuario interactuar con la computadora.

**Java:** Lenguaje de programación, similar al C++, que extiende las capacidades de Web.

**Software:** Palabra en inglés utilizada para indicar a los programas de computadoras, a las aplicaciones.

**TCP/IP:** Protocolo de control de transmisiones/Protocolo Internet. Es el protocolo estándar de comunicaciones en red utilizado para conectar sistemas informáticos a través de Internet.

**XP:** Programación Extrema, metodología de desarrollo de software basada en valores como simplicidad, comunicación y retroalimentación.

**Plugins:** Es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, generalmente muy específica, como por ejemplo servir como driver (controlador) en una aplicación, para hacer así funcionar un dispositivo en otro programa.

**Patrón Decorator:** Patrón de diseño que se encarga de decorar las responsabilidades de un objeto de forma dinámica y transparente a los clientes.

**OMT:** Modelado Técnico de Objeto, metodologías de análisis y diseño orientadas a objetos.

**OOSE:** Ingeniería de Software Orientada a Objeto. Metodología que se basa en el desarrollo de sistemas basado en el uso de distintos modelos.

**Logs:** Registro de actividad de un sistema, que generalmente se guarda en un fichero de texto, al que se le va añadiendo líneas a medida que se realizan acciones sobre el sistema.

**BSD:** Licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Pertenece al grupo de licencias de software Libre. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

**C++:** Lenguaje de programación orientado a objetos, basado en el lenguaje C.

**C#:** Es el nuevo lenguaje de propósito general orientado a objetos creado por Microsoft para su nueva plataforma .NET.

**ASCII:** Acrónimo del código estándar americano para el intercambio de información. Es el código estándar de conjunto de caracteres que cualquier ordenador puede entender, usado para representar las letras latinas, en mayúsculas, minúsculas, números, puntuación, etc.

**FDD:** Desarrollo Basado en Funcionalidades, es un proceso ágil para el desarrollo de sistemas, se preocupa por la calidad, por lo que incluye un monitoreo constante del proyecto.

**PT:** Plan de Trabajo.

**PA:** Programa Analítico.

**LDAP:** (Lightweight Directory Access Protocol) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

**SOAP:** ( Simple Object Access Protocol) es un protocolo estándar creado por Microsoft, IBM y otros, está actualmente bajo el auspicio de la W3C que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los servicios Web.

Anexos

Anexo # 1: Diagramas de Clases del Análisis.

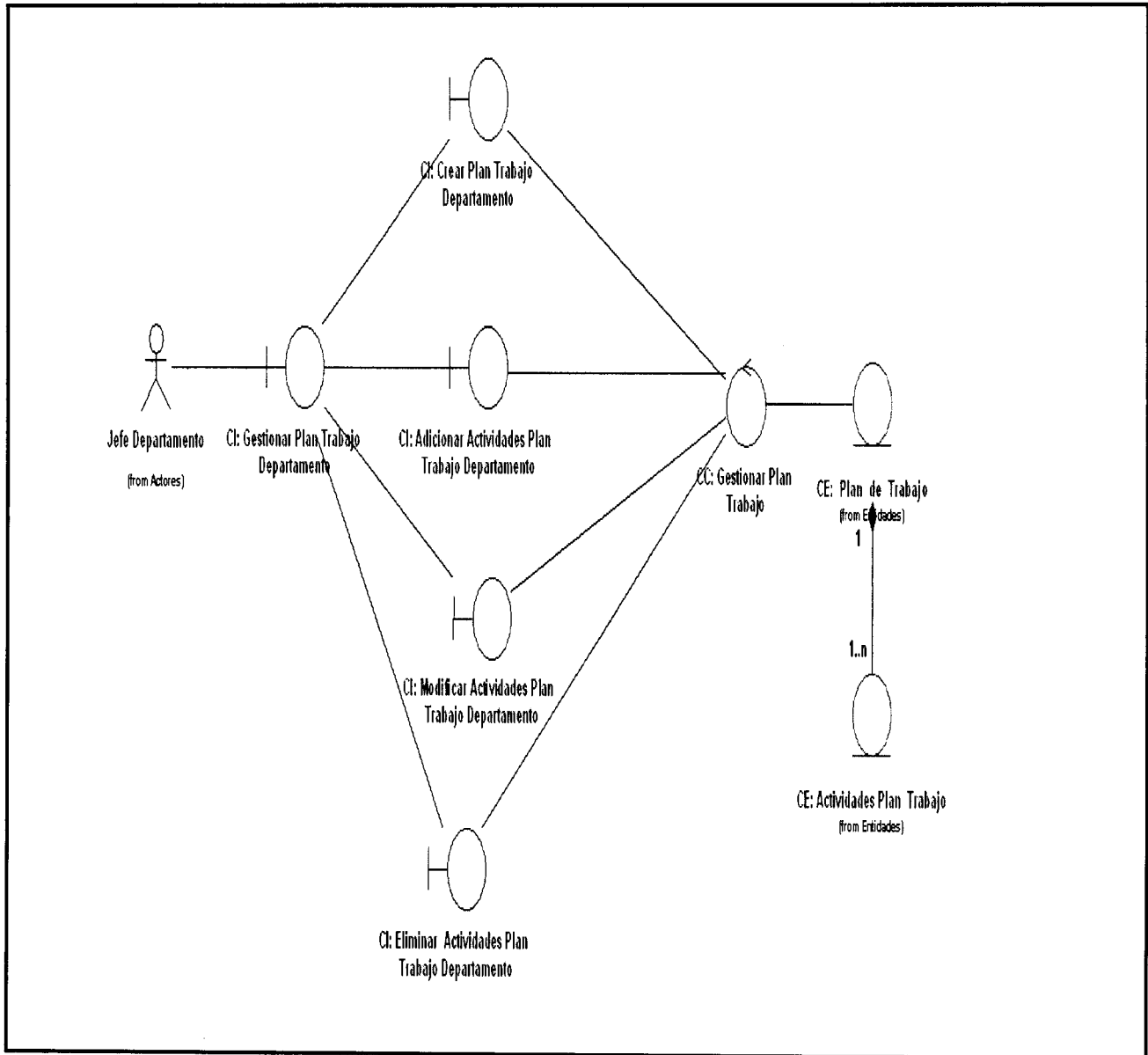


Figura # 20: Gestionar Plan de Trabajo del Departamento.

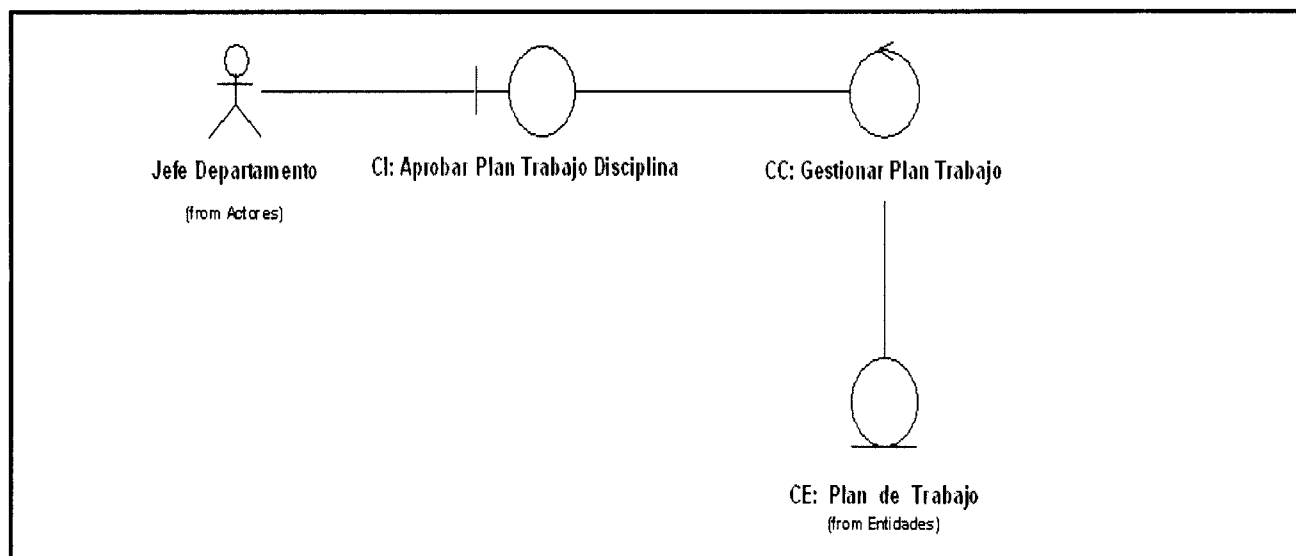


Figura # 21: Aprobar Plan de Trabajo Disciplina.

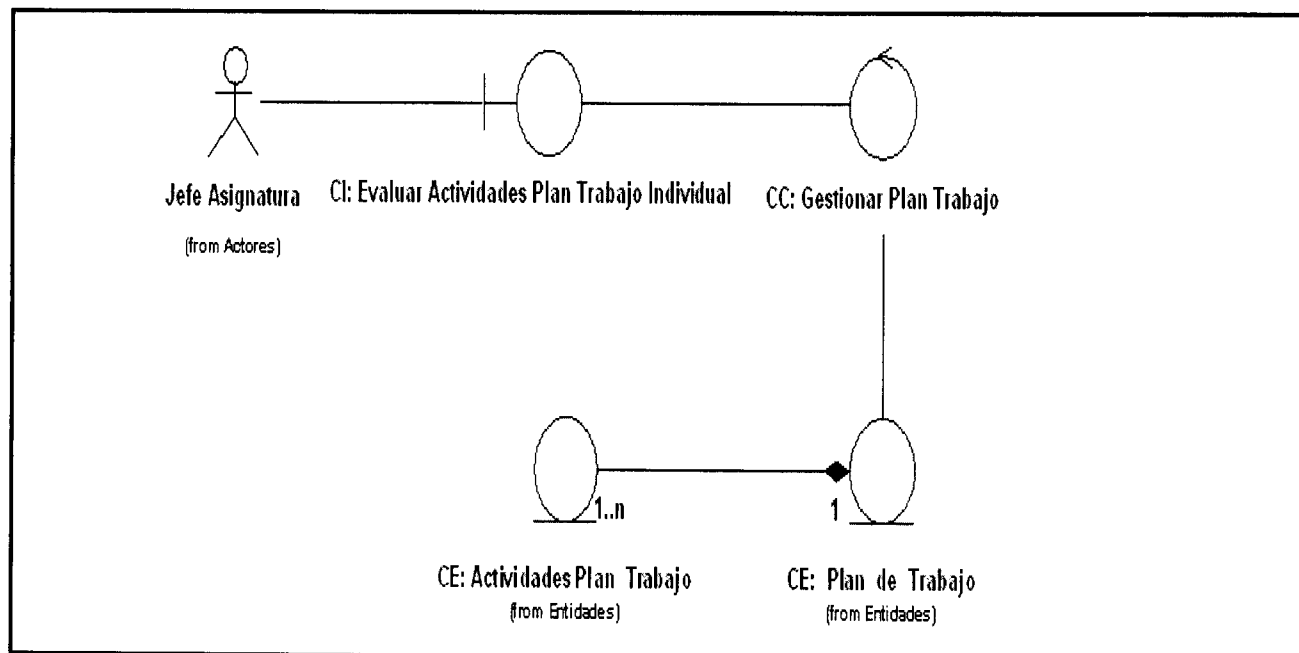


Figura # 22: Evaluar Actividades Plan de Trabajo Individual.

Anexo # 2: Diagramas de Interacción.

Gestionar Plan de Trabajo Departamento

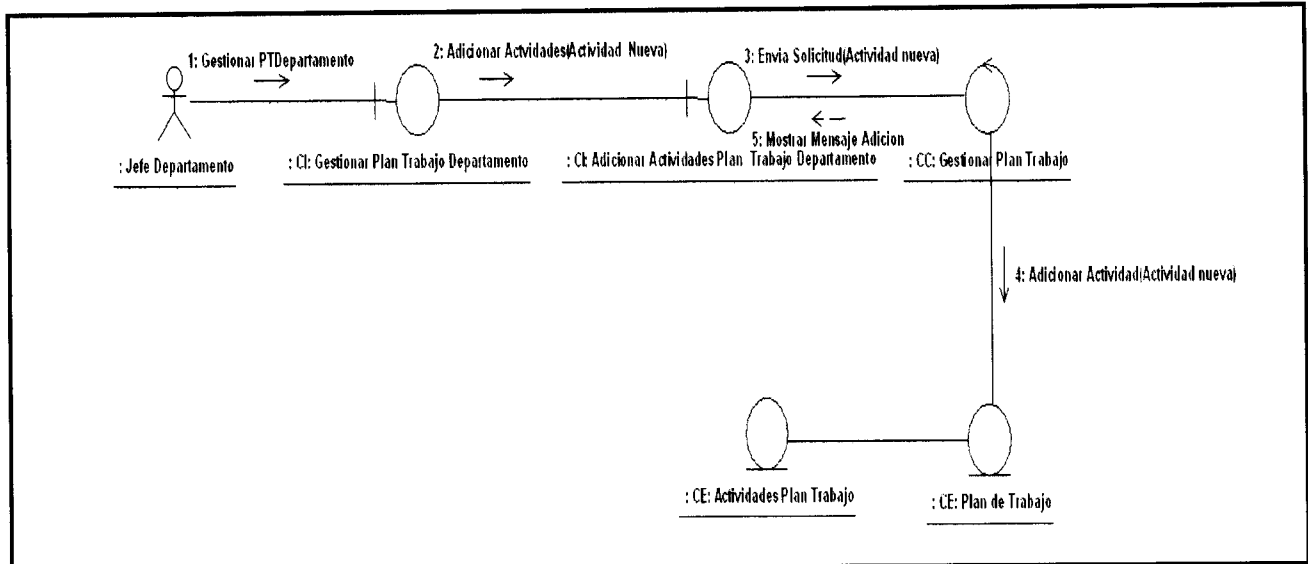


Figura # 23: Adicionar Actividades Plan de Trabajo Departamento (Gestionar Plan de Trabajo Departamento).

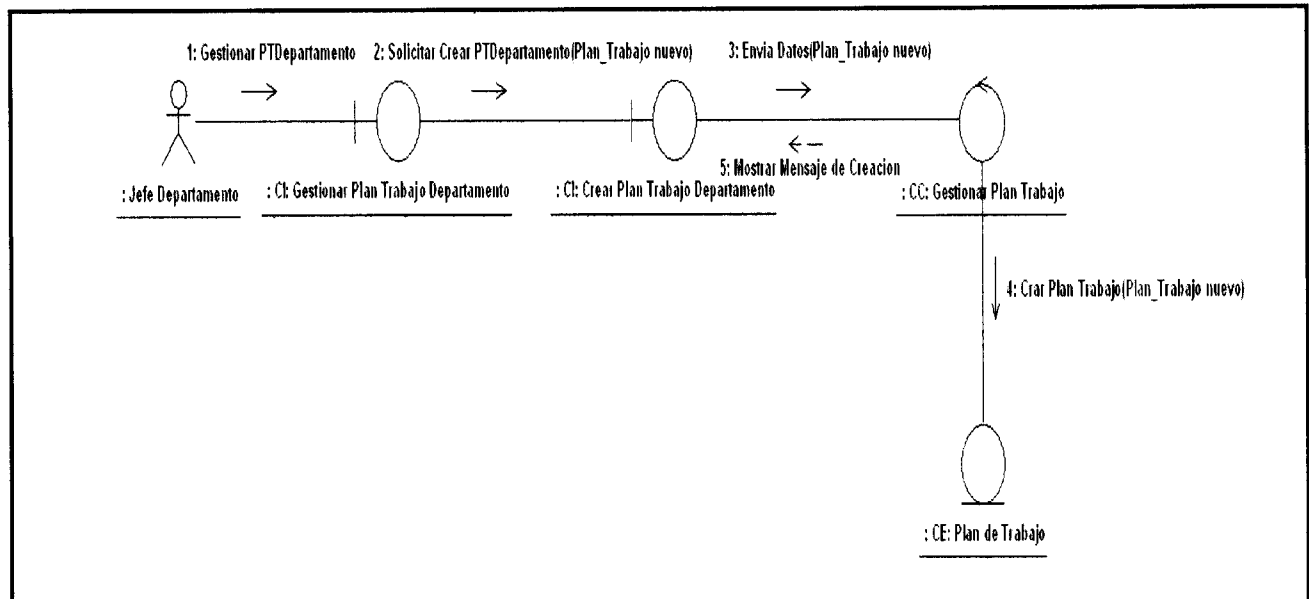


Figura # 24: Crear Plan de Trabajo Departamento (Gestionar Plan de Trabajo Departamento).

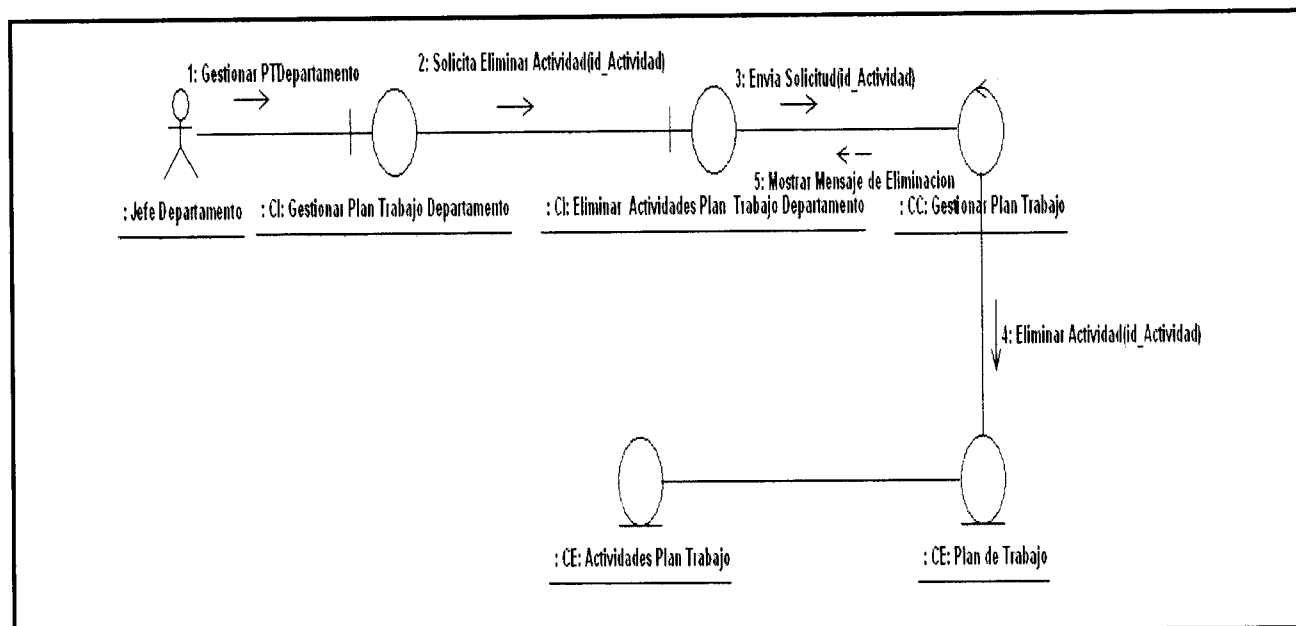


Figura # 25: Eliminar Actividades Plan de Trabajo Departamento (Gestionar Plan de Trabajo Departamento).

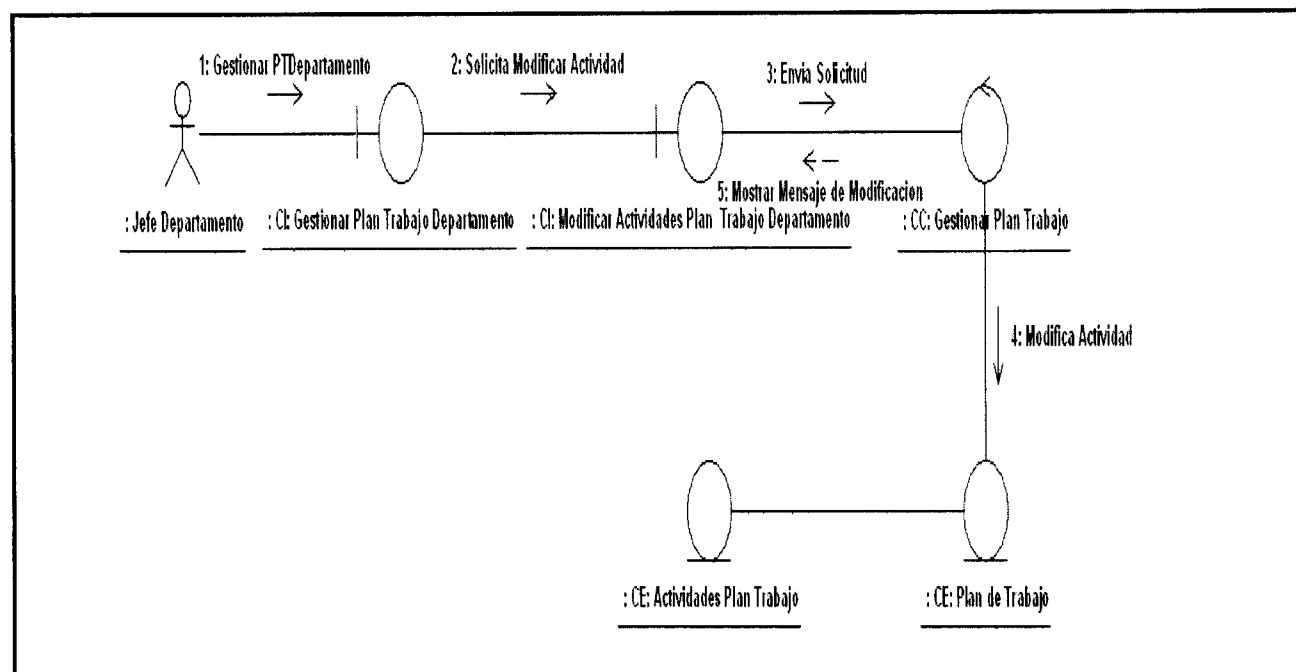


Figura # 26: Modificar Actividades Plan de Trabajo Departamento (Gestionar Plan de Trabajo Departamento).

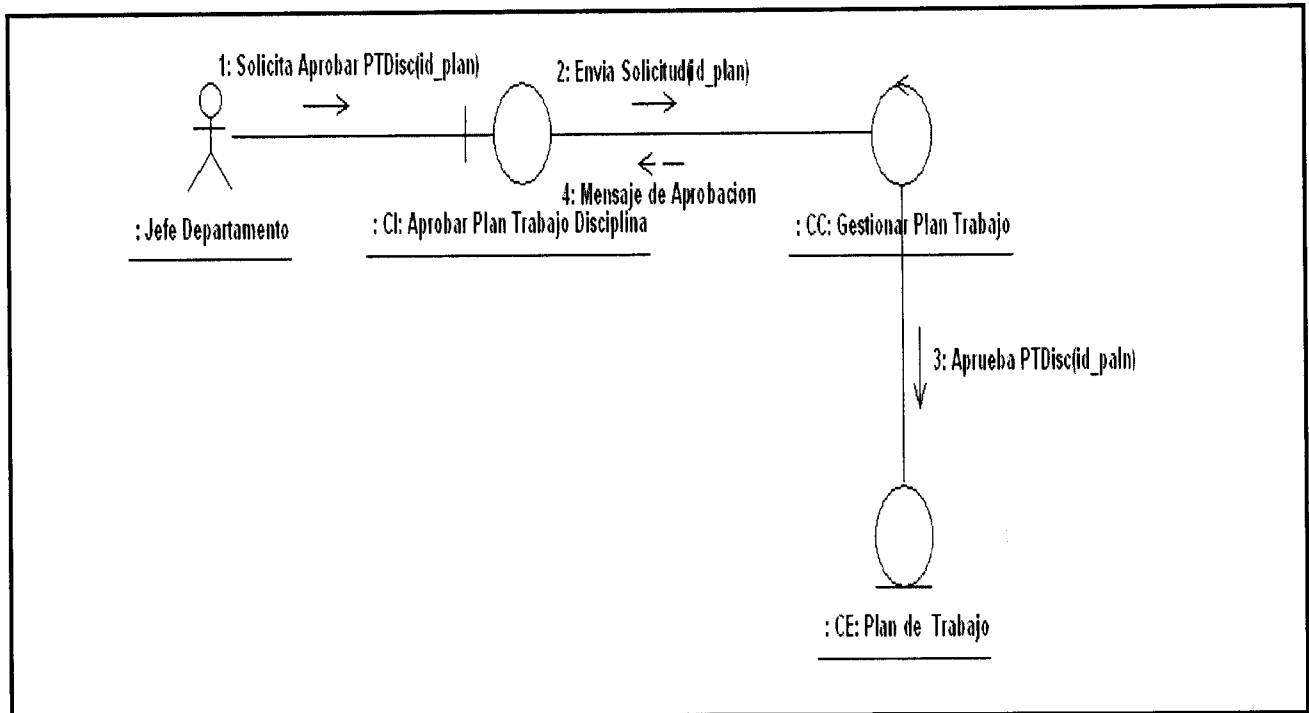


Figura # 27: Aprobar Plan de Trabajo Disciplina.

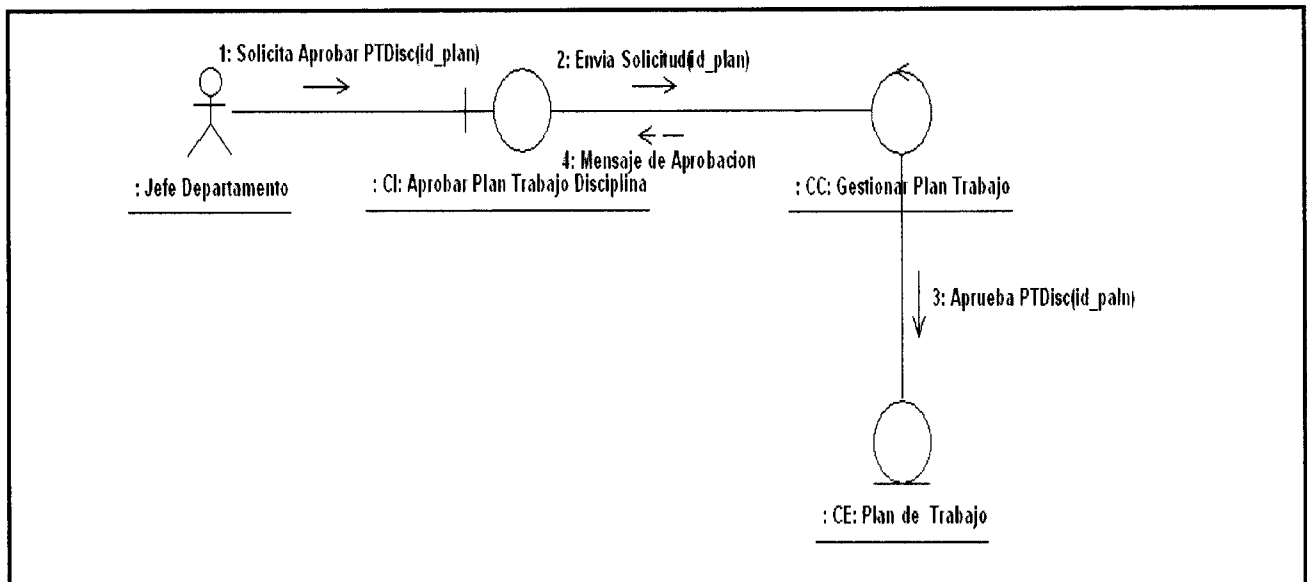


Figura # 28: Evaluar Actividades Plan de Trabajo Individual.

Anexo # 3: Diagramas de Componentes.

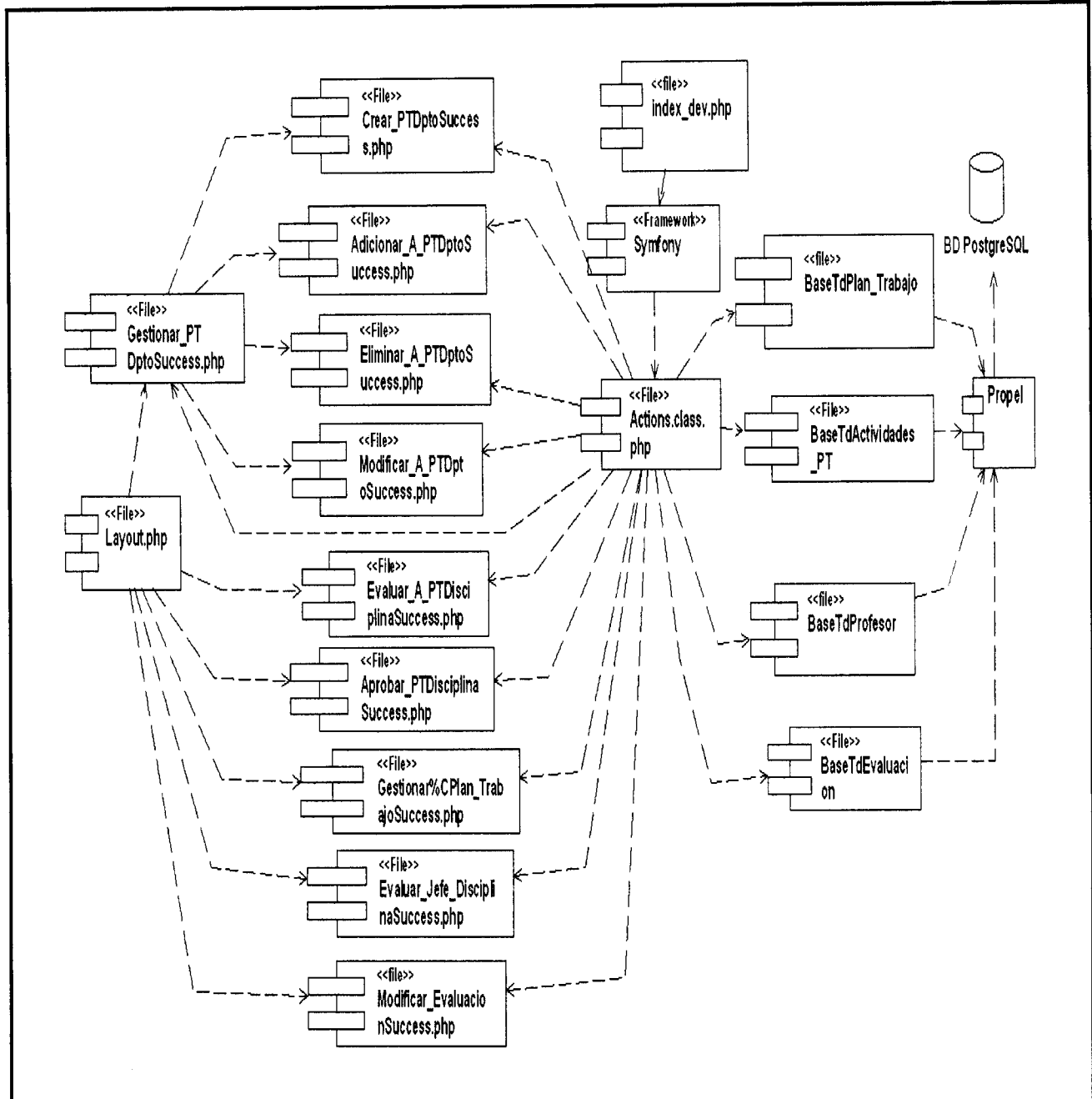


Figura # 29: Diagrama de Componentes Subsistema Jefe Departamento.



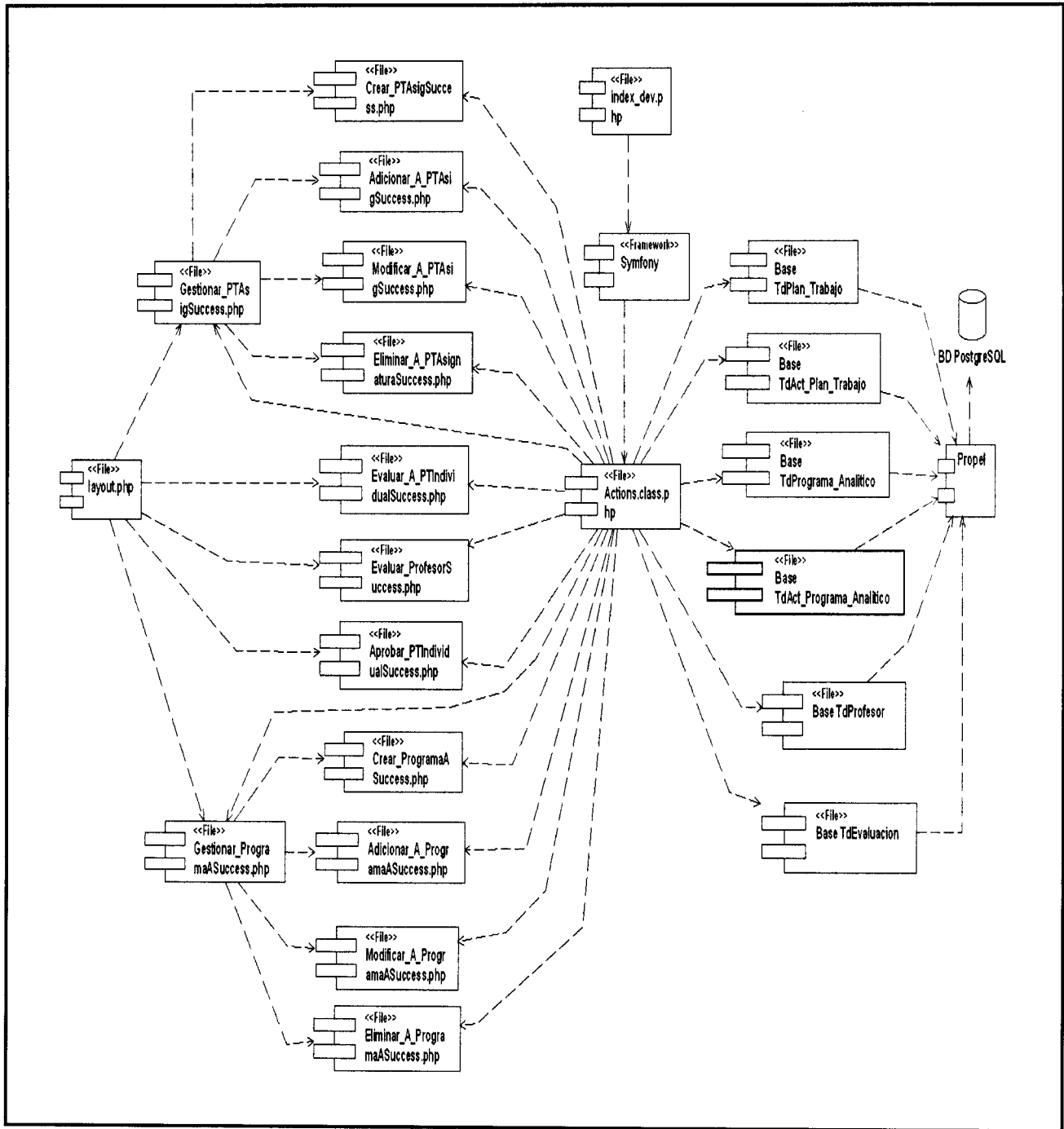


Figura # 30: Diagrama de Componentes Subsistema Jefe Asignatura.

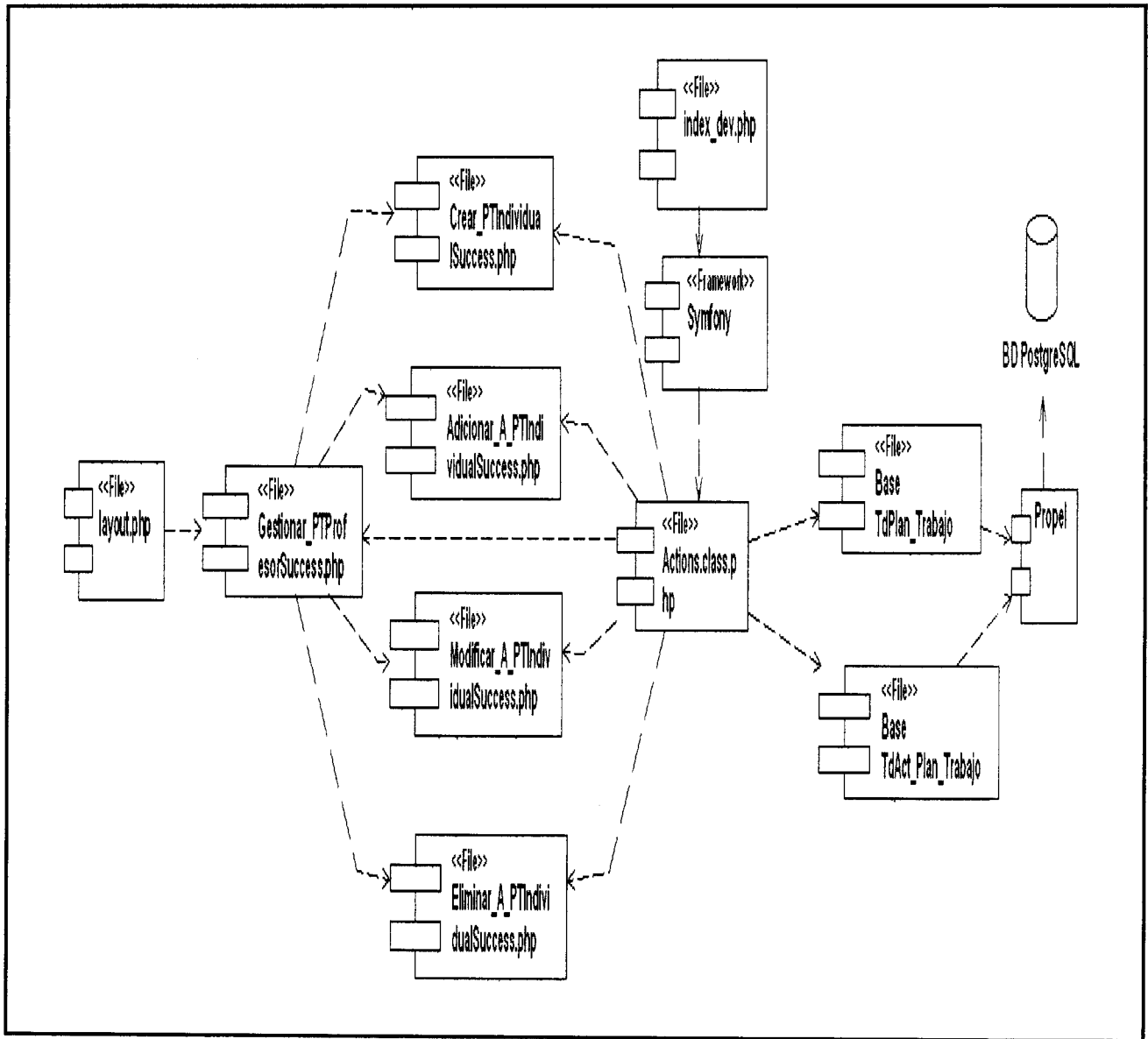


Figura # 31: Diagrama de Componentes Subsistema Profesor Regular.

Anexo # 4. Descripción de los Casos de Uso.

Tabla # 12. CUS Evaluar Actividades Plan Trabajo Individual.

Caso de Uso		Evaluar Actividades Plan Trabajo Individual.
<b>Actor:</b>	Jefe Asignatura (inicia)	
<b>Propósito:</b>	Permite al Jefe de Asignatura evaluar las actividades de los Planes de Trabajo individual de los Profesores que pertenecen a su Asignatura.	
<b>Resumen:</b>	El CUS se inicia cuando el Jefe del Asignatura selecciona la opción Evaluar Actividades Plan Trabajo Individual, el sistema le pide que seleccione el Plan de Trabajo Individual que desee evaluar, el Jefe Asignatura selecciona el Plan, el sistema le muestra los Plan de Trabajo Individual con sus respectivas actividades, luego el Jefe Asignatura evalúa cada actividad y el sistema adiciona cada evaluación en las actividades del Plan de Trabajo Individual y termina el CUS.	
<b>Referencias:</b>	RF 13.	
<b>Precondiciones:</b>	El Plan de Trabajo Individual seleccionado debe estar creado.	
<b>Poscondiciones:</b>		
<b>Curso Normal de Eventos:</b>		
<b>Acción del Actor:</b>	<b>Respuesta del Sistema:</b>	
1. El Jefe Asignatura selecciona la opción de Evaluar Actividades Plan Trabajo Individual.	1.1- El sistema verifica que sea el Jefe de Asignatura.	
2. El Jefe Asignatura selecciona el Plan de Trabajo Individual.	1.2- El sistema le pide al Jefe Asignatura que seleccione el Plan de Trabajo Individual que desea evaluar.	
	2.1- El sistema verifica que el Plan de Trabajo Individual seleccionado cuente con al menos actividad.	
3. El Jefe Asignatura Evalúa las Actividades.	2.2- El sistema muestra el Plan de Trabajo Individual con sus actividades.	

	3.1- El sistema actualiza la evaluación de las actividades finalizando el CUS.
<b>Curso alterno de los eventos:</b>	
<b>Acción 2.1:</b>	Si el Plan de Trabajo Individual seleccionado no tiene actividades creadas el sistema muestra un mensaje de error indicando que el Plan de Trabajo Individual no tiene actividades creadas y le indica al usuario retomar la acción 2.
<b>Prioridad:</b>	Crítico.

Tabla # 13. CUS Aprobar Plan Trabajo Individual.

Caso de Uso		Aprobar Plan Trabajo Individual.
<b>Actor:</b>	Jefe Asignatura (inicia)	
<b>Propósito:</b>	Permite al Jefe de Asignatura aprobar el Plan Trabajo Individual de cualquier Profesor que pertenezca a su Asignatura.	
<b>Resumen:</b>	El CUS se inicia cuando el Jefe de Asignatura selecciona la opción Aprobar Plan Trabajo Individual, el sistema le pide que seleccione el Plan de Trabajo Individual que desea aprobar, el Jefe Asignatura selecciona el plan, el sistema le muestra el Plan de Trabajo Individual con sus actividades, luego el Jefe Asignatura aprueba el Plan y el sistema actualiza el Plan de Trabajo Individual y termina el CUS.	
<b>Referencias:</b>	RF 16.	
<b>Precondiciones:</b>	El Plan Trabajo Individual debe estar creado.	
<b>Poscondiciones:</b>		
<b>Curso Normal de Eventos:</b>		
<b>Acción del Actor:</b>	<b>Respuesta del Sistema:</b>	
1. El Jefe Asignatura selecciona la opción Aprobar Plan Trabajo Individual.	1.1- El sistema verifica que sea el Jefe de Asignatura.	
	1.2- El sistema le pide al Jefe Asignatura que seleccione el Plan de Trabajo Individual que desea aprobar.	
2. El Jefe Asignatura selecciona el Plan de Trabajo del Individual.	2.1- El sistema verifica que el Plan de Trabajo	

3. El Jefe Asignatura aprueba el Plan de Trabajo Individual.	Individual seleccionado no este aprobado.  2.2- El sistema muestra el Plan de Trabajo Individual con sus actividades.  3.1- El sistema actualiza el Plan de Trabajo Individual finalizando el CUS.
<b>Curso alterno de los eventos:</b>	
<b>Acción 2.1:</b>	Si el Plan de Trabajo Individual seleccionado ya se encuentra aprobado el sistema muestra un mensaje de error indicando que el Plan de Trabajo Individual ya esta aprobado y le indica al usuario retomar la acción 2.
<b>Prioridad:</b>	Crítico.

Tabla # 14. CUS Evaluar Actividades Plan Trabajo Asignatura.

Caso de Uso		Evaluar Actividades Plan Trabajo Asignatura.
<b>Actor:</b>	Jefe Disciplina (inicia)	
<b>Propósito:</b>	Permite al Jefe de Disciplina evaluar las actividades de los Planes de Trabajo de las Asignaturas.	
<b>Resumen:</b>	El CUS se inicia cuando el Jefe del Disciplina selecciona la opción Evaluar Actividades Plan Trabajo de la Asignatura, el sistema le muestra los planes de trabajo de las Disciplinas con sus respectivas actividades, luego el Jefe Disciplina evalúa cada actividad y el sistema adiciona cada evaluación en las actividades del Plan de Trabajo de la Asignatura y termina el CUS.	
<b>Referencias:</b>	RF 9.	
<b>Precondiciones:</b>	El Plan de Trabajo de la Asignatura debe estar creado.	
<b>Poscondiciones:</b>		
<b>Curso Normal de Eventos:</b>		
<b>Acción del Actor:</b>	<b>Respuesta del Sistema:</b>	
1. El Jefe Disciplina selecciona la opción de Evaluar Actividades Plan Trabajo Asignatura.	1.1- El sistema verifica que sea el Jefe de Disciplina.	

<p>2. El Jefe Disciplina selecciona el Plan de Trabajo de la Asignatura.</p>	<p>1.2- El sistema le pide al Jefe Disciplina que seleccione el Plan de Trabajo de la Asignatura que desea evaluar.</p> <p>2.1- El sistema verifica que el Plan de Trabajo de la Asignatura seleccionado cuente con al menos actividad.</p>
<p>3. El Jefe Disciplina Evalúa las Actividades.</p>	<p>2.2- El sistema muestra el Plan de Trabajo de la Asignatura con sus actividades.</p> <p>3.1- El sistema Adiciona las actividades finalizando el CUS.</p>
<p><b>Curso alternativo de los eventos:</b></p>	
<p><b>Acción 2.1:</b></p>	<p>Si el Plan de Trabajo de la Asignatura seleccionado no tiene actividades creadas el sistema muestra un mensaje de error indicando que el Plan de Trabajo de la Asignatura no tiene actividades creadas y le indica al usuario retomar la acción 2.</p>
<p><b>Prioridad:</b></p>	<p>Crítico.</p>

Tabla # 15. CUS Gestionar Plan Trabajo Disciplina.

<p><b>Caso de Uso: Gestionar Plan Trabajo Disciplina.</b></p>	
<p><b>Actor:</b></p>	<p>Jefe Disciplina (inicia)</p>
<p><b>Propósito:</b></p>	<p>Permite al Jefe Disciplina gestionar (Modificar, Eliminar o Adicionar Actividades al PT Disciplina, además de crear el Plan de Trabajo de la Disciplina).</p>
<p><b>Resumen:</b></p>	<p>El CUS se inicia cuando el Jefe Disciplina selecciona la opción de Gestionar Plan Trabajo Disciplina, luego selecciona el tipo de gestión, introduce los datos necesarios, el sistema realiza la acción seleccionada por el Jefe del Disciplina y termina el CUS.</p>
<p><b>Referencias:</b></p>	<p>RF 7, RF 7.1, RF 7.2, RF 7.3, RF 7.4, RF 7.5</p>
<p><b>Precondiciones:</b></p>	<p>El Jefe de Disciplina debe pertenecer al Departamento.</p>

<p><b>Poscondiciones:</b></p>	<ol style="list-style-type: none"> <li>1. Información de las Actividades del Plan de Trabajo de la Disciplina adicionada a la Base de Datos.</li> <li>2. Información de las Actividades del Plan de Trabajo de la Disciplina modificadas a la Base de Datos.</li> <li>3. Información de las Actividades del Plan de Trabajo de la Disciplina eliminadas de la Base de Datos.</li> <li>4. Información del Plan de Trabajo de la Disciplina creado en la Base de Datos.</li> </ol>
<p><b>Curso Normal de Eventos:</b></p>	
<p><b>Acción del Actor:</b></p>	<p><b>Respuesta del Sistema:</b></p>
<ol style="list-style-type: none"> <li>1. El Jefe Disciplina selecciona la opción de Gestionar Plan Trabajo Disciplina.</li> </ol>	<ol style="list-style-type: none"> <li>1.1- El sistema verifica que sea el Jefe de Disciplina.</li> <li>1.2- El sistema muestra las opciones: Crear Plan Trabajo Disciplina, Adicionar Actividades Plan Trabajo Disciplina, Modificar Actividades Plan Trabajo Disciplina y Eliminar Actividades Plan Trabajo Disciplina.</li> </ol>
<p><b>Escenario 1: Crear Plan Trabajo Disciplina.</b></p>	
<ol style="list-style-type: none"> <li>1. El Jefe Disciplina selecciona la opción de Crear Plan Trabajo Disciplina.</li> <li>2. El Jefe Disciplina introduce los datos solicitados por el sistema.</li> </ol>	<ol style="list-style-type: none"> <li>1.1- El sistema muestra el formulario a completar para la creación del nuevo Plan de Trabajo de la Disciplina.</li> <li>2.1- El sistema verifica los datos introducidos por el Jefe Disciplina.</li> <li>2.2- Si los datos introducidos por el Jefe Disciplina son correctos el sistema adiciona dicha</li> </ol>

	información en la Base de Datos correspondiente y termina el CUS.
<b>Curso alterno de los eventos:</b>	
<b>Acción 2.2:</b>	Si los datos introducidos por el Jefe Disciplina son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al usuario retornar a la acción 2.
<b>Escenario 2: Adicionar Actividades Plan Trabajo Disciplina.</b>	
1. El Jefe Disciplina selecciona la opción de Adicionar Actividades Plan Trabajo Disciplina.  2. El Jefe Disciplina introduce los datos solicitados por el sistema.	1.1- El sistema muestra el formulario a completar para adicionar actividades al Plan de Trabajo de la Disciplina.  2.1- El sistema verifica los datos introducidos por el Jefe Disciplina.  2.2- Si los datos introducidos por el Jefe Disciplina son correctos el sistema adiciona dicha información en la Base de Datos correspondiente y termina el CUS.
<b>Curso alterno de los eventos:</b>	
<b>Acción 2.1:</b>	Si los datos introducidos por el Jefe por el Jefe Disciplina son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al usuario retornar a la acción 2.
<b>Escenario 3: Modificar Actividades Plan Trabajo Disciplina.</b>	
1. El Jefe Disciplina selecciona la opción de Modificar Actividades Plan Trabajo Disciplina.  2. El Jefe Disciplina realiza las modificaciones a las Actividades del Plan Trabajo Disciplina.	1.1- El sistema muestra el Plan de Trabajo del Disciplina con sus Actividades al cual el Jefe Disciplina tiene acceso.  2.1- El sistema verifica los datos modificados por el Jefe Disciplina.  2.2- Si los datos modificados por el Jefe por el Jefe Disciplina son correctos el sistema adiciona dicha información en la Base de Datos correspondiente y termina el CUS.
<b>Curso alterno de los eventos:</b>	



<b>Acción 2.1:</b>	Si los datos modificados por el Jefe por el Jefe Disciplina son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al usuario retornar a la acción 2.
<b>Escenario 4: Eliminar Actividades Plan Trabajo Disciplina.</b>	
1. El Jefe Disciplina selecciona la opción de Eliminar Actividades Plan Trabajo Disciplina.	1.1- El sistema muestra las actividades del Plan de Trabajo de la Disciplina a las cuales el Jefe Disciplina tiene acceso.
2. El Jefe Disciplina selecciona la actividad que desea eliminar.	2.1- El sistema muestra un mensaje de advertencia para la acción a realizar.
3. El Jefe Disciplina confirma si quiere o no eliminar la actividad seleccionada.	3.1- Si el Jefe Disciplina acepta el sistema elimina la actividad del Plan de Trabajo de la Disciplina y culmina el CUS.
<b>Curso alternativo de los eventos:</b>	
<b>Acción 3.1:</b>	Si el Jefe Disciplina cancela la acción se culmina el CUS sin ejecutar ninguna acción.
<b>Prioridad:</b>	Crítico.

Tabla # 16. CUS Evaluar Jefe Asignatura.

Caso de Uso	Evaluar Jefe Asignatura.
<b>Actor:</b>	Jefe Disciplina (inicia)
<b>Propósito:</b>	Permite al Jefe de Disciplina evaluar a los Jefes de Asignatura que pertenezcan a su Disciplina.
<b>Resumen:</b>	El CUS se inicia cuando el Jefe de Disciplina selecciona la opción Evaluar Jefe Asignatura, el sistema le pide que inserte el solapín y el intervalo de fecha de evaluación del Profesor que desea evaluar, el Jefe de Disciplina inserta el solapín el intervalo de fecha de evaluación, el sistema verifica el por ciento de cumplimiento de sus actividades y le asigna una evaluación y termina el CUS.
<b>Referencias:</b>	RF 8.
<b>Precondiciones:</b>	El Plan Trabajo de la Asignatura que debe realizar el Jefe de Asignatura

	debe estar creado y que al menos tenga una actividad esté evaluado.
<b>Poscondiciones:</b>	
<b>Curso Normal de Eventos:</b>	
<b>Acción del Actor:</b>	<b>Respuesta del Sistema:</b>
<p>1. El Jefe Disciplina selecciona la opción Evaluar Jefe Asignatura.</p> <p>2. El Jefe Disciplina inserta el usuario del Profesor y el intervalo de fecha de evaluación.</p>	<p>1.1- El sistema verifica que sea el Jefe de Disciplina.</p> <p>1.2- El sistema le pide al Jefe de Disciplina que entre el usuario del Profesor y el intervalo de fecha de evaluación del Jefe de Asignatura.</p> <p>2.1- El sistema verifica que el Jefe de Asignatura pertenezca a la Disciplina que dirige el Jefe de Disciplina.</p> <p>2.2- El sistema verifica el porcentaje del cumplimiento de las actividades del Plan de Trabajo de la Asignatura correspondiente al Jefe de Asignatura que seleccionó.</p> <p>2.3- El sistema le asigna una evaluación al Jefe de Asignatura finalizando el CUS.</p>
<b>Curso alternativo de los eventos:</b>	
<b>Acción 2.1:</b>	Si el Jefe Asignatura seleccionado no pertenece a la Disciplina del Jefe de Disciplina, el sistema muestra un mensaje de error indicando que el Jefe de Asignatura seleccionado no pertenece a la Disciplina que dirige el Jefe de Disciplina y le indica al usuario retomar la acción 2.
<b>Prioridad:</b>	Crítico.

**Anexo # 5.** Entrevista realizada al cliente M.Sc. Febe Angel Ciudad Ricardo. [27 de noviembre de 2007]

1. ¿Cómo funciona en la actualidad el flujo de información que se manejan en los Departamentos Docentes de la Facultad 9?

El flujo de las estadísticas en estos momentos se realiza de forma manual, los documentos que posibilitan que el trabajo de los Departamento Docentes fluyan correctamente en algunas ocasiones no son actualizados ya que no se cuenta con un sistema que posibilite la actualización y tratamiento de los mismos.

2. ¿Cómo se encuentran organizados los departamentos docentes de la facultad?

Los Departamentos Docentes cuentan para un buen funcionamiento con 5 Departamentos, los cuales son: Departamento de Humanidades, Departamento de Ciencias Básicas, Departamento de Programación, Departamento de Sistemas Digitales y el Departamento de Ingeniería de Software y Práctica Profesional. Cada Departamento esta compuesto por un Jefe de Departamento, varios Jefes de Disciplina y Asignaturas y los demás profesores que pertenecen al mismo.

3. ¿Cuáles son las actividades principales que realizan los profesores en la facultad?

En el Departamento están definidas las realizaciones de las actividades según las categorías docentes de los profesores, por ejemplo; el Jefe de Departamento es el encargado de gestionar el plan de trabajo del departamento, es el que evalúa a los Jefes de Disciplina que pertenecen a su departamento, como máxima autoridad en el Departamento Docente tiene la posibilidad de modificar la evaluación de cualquier profesor que pertenezca a su departamento. Para ver como va marchando el cumplimiento de los planes de trabajo, es el encargado de realizar reportes a cualquier profesor de su departamento durante el curso. Los Jefes de Disciplina tienen la responsabilidad de gestionar el plan de trabajo de la disciplina, es el encargado de evaluar a los Jefes de Asignatura que pertenecen a su disciplina. El Jefe de Asignatura es el encargado de realizar el plan de trabajo de la asignatura, es el responsable de evaluar a los profesores que pertenecen a su asignatura y además gestionar el programa analítico de la asignatura. El profesor regular sólo puede gestionar su plan de trabajo individual.