

005.269
VEL
S
TD-0155-06

TD-0155-06



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



INSTITUTO SUPERIOR POLITÉCNICO "JOSE ANTONIO ECHEVERRIA"
CENTRO DE ESTUDIOS DE INGENIERÍA Y SISTEMAS
FACULTAD DE INGENIERÍA INDUSTRIAL

SISTEMA DE GESTIÓN METODOLÓGICA CENTRAL DE LA UCI

Trabajo de diploma para optar por el título de
Ingeniería en Informática

Autor: Yunier Velázquez Batista

Tutor: MSc. Yamilis Fernández Pérez

Ciudad de La Habana, Cuba
Junio, 2006

AGRADECIMIENTOS

...A mi abuela Iraelia por haberme mostrado el camino correcto de la vida.

...A mi Mamá por confiar en mí y darme todo su apoyo. A mí Papá que aunque no esta entre nosotros me dio la oportunidad de vivir esta experiencia. A toda mi familia - tías, tío, primos, primas- que siempre han confiado en mi.

...A mi querida Ary por su preocupación y cariño.

...A mi tutora Yamilis por su empeño, esfuerzo y responsabilidad.

...A mis amigos y compañeros Vlair, Maykel, Alian, Yobannys, Guillermo que me dieron su apoyo y siempre estuvieron ahí cuando me hicieron falta.

...A mi coautor Yanier que a pesar de estar complicado con su trabajo me ayudó en todo lo que pudo.

...A Chony, Jose y Julito que me soportaron durante este tiempo en el Apto.

...A Ailec y Bety por preocupación y ayuda.

...A todos mis profesores de Camagüey que fueron un peldaño en la carrera.

...A la Revolución por darme la oportunidad de realizar un sueño.

En fin, a todos aquellos que de una forma u otra han contribuidos con mi formación.

A todos, GRACIAS.

DEDICATORIA

Dedico este trabajo a mi familia en general, principalmente a mi abuela Iraelia, a mi mamá Nancy, a mi difunto padre que le hubiese gustado mucho haberme visto graduado. Se lo dedico además a mis hermanos Isdel, Carlos Alexis y Juan Manuel.

Finalmente me lo dedico a mi mismo por el esfuerzo realizado en estos últimos 5 años de estudio y esfuerzo.

RESUMEN

El trabajo de los departamentos docentes centrales en la Universidad de las Ciencias Informáticas (UCI) es de vital importancia en el proceso docente metodológico. La labor fundamental de los mismos está encaminada a elevar la calidad del proceso de formación del profesional y el trabajo metodológico del claustro. En los departamentos se realizan una serie de tareas y asociadas a ellas hay un gran flujo de información que debe ser de un seguro y rápido acceso. Actualmente muchas de estas tareas se llevan a cabo de forma manual con ayuda de máquinas computadoras, pero que no cumplen ciertos requisitos de seguridad y control, produciendo pérdidas de tiempo y recursos.

Por tanto el objetivo concreto de este trabajo consiste en definir una arquitectura para una aplicación Web, segura y de interfaz amigable, que facilite el control y la gestión de los procesos fundamentales de un departamento docente central con el desarrollo de su prototipo funcional. Cumplimentando este objetivo surge el Sistema de Gestión Metodológica Central de la UCI, que automatiza la gestión de los procesos de conformación del claustro, planeación y control de planes de trabajo metodológico, la solicitud de materiales, la confección de la prenómina de pago y aseo entre otros. Siendo esta una herramienta para los directivos de los departamentos, jefes de colectivos y profesores en el proceso metodológico

Este documento recoge los resultados de todo el trabajo realizado. Se estudian las características esenciales de los departamentos docentes centrales y se realiza un análisis de las tendencias y tecnologías existentes seleccionándose las más apropiadas para desarrollar la aplicación. Además, se muestran los resultados de los flujos de trabajo de Modelo de Negocio, levantamiento de requerimientos, y análisis y diseño de la propuesta del sistema, llegando hasta una primera iteración de la fase de elaboración. Finalmente se concluye, y se recomiendan algunos aspectos para el mejoramiento futuro del mismo.

ÍNDICE

INTRODUCCIÓN.....	1
1 FUNDAMENTACIÓN TEÓRICA.....	4
1.1 INTRODUCCIÓN	4
1.2 OBJETO DE ESTUDIO.....	4
1.2.1 <i>Objetivos estratégicos de la organización</i>	5
1.2.2 <i>Flujo actual de los procesos</i>	6
1.2.3 <i>Análisis crítico de la ejecución de los procesos</i>	7
1.3 PROCESOS OBJETO DE AUTOMATIZACIÓN.....	8
1.4 SISTEMAS AUTOMATIZADOS EXISTENTES VINCULADOS AL CAMPO DE ACCIÓN	9
1.4.1 <i>Justificación</i>	9
1.5 FUNDAMENTACIÓN DE LOS OBJETIVOS.....	10
1.5.1 <i>Objetivo general</i>	10
1.5.2 <i>Objetivos específicos</i>	10
1.6 TENDENCIAS Y TECNOLOGÍAS ACTUALES.....	10
1.6.1 <i>Política de migración a software libre</i>	10
1.6.2 <i>PHP</i>	11
1.6.3 <i>Plataforma de desarrollo Web</i>	12
1.6.4 <i>Drupal</i>	12
1.6.5 <i>Zend Studio</i>	14
1.6.6 <i>SQL</i>	14
1.6.7 <i>Sistema gestor de bases de datos MySQL</i>	14
1.6.8 <i>El Proceso Unificado de Modelado (RUP)</i>	15
1.6.9 <i>Diseño Arquitectónico</i>	16
1.6.10 <i>Selección de las tecnologías a utilizar en la propuesta de solución</i>	19
1.7 CONCLUSIONES.....	19
2 MODELO DEL NEGOCIO.....	20
2.1 INTRODUCCIÓN	20
2.2 MODELO DEL NEGOCIO ACTUAL.....	20
2.3 REGLAS DEL NEGOCIO A CONSIDERAR	22
2.4 ACTORES DEL NEGOCIO	22
2.5 DIAGRAMA DE CASOS DE USO DEL NEGOCIO	23
2.6 TRABAJADORES DEL NEGOCIO	24
2.7 CASOS DE USO DEL NEGOCIO.....	24
<i>Tabla 3. Caso de uso: Confeccionar Plan de Trabajo Metodológico (PTM)</i>	24

<i>Tabla 4 Caso de uso: Confeccionar Plan de Trabajo de Asesor</i>	27
<i>Tabla 5. Caso de uso: Confeccionar Plantilla del Departamento</i>	30
<i>Tabla 6. Caso de uso: Realizar Control a Clases</i>	32
<i>Tabla 7. Caso de uso: Solicitar Materiales</i>	35
<i>Tabla 8. Caso de uso: Confeccionar Prenómina y Aseo</i>	38
<i>Tabla 9. Caso de uso: Recoger Información</i>	40
2.8 MODELO DE OBJETOS	43
FIGURA 16. MODELO DE OBJETOS	44
2.9 CONCLUSIONES	44
3 REQUISITOS	45
3.1 INTRODUCCIÓN	45
3.2 ACTORES DEL SISTEMA A AUTOMATIZAR	45
3.3 DEFINICIÓN DE LOS REQUISITOS FUNCIONALES	46
3.4 DIAGRAMA DE CASOS DE USO DEL SISTEMA A AUTOMATIZAR	50
3.5 DEFINICIÓN DE LOS REQUISITOS NO FUNCIONALES	52
3.6 DESCRIPCIÓN DE LOS CASOS DE USO	53
3.7 CONCLUSIONES	60
4 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA	61
4.1 INTRODUCCIÓN	61
4.2 DESCRIPCIÓN DE LA ARQUITECTURA	61
4.3 DIAGRAMA DE CLASES DEL DISEÑO	65
4.3.1 <i>Paquete <Autenticación></i>	67
4.3.2 <i>Paquete <Prenómina y Aseo></i>	67
4.3.3 <i>Paquete <Solicitud de Materiales></i>	68
4.3.4 <i>Paquete <Control a Clases></i>	69
4.3.5 <i>Paquete <Plan de Trabajo Metodológico></i>	70
4.3.6 <i>Paquete <Plan de Trabajo Individual></i>	71
<i>Figura 26. Diagrama de clases web del paquete < Plan de Trabajo Individual ></i>	71
4.3.7 <i>Paquete <Plantilla ></i>	72
4.4 PRINCIPIOS DE DISEÑO	73
4.4.1 <i>Interfaz de usuario</i>	73
4.5 TRATAMIENTO DE ERRORES	74
4.6 DISEÑO DE LA BASE DE DATOS	76
4.6.1 <i>Modelo lógico de datos</i>	76
4.6.2 <i>Modelo físico de datos</i>	77
4.7 DIAGRAMA DE DESPLIEGUE	78

4.8	CONCLUSIONES.....	78
5	ESTUDIO DE FACTIBILIDAD.....	80
5.1	INTRODUCCIÓN.....	80
5.2	ESTIMACIÓN DE COSTO.....	80
5.2.1	<i>Planificación</i>	81
5.2.2	<i>Costos</i>	85
5.3	BENEFICIOS TANGIBLES E INTANGIBLES.....	87
5.4	ANÁLISIS DE COSTOS Y BENEFICIOS.....	88
	CONCLUSIONES.....	89
	RECOMENDACIONES.....	90
	REFERENCIAS BIBLIOGRÁFICAS.....	91
	BIBLIOGRAFÍA.....	92
	GLOSARIO DE TÉRMINOS.....	I

ÍNDICE DE TABLAS

Tabla 1. Descripción de los actores del negocio22

Tabla 2. Descripción de los trabajadores del negocio24

Tabla 3. Caso de uso: Confeccionar Plan de Trabajo Metodológico (PTM).24

Tabla 4 Caso de uso: Confeccionar Plan de Trabajo de Asesor.....27

Tabla 5. Caso de uso: Confeccionar Plantilla del Departamento.30

Tabla 6. Caso de uso: Realizar Control a Clases.....32

Tabla 7. Caso de uso: Solicitar Materiales.....35

Tabla 8. Caso de uso: Confeccionar Prenómina y Aseo.....38

Tabla 9. Caso de uso: Recoger Información.40

Tabla 10. Definición de actores del sistema a automatizar45

Tabla 11. Confeccionar Plan de Trabajo Metodológico53

Tabla 12. Evaluar Plan de Trabajo Metodológico.....54

Tabla 13. Gestionar actividad.....54

Tabla 14. Confeccionar plan de trabajo individual.....55

Tabla 15. Evaluar asesor.55

Tabla 16. Mostrar plan de trabajo individual.55

Tabla 17. Confeccionar plantilla de la disciplina.....56

Tabla 18. Confeccionar plantilla de Colectivo.....56

Tabla 19. Mostrar profesores.57

Tabla 20. Planificar controles a clases.57

Tabla 21. Mostrar resultado de controles a clases.....58

Tabla 22. Realizar solicitud de materiales.....58

Tabla 23. Confeccionar prenómina y aseo.58

Tabla 24. Gestionar noticias.....59

Tabla 25. Autenticar usuario.....59

Tabla 27. Entradas Externas.81

Tabla 28. Salidas Externas.....82

Tabla 29. Peticiones.....83

Tabla 30. Ficheros Internos.84

Tabla 31. Interfaces Externas.85

Tabla 32. Cantidad de instrucciones fuentes.....85

Tabla 33. Factores de escala.85

ÍNDICE DE FIGURAS

Figura 1. Diagrama de casos de uso del negocio..... 23

Figura 2. Diagrama de actividad del caso de uso: Confeccionar Plan de Trabajo Metodológico..... 26

Figura 3. Modelo de objeto del caso de uso: Confeccionar Plan de Trabajo Metodológico..... 27

Figura 4. Diagrama del caso de uso: Confeccionar Plan de Trabajo de Asesor. 29

Figura 5. Modelo de objeto del caso de uso: Confeccionar Plan de Trabajo de Asesor. 30

Figura 6. Diagrama de actividad del caso de uso: Confeccionar Plantilla del Departamento. 32

Figura 7. Modelo de objeto del caso de uso: Confeccionar Plantilla del Departamento. 32

Figura 8. Diagrama de actividad del caso de uso: Realizar Control a Clases. 34

Figura 9. Modelo de objeto del caso de uso: Realizar Control a Clases. 35

Figura 10. Diagrama de actividad del caso de uso: Solicitar Materiales. 37

Figura 11. Modelo de objeto del caso de uso: Solicitar Materiales. 38

Figura 12. Diagrama de actividad del caso de uso: Confeccionar Prenómina y Aseo. . 39

Figura 13. Modelo de objeto del caso de uso: Confeccionar Prenómina y Aseo. 40

Figura 14. Diagrama de actividad del caso de uso: Recoger Información. 42

Figura 15. Modelo de objeto del caso de uso: Recoger Información. 43

Figura 16. Modelo de objetos. 44

Figura 17. Diagrama del casos de uso del sistema. 51

Figura 18. Arquitectura de Drupal. 62

Figura 19. Estructura de paquetes del sistema. 65

Figura 20. Diagrama de paquetes del subsistema GMC. 66

Figura 21. Diagrama de clases web del paquete <Autenticación>..... 67

Figura 22. Diagrama de clases web del paquete <Prenómina y Aseo>..... 67

Figura 23. Diagrama de clases web del paquete <Solicitud de Materiales>. 68

Figura 24. Diagrama de clases web del paquete <Control a Clases> 69

Figura 25. Diagrama de clases web del paquete <Plan de Trabajo Metodológico>..... 71

Figura 26. Diagrama de clases web del paquete < Plan de Trabajo Individual >..... 71

Figura 27. Diagrama de clases web del paquete <Plantilla>. 72

ÍNDICE DE FIGURAS

Figura 17. Esquema general del diseño.....	74
Figura 29. Validación de formularios.....	75
Figura 30. Confirmación de eliminación.....	75
Figura 31. Diagrama de Clases Persistentes.....	76
Figura 32. Diagrama de Base de Datos.....	77
Figura 33. Diagrama de despliegue.....	78

INTRODUCCIÓN

En la universidad se forja el profesional de nuestra sociedad, el cual contribuye a impulsar su desarrollo, de ahí la importancia de la formación del mismo. Por ello es necesario lograr una buena organización y control del proceso metodológico, que constituye el alma de los centros de educación superior. En la actualidad el sistema de educación realiza un arduo trabajo en este sentido, con la finalidad de lograr una mayor eficiencia en la planificación, y elevar la calidad de los resultados.

En la Universidad de las Ciencias de Informática (UCI) la responsabilidad del proceso docente metodológico recae en la Vicerrectoría de Formación, específicamente en la Dirección de Formación del Profesional y en los Departamentos Docentes Centrales, que son los encargados de organizar y dirigir las actividades metodológicas.

En el Departamento Docente Central de Ingeniería y Gestión de Software de la UCI no existe un control eficiente de la información en el proceso docente metodológico, y laboral. Esto trae como consecuencia que se dificulte y demore el acceso a la información actual, pues esta es procesada manualmente, lo que hace más engorroso el trabajo del jefe de departamento y asesores, en el chequeo del cumplimiento de las actividades docente, y planeación metodológica.

Para dar solución a estos problemas se decide realizar un sistema informático que garantice la seguridad y consistencia de la información para ayudar a directivos en la toma de decisiones. Así surge el Sistema de Gestión Metodológica Centrales de la UCI. Para la implementación del mismo se escogió desarrollar una aplicación Web, de manera que los usuarios tengan acceso al sistema desde cualquier lugar de la UCI.

El objeto de estudio radica en la gestión de información asociada a los procesos de dirección y control metodológico del Departamento Docente Central de Ingeniería y Gestión de Software de la UCI, así como el desarrollo de una aplicación y diseño de base de datos que permitan su automatización.

El campo de acción de este trabajo se centra en el estudio de las actividades metodológicas que planifica y ejecuta el Departamento Docente Central de Ingeniería y Gestión de Software.

Como hipótesis se plantea que si se desarrolla una aplicación Web, con una arquitectura adecuada, que automatice la gestión de información en el Departamento Docente Central de Ingeniería y Gestión de Software de la UCI, entonces se facilitará la planeación y toma de decisiones en el proceso docente metodológico del departamento.

El objetivo principal de este trabajo es definir una arquitectura para una aplicación Web, segura y de interfaz amigable, que facilite el control y la gestión de los procesos fundamentales del Departamento Docente Central de Ingeniería y Gestión de Software de la UCI con el desarrollo de su prototipo funcional.

Objetivos específicos:

- Investigar y describir sobre la organización del proceso docente metodológico del departamento.
- Analizar Sistemas de Gestión Docente existentes y similares en el país y en el extranjero, y su posible aplicación según las condiciones actuales del departamento.
- Analizar el Diseño Arquitectónico a utilizar.
- Desarrollo del sistema hasta una primera iteración de la fase de elaboración haciendo uso de RUP.

Para cumplir estos objetivos se han trazado las siguientes tareas:

- Realizar estudio del entorno de trabajo.
- Identificar las necesidades del cliente y objeto de automatización.
- Seleccionar la metodología de desarrollo de software, que facilite la creación y garantice la calidad de la aplicación.
- Selección y justificación de las herramientas para llevar a cabo el desarrollo de la aplicación.
- Realizar el análisis y el diseño de la aplicación.
- Implementación de la arquitectura
- Realizar pruebas.

El presente trabajo ha sido organizado de la siguiente manera:

Capítulo 1: Aborda los conceptos principales relacionados con los procesos que tienen lugar en el Departamento Docente Central de Ingeniería y Gestión de Software, se detallan los procesos que se desean automatizar, fundamentándose los objetivos generales y específicos a cumplimentar. Además se realiza un análisis de las tecnologías a utilizar en el desarrollo del sistema, realizando la selección de las más adecuadas para el desarrollo del trabajo.

Capítulo 2: Se hace una descripción del negocio a través de un Modelo de Negocio. Para esto se identifican los actores y trabajadores del negocio, y se realizan diagramas de casos de uso del negocio, actividad y modelo de objetos.

Capítulo 3: En este capítulo se definen las principales funcionalidades del software, describiéndolas detalladamente mediante herramientas de modelación. Además se definen los actores y casos de uso del sistema, expandiéndose cada uno de ellos.

Capítulo 4: Aborda aspectos relacionados con la construcción de la solución propuesta, se modelan los diagramas de clases del diseño, se plantea el modelo de datos, y se especifican los principios para el diseño gráfico y la implementación.

Capítulo 5: Se hace un estudio de factibilidad de la construcción del sistema, enumerándose los beneficios tangibles e intangibles.



1 FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se realiza una descripción general de los procesos actuales del Departamento Docente Central de Ingeniería y Gestión de Software y se detallan los principales problemas que motivan esta investigación. Se analiza además las tecnologías actuales que serán utilizadas para el desarrollo del trabajo, fundamentándose finalmente los objetivos propuestos.

1.2 Objeto de estudio

La Universidad de las Ciencias Informáticas cuenta con una Vicerrectoría de Formación conformada por la vicerrectora y diferentes direcciones como: la Dirección de Planificación y Control, Dirección de Teleformación, Dirección de Formación del Posgraduado, y Dirección de Formación del Profesional correspondientes a las 11 disciplinas de la carrera, la cual dirige los Departamentos Docentes Centrales, donde se encuentra el Departamento Docente Central de Ingeniería y Gestión de Software. Teniendo en cuenta las características de formación del profesional que distinguen a la UCI y al ser una universidad sui generis por su condición de estudiarse una sola profesión, en la que existe un plan de estudios troncal que debe garantizar la homogeneidad en la calidad del egresado, es necesario contar con Departamentos Docentes Centrales que garanticen la organización y dirección de los procesos en el trabajo docente metodológico.

1.2.1 Objetivos estratégicos de la organización

El Departamento Docente Central de Ingeniería y Gestión de Software de la UCI, esta orientado a brindar formación académica y profesional a estudiantes y profesores, con calidad, que respondan a los intereses de nuestra sociedad, así como lograr una gestión educativa productiva y eficiente en la rama de la Ingeniería del Software, Calidad y Bases de Dato. Su dirección esta enfocada en los siguientes objetivos:

- Formar en los egresados los hábitos de organización personal y responsabilidad que se requiere en la proyección de sistemas automatizados.
- Contribuir a consolidar un estilo de trabajo independiente colectivo y creativo en la solución de los problemas con un alto nivel de profesionalidad, así como la eficiencia y el ahorro de recursos en las soluciones adoptadas.
- Contribuir a crear en el egresado un enfoque independiente y creativo en la proyección de sistemas.
- Desarrollar en los estudiantes la necesidad de búsqueda de información adicional acerca de los avances en las técnicas e instrumentos de desarrollo de software, contribuyendo de esta forma en la creación de un fuerte espíritu de autopreparación.
- Profundizar en el desarrollo de las formas de pensamiento lógico y la capacidad de abstracción y de razonamiento mediante el análisis de los problemas y la modelación de los fenómenos de la realidad objetiva.
- Analizar un sistema en una organización de base productiva o de servicio o en cualquier otro medio para determinar la factibilidad económica, las tareas a automatizar, los medios técnicos necesarios y la planificación del proyecto.
- Contribuir a la aplicación de un enfoque de procesos a los problemas de gestión en el desarrollo de aplicaciones.
- Aplicar técnicas para asegurar la calidad de los productos informáticos en la empresa.
- Realizar análisis de factibilidad económica, tareas a automatizar, medios técnicos necesarios y planificación del proyecto.
- Planificar estratégicamente la informática en una entidad.
- Aplicar los procedimientos para gestionar configuraciones de software.

- Desarrollar los procesos de gestión de software, asegurar la calidad de los productos de software.
- Preparar y entrenar al personal necesario en la puesta en marcha de los sistemas informáticos.
- Aplicar y seleccionar las diferentes técnicas de obtención de información durante el estudio preliminar y el análisis de los sistemas informáticos.
- Analizar el sistema organizativo e informativo de una entidad productiva; de servicio o técnica a cualquier nivel aplicando una metodología moderna de análisis de sistemas.
- Diseñar la arquitectura apropiada de sistemas informáticos según su naturaleza y complejidad.
- Utilizar adecuadamente patrones de diseño y de arquitectura para facilitar el desarrollo y asegurar la calidad del producto de software.
- Desarrollar técnicas modernas de trabajo en grupo. Realizar proyectos en equipo jugando los diferentes roles presentes en éstos.
- Preparar y entrenar personal para el uso de instrumentos CASE que le permitan el desarrollo de sus propias aplicaciones.
- Programar y poner a punto algoritmos complejos para el procesamiento de datos almacenados en una base de datos a través de un lenguaje de gestión de bases de datos y documentar adecuadamente los programas.
- Representar un fenómeno de la realidad objetiva a través del modelo lógico global de los datos.
- Desarrollar algoritmos complejos para el procesamiento de datos almacenados en una base de datos.
- Confeccionar la documentación técnica del sistema informático y la orientada a los usuarios finales.

1.2.2 Flujo actual de los procesos

La organización del Departamento Docente Central de Ingeniería de Software, Base de datos y Gestión de Software, esta formada por: un Jefe de Departamento, un asesor por asignatura, una secretaria, y un colectivo por facultad con su correspondiente claustro de profesores. Este departamento es el encargado de:

- Brindar asesoramiento a los colectivos en la planificación de las asignaturas de la disciplina.
- Es responsable de los proyectos de preparación, desarrollo y perfeccionamiento de la disciplina.
- Coordinar y fiscalizar el trabajo metodológico y actividades académicas a nivel central, así como otras actividades de preparación y superación de estudiantes y profesores.
- Dirigir el grupo de asesores encargados de la dirección metodológica de las asignaturas de la disciplina.
- Controlar el plan de trabajo de los asesores, jefe de colectivo y profesores; de estos dos últimos en coordinación con las facultades, en los aspectos docentes y metodológicos.

En la actualidad los procesos se llevan a cabo de forma manual, comenzando por conformar la plantilla del departamento, donde se seleccionan los profesores que impartirán cada asignatura en las facultades. A partir de los problemas de semestres anteriores y de las características del claustro se elabora el Plan de Trabajo Metodológico. Además, se confeccionan y se concilian los planes individuales de cada asesor teniendo en cuenta los objetivos del departamento. Para garantizar la obtención de recursos y medios, en el curso, se hacen solicitudes de materiales a la Vicerrectoría de Formación (VRF). Al final de cada mes se confecciona la pre Nómina de pago y aseo.

Con el desarrollo del semestre se comienzan a realizar los controles a clases, aspecto importante como medidor de la calidad del proceso docente. Comienzan las evaluaciones parciales y se hace necesario al culminar cada evaluación el análisis de los resultados. Cada cierto tiempo se controla cómo ha sido el cumplimiento del Plan de Trabajo Metodológico y el Plan de Trabajo Individual de los asesores. Al final de cada semestre se elabora el informe semestral donde se plasman los resultados alcanzados, así como los problemas detectados para su corrección en etapas posteriores.

1.2.3 Análisis crítico de la ejecución de los procesos

Comienza el curso y el departamento docente central desconoce los profesores que van a impartir las asignaturas de su disciplina debido a demoras en obtener la información. Esto provoca que el Plan de Trabajo Metodológico se realice sobre un

diagnóstico del claustro que no refleja en todas las ocasiones, la situación real, provocando que disminuya la calidad del proceso docente. Al no llevarse un control eficiente de lo que se ha solicitado al almacén, el departamento se sorprende con la ausencia de algún recurso que es importante para un buen desarrollo del proceso. Llega el final del mes y por la vorágine de trabajo no se realiza la pre Nómina y aseo, provocando un descontento en el personal por retraso en el pago. Al no planificarse los controles a clases y no llevarse un seguimiento eficiente de los mismos, tomando en cuenta los aspectos positivos y negativos detectados, no se puede trabajar sobre las mayores deficiencias que tiene el claustro. Todo esto provoca demoras en la toma de decisiones así como gasto innecesario de recursos en el proceso docente metodológico.

1.3 Procesos objeto de automatización

Los procesos objeto de automatización son:

1. **Confeccionar la plantilla del departamento:** Para una buena organización del claustro se registra la plantilla de profesores de cada colectivo y trabajadores del departamento en sí, con sus respectivos datos. Si en el transcurso del curso surgen reestructuraciones o modificaciones del claustro, el departamento debe tener control de los cambios, lo cual es a través de la plantilla del departamento o disciplina.
2. **Elaboración y control del Plan de Trabajo Metodológico:** Las actividades que se desarrollan en el semestre son el resultado del análisis de etapas anteriores y nuevas proyecciones para mejorar el proceso docente metodológico. Una buena planificación, ejecución y control garantizan el éxito del resultado docente metodológico. Esta planificación se efectúa en el Plan de Trabajo Metodológico.
3. **Elaboración y control del Plan de Trabajo Individual:** Relacionado al Plan de Trabajo Metodológico está el Plan de Trabajo Individual de los asesores, donde éstos se trazan objetivos, y actividades a cumplir. Es de suma importancia el papel de los asesores, pues ellos complementan los objetivos del departamento. Periódicamente el Jefe de Departamento evalúa el cumplimiento de los objetivos por parte de los asesores.
4. **Elaboración y control de los controles a clases:** Los controles a clases se planifican seleccionando responsables, facultades y profesores a controlar.

Analizan las deficiencias detectadas y permite introducir actividades en el Plan de Trabajo Metodológico en favor de erradicarlas, registrando los resultados de los controles. Esto da un indicador del nivel de preparación de los profesores y dónde están los principales problemas.

5. **Confección y archivo de solicitud de materiales:** Para la obtención de materiales docentes, el departamento realiza una solicitud a la Vicerrectoría de Formación. Es de suma importancia un control eficaz de estas solicitudes, con el fin de garantizar peticiones acorde a las solicitudes anteriores.
6. **Confección de la nómina de pago y aseo:** Para la obtención del pago y aseo que reciben los trabajadores del departamento cada mes, es responsabilidad del Jefe de Departamento confeccionar la nómina y enviarla a la dirección de recursos humanos.

1.4 Sistemas automatizados existentes vinculados al campo de acción

Actualmente no se tiene conocimiento de la existencia de sistemas que se ajusten a las condiciones y estructura del Departamento Docente Central de Ingeniería y Gestión de Software.

1.4.1 Justificación

Teniendo en cuenta las características de los procesos que se desean automatizar, enmarcados en las actividades que desarrolla el Departamento Docente Central de Ingeniería y Gestión de Software de la UCI, existe una especificidad que delimita la semejanza con otros centros y organizaciones del país y el extranjero, lo que conlleva de igual forma a la existencia de sistemas que gestionen información acorde a los procesos en cuestión.

Después de una búsqueda en otros centros e Internet de sistemas que se ajusten a las características del departamento, no se encontraron resultados satisfactorios. En centros de educación superior del país y en la UCI, los sistemas existentes están más enfocados al proceso docente en sí y no a las actividades metodológicas.

1.5 Fundamentación de los objetivos

Las actividades metodológicas de controles a clases, preparación, superación, trabajo científico, teleformación, entre otras, son claves en el proceso docente metodológico que representa el departamento. Muchas veces por falta de información del claustro de profesores que conforman la disciplina, ocurren desajustes e incumplimiento de las actividades. Además, no existen mecanismos que le permita al Jefe de Departamento y asesores conocer y evaluar el estado de cumplimiento de las actividades en determinado momento. Para darle solución a lo anterior expuesto, se propone este trabajo, trazándose un conjunto de objetivos.

1.5.1 Objetivo general

El objetivo de este trabajo es elaborar una aplicación Web para automatizar la gestión de información del proceso docente metodológico en el Departamento Docente Central de Ingeniería y Gestión de Software, permitiendo un rápido y fácil acceso desde cualquier lugar de la UCI.

1.5.2 Objetivos específicos

Para cumplir el objetivo principal de este trabajo se trazaron los siguientes objetivos específicos:

- Investigar y describir sobre la organización del proceso docente metodológico del departamento.
- Analizar Sistemas de Gestión Docente existentes y similares en el país y en el extranjero, y su posible aplicación según las condiciones actuales del departamento.
- Analizar la arquitectura a utilizar para el desarrollo del sistema.
- Desarrollar el ciclo de vida del sistema haciendo uso de una metodología de desarrollo de Software.

1.6 Tendencias y tecnologías actuales

1.6.1 Política de migración a software libre

Actualmente existe una tendencia en la Universidad de las Ciencias Informáticas, y en el país en general, hacia la utilización en grado creciente de software libre. Cada vez

se promociona más la migración desde los sistemas con licencia comercial que están en poder de unos pocos monopolios de la rama de la informática, hacia aquellos que se denominan "libres", es decir, aquellos cuyo uso por parte de cualquiera que esté interesado en hacerlo está exento de pago.

Como institución de avanzada en el campo de la informática, la UCI está prácticamente obligada a llevar a cabo, y cuánto antes mejor, esta migración. El presente trabajo parte de esa premisa y se propone la construcción de un sistema que satisfaga las necesidades que lo originaron, haciendo uso de herramientas y tecnologías libres.

1.6.2 PHP

"PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje "Open Source" interpretado y de alto nivel, especialmente pensado para desarrollos Web. La mayoría de su sintaxis es similar a C, Java y Perl y es extremadamente fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas Web, páginas dinámicas de una manera rápida y fácil. "[1]

El código escrito en PHP es independiente de la plataforma, por lo puede ser ejecutado en cualquier sistema operativo, gracias a que existen interpretes de PHP para los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluido HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS, entre otros. PHP es soportado por la mayoría de servidores Web de hoy día, incluyendo Apache, Microsoft Internet Information Services, Personal Web Server, Netscape e iPlanet, Oreilly Website Pro Server, Caudium, Xitami y OmniHTTPd. Este lenguaje es muy rápido, con una gran librería de funciones y mucha documentación, además es interpretado y embebido en el HTML

Este lenguaje resulta muy atractivo y está respaldado por una gran comunidad de desarrolladores, debido a la perfecta combinación que proporcionan su simpleza para los programadores inexpertos, y a la vez su poder para los programadores profesionales. [1]

Con PHP4 se han logrado aumentos de entre 5 y 10 veces en la velocidad de ejecución de páginas PHP. Estos resultados le ha puesto por delante de ASP, la

tecnología de Microsoft, permitiéndole competir directamente con ASP en la plataforma de Microsoft. La plataforma de desarrollo Apache+PHP en entornos Microsoft ha logrado rendimientos superiores a IIS+ASP.

Actualmente se encuentra en su versión 5. Es evidente que se ha convertido en la gran tendencia en el mundo de Internet. Las estadísticas arrojan que cada mes su uso crece en un 15% y cuenta con una de las comunidades más grandes de Internet, lo cual facilita encontrar ayuda, documentación, y otros recursos relacionados.

1.6.3 Plataforma de desarrollo Web

En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, librerías y un lenguaje de scripting entre otros softwares para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Un framework representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.[2]

Entre los principales tenemos los siguientes: Plone que utiliza el lenguaje de programación Pitón; Alfresco con plataforma Java y soporta los gestores de Base de Datos como MySQL, Oracle, SQLServer, PostgreSQL; Atlantic WebFitters con plataforma Visual Basic.Net y gestor de Base de Datos SQLServer; Caravel CMS con plataforma PHP y gestores de Base de Datos OpenLDAP y PostgreSQL; CMScout con plataforma PHP y gestor de Base de Datos MySQL.

1.6.4 Drupal

Drupal es una plataforma dinámica para la construcción de sitios Web que permite a individuos o comunidad de usuarios publicar, manejar y organizar una variedad de contenido, Drupal integra muchas características populares de los Sistemas de Gestión de Contenido, weblogs, herramientas de colaboración y comunidad de discusión, todo en un solo paquete fácil de utilizar. Como software de código abierto desarrollado y mantenido por una comunidad Drupal es libre para descargarlo de Internet y usarlo. [3]

Drupal posee un conjunto de características que lo hacen muy adaptable como son:

Independencia de la base de datos: Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal incorpora una 'capa de abstracción de base de datos' que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.

Multiplataforma: Drupal ha sido diseñado desde el principio para ser multiplataforma. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X. Por otro lado, al estar implementado en PHP, es totalmente portable.

Múltiples idiomas y Localización: Drupal está pensado para una audiencia internacional y proporciona opciones para crear aplicaciones Web multilingüe. Todo el texto puede ser fácilmente traducido utilizando una interfaz web, importando traducciones existentes o integrando otras herramientas de traducción como *GNU Gettext*.

Requisitos básicos de instalación y configuración inicial: El núcleo de la plataforma Drupal, módulos adicionales, y muchas plantillas de diseño están libremente disponibles para la ser descargadas bajo una licencia GPL. [4]

En la arquitectura de Drupal se observan patrones de diseño como son Singleton, Observer, Decorator, Bridge, Command, además se encuentran principios de diferentes paradigmas de la programación (Programación Orientada a Objetos, Programación Orientada a Componentes, Programación Dirigida por Eventos, Programación Orientada a Aspectos).

No obstante el patrón que define la arquitectura de Drupal y marca la diferencia con otros sistemas de su categoría es el "Reflection Pattern". Gracias a este patrón el sistema se convierte en una aplicación muy adaptable a diferentes entornos puesto que permite prácticamente la modificación de todo su comportamiento a través de módulos instalables, sin la necesidad de modificar del núcleo.

1.6.5 Zend Studio

Zend Studio es uno de los ambientes de desarrollo integrado o *Integrated Development Environment* (IDE) disponible para desarrolladores profesionales, agrupa todos los componentes necesarios para ciclo de desarrollo de aplicaciones PHP. A través de un comprensivo conjunto de herramientas de edición, depurado, análisis, optimización y bases de datos, Zend Studio acelera los ciclos de desarrollo y simplifica los proyectos complejos.

1.6.6 SQL

Debido a la diversidad de lenguajes y de bases de datos existentes, la manera de comunicar entre unos y otras sería realmente complicada a gestionar de no ser por la existencia de estándares que nos permiten el realizar las operaciones básicas de una forma universal.

El Structured Query Language (SQL) es una herramienta para organizar, gestionar y recuperar datos almacenados en una base de datos informática. Como su propio nombre indica, SQL es un lenguaje informático que se puede utilizar para interactuar con una base de datos y más concretamente con un tipo específico llamado *base de datos relacional*. [5]

SQL es a la vez un lenguaje fácil de aprender y una herramienta completa para gestionar datos. Las peticiones sobre los datos se expresan mediante sentencias, que deben escribirse de acuerdo con unas reglas sintácticas y semánticas de este lenguaje. Su aprendizaje sirve para todas las aplicaciones gestoras de bases de datos existentes en el mercado que soporten este lenguaje (que son muchas) ya que es un lenguaje estándar por haberse visto consolidado por el Instituto Americano de Normas (ANSI) y por la Organización de Estándares Internacional (ISO). [5]

1.6.7 Sistema gestor de bases de datos MySQL

Desde sus inicios el MySQL se ha convertido en el Gestor de Bases de datos de código abierto más popular de Internet, principalmente por su simpleza y robustez. Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas Web con contenido dinámico, justamente por su

simplicidad; aquellos elementos faltantes fueron llenados por la vía de las aplicaciones que la utilizan.

Poco a poco los elementos faltantes en MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre.

Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL.
- Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Procedimientos almacenados.

1.6.8 El Proceso Unificado de Modelado (RUP)

El software es un componente esencial de toda actividad basado en el uso de la informática, es por ello que la calidad en su desarrollo y mantenimiento resulta hoy en día uno de los principales objetivos estratégicos de las organizaciones. En los últimos años se han publicado diversos estudios en los que se presentan los principios que se deben seguir para mejorar los procesos de software, y de esta forma evitar las grandes catástrofes que conllevan al fracaso de un gran número de proyectos.

RUP es una de las metodologías más generales y más usadas de las que existen en la actualidad, pues está pensada para adaptarse a cualquier proyecto. Constituye además, una propuesta de proceso para el desarrollo de software orientado a objeto, utilizando UML (del inglés Unified Model Language), para describir todo el proceso, basándose en componentes. Este lenguaje es estándar, con él se puede visualizar, especificar, construir y documentar los artefactos de un sistema.

Las principales características de esta metodología son:

Centrado en los Modelos: Los diagramas son un vehículo de comunicación más expresivo que las descripciones en lenguaje natural. Se trata de minimizar el uso de descripciones y especificaciones textuales del sistema.

Guiado por los casos de uso: Los casos de uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.

Iterativo e incremental: Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo.

RUP incluye cuatro etapas importantes que son: Inicio, Elaboración, Construcción y Transición, cada una de ellas compuesta de una o varias iteraciones. Estas etapas revelan que para producir una versión del producto en desarrollo se emplean todas las actividades de ingeniería pero con diferente énfasis; en las primeras versiones se hace más énfasis en el modelado del negocio, requisitos, análisis y diseño; mientras en las posteriores el énfasis recae sobre las actividades de implementación, pruebas y despliegue. Además contempla flujos de trabajo de soporte que involucran actividades de planificación de recursos humanos tecnológicos y financieros.

1.6.9 Diseño Arquitectónico

El diseño es una actividad en la que se toman decisiones importantes, frecuentemente de naturaleza estructural. Comparte con la programación un interés por la abstracción de la representación de la información y de las secuencias de procesamiento, pero el nivel de detalle es muy diferente en ambos casos. El diseño constituye representaciones coherentes y bien planificadas de los programas, concentrándose en las interrelaciones de los componentes de mayor nivel y en las operaciones lógicas implicadas en los niveles inferiores. [02]

La arquitectura de software de un sistema es la estructura general la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente, y las relaciones entre ellos.

Existen tres razones por las que la arquitectura de software es importante:

- Las representaciones de la arquitectura de software facilitan la comunicación entre todas las partes interesadas en el desarrollo de un sistema basado en computadoras.
- La arquitectura destaca decisiones tempranas de diseño que tendrán un profundo impacto en todo el trabajo de ingeniería del software que sigue, y es tan importante en el resultado final del sistema como una entidad operacional.
- Constituye un modelo relativamente pequeño e intelectualmente comprensible de cómo está estructurado el sistema y de cómo trabajan juntos sus componentes.[8]

1.6.9.1 Patrón arquitectónico Modelo-Vista-Controlador (MVC)

El Modelo-Vista-Controlador, fue introducido inicialmente por la comunidad de desarrolladores de Smalltalk, separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes.

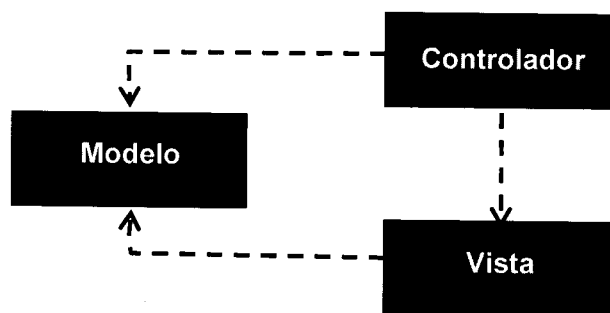


Figura 2: Modelo-Vista-Controlador.

El **Modelo** es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

La **Vista** es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

El **Controlador** es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

Entre las ventajas señaladas están las siguientes:

- **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 de 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 o PDAs (del inglés, **Personal Digital Assistant**). Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo. Este patrón sentó las bases para especializaciones ulteriores, tales como Page Controller y Front Controller.

Entre las desventajas, se han señalado:

- **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. En rigor, la configuración basada en eventos de dicha interfaz corresponde a un estilo particular (arquitectura basada en eventos) que aquí se examina por separado.

- **Costo de actualizaciones frecuentes.** Desacoplar el modelo de la vista no significa que los desarrolladores del modelo puedan ignorar la naturaleza de las vistas.

1.6.10 Selección de las tecnologías a utilizar en la propuesta de solución

Después de haber realizado un estudio de las principales características de las tecnologías más usadas en la actualidad, se decidió utilizar PHP como lenguaje del lado del servidor, por todas las ventajas que presenta por su propia esencia. Es multiplataforma, con una sintaxis familiar a los programadores que participaran en la construcción del trabajo propuesto, y cuenta con gran disponibilidad de recursos en Internet. Como sistema gestor de base de datos se seleccionó MySQL por ser capaz de implementar funcionalidades Web que permiten un acceso a los datos, seguro y fácil, desde Internet. Además es uno de los SGBD más populares, desarrollado bajo la filosofía de código abierto. Es escogido Drupal como plataforma de desarrollo Web por su adaptabilidad, flexibilidad y por ser además de código abierto cumpliendo con los términos de GNU/ GPL (General Public License). Como ambiente de desarrollo se escogió Zend Studio, y como metodología de desarrollo RUP por todas las ventajas de organización que brinda y por venir acompañada de una potente herramienta que soporta todos los procesos básicos de RUP: Suite del Rational.

1.7 Conclusiones

En este capítulo se ha descrito el objeto de estudio de este trabajo, determinándose los problemas que presenta. Se ha demostrado la necesidad de cambio y se ha planteado una propuesta de solución, junto a los objetivos generales y específicos de este trabajo. Además se realizó un análisis de las tecnologías actuales, describiéndose las características fundamentales de las seleccionadas para el desarrollo del sistema.



2 MODELO DEL NEGOCIO

2.1 Introducción

En el presente capítulo se describen los procesos del negocio asociados al objeto de estudio. Es importante conocer cómo funcionan estos procesos, ya que ayudan a tener una concepción más clara del funcionamiento del Departamento Docente Central de Ingeniería y Gestión de Software, para elaborar una aplicación que se ajuste a las reglas del mismo.

2.2 Modelo del negocio actual

El Departamento Docente Central de Ingeniería y Gestión de Software realiza las siguientes actividades:

1. Gestión de los colectivos de profesores por asignatura.
 - Los datos de cada profesor (nombre, apellidos, número de solapín, responsabilidades académicas, productivas, categoría docente, científica, dirección particular, y teléfonos tanto de internos como externos).
 - Las habilidades que ha desarrollado cada profesor en los diferentes temas de la asignatura.
 - Los resultados de los controles a clases de los profesores.
 - Conserva los datos anteriores de los profesores en los diferentes cursos para tomar decisiones.
 - Los horarios docentes de cada año y de cada facultad.
2. Se elabora y controla el Plan de Trabajo Metodológico (PTM).

- Planificación de las diferentes actividades del PTM, según tipo de actividad (de superación, trabajo científico metodológico, control a clases, tele formación, preparación de las asignaturas, y del trabajo metodológico como tal).
 - Control de cumplimiento del plan de trabajo metodológico. Estado de las actividades (cumplida, parcialmente cumplida, e incumplida).
 - El control de las actas de las reuniones metodológicas.
 - Resumen semestral de del cumplimiento del PTM y otras.
 - Acta del departamento conocido como Informe semestral.
3. Cuando se necesita materiales de oficina y de cómputo se hace la solicitud a las diferentes Vicerreorías.
- Se confecciona el modelo de solicitud.
 - Se justifica la necesidad de cada medio pedido.
 - A partir de la aprobación de la solicitud, se controla que materiales abasteció el almacén.
4. Cada mes se elabora la pre Nómina y la solicitud de aseo.
- Confeccionar pre Nómina de pago y aseo para trabajadores que constituyen plantillas del departamento.
5. La recogida de pruebas parciales, cortes de proyectos y pruebas finales por asignatura y por grupos de las facultades.
6. Elaboración del plan de trabajo de los asesores.

Actualmente se confecciona el modelo de pre Nómina de pago y aseo correspondiente a cada mes, se envía a la Dirección de Capital Humano para que este proceda con el pago y la asignación de aseo a los trabajadores del departamento, si el trabajador a laborado por lo menos un día. En la solicitud de materiales se confecciona el modelo del pedido según tipos de materiales, redactando una justificación por cada uno, y se envía a la Dirección de Formación para su revisión, y a la Vicerreoría Docente donde es evaluado y aprobado.

El PTM elaborado en el departamento está compuesto por actividades. Para cada actividad se conoce la categoría a la que pertenece, fecha de cumplimiento, responsables, y participantes en las mismas. En algunos tipos de actividades es necesario conservar que la explica. En cuanto al Plan de Trabajo Individual de los

asesores, estos lo elaboran y es aprobado por el Jefe de Departamento. En el se plasman los objetivo y las actividades fundamentales, No solo se planifican actividades si no que se chequea su cumplimiento.

2.3 Reglas del negocio a considerar

- La solicitud de materiales a la Vicerrectoría de Formación se hace por separado para los que son clasificados de oficina y para los materiales técnicos.
- El aseo le corresponde a los trabajadores que tengan al menos un día trabajado en el mes.
- En las actividades correspondientes a las reuniones metodológicas se analiza un tema metodológico y constan de la elaboración de un acta.
- El Plan de Trabajo Metodológico se debe terminar en las tres primeras semanas del curso.
- Cuando se concilia el Plan de Trabajo Metodológico no se puede modificar.
- Después de conciliado el Plan de Trabajo Individual, debe actualizarse el cumplimiento periódicamente.
- La recogida de información de los resultados de las evaluaciones se efectúa tres días después de aplicado.

2.4 Actores del negocio

Tabla 1. Descripción de los actores del negocio

Nombre del actor	Descripción
Director de Formación	Es aquel que se beneficia con el sistema al solicitar al Jefe de Departamento la composición de la plantilla, planes de trabajos, controles a clases. Así como los resultados generales en el desempeño docente metodológico.
Director de Capital Humano	Es el que recibe el modelo de nómina y aseo para efectuar el pago a los trabajadores.

2.5 Diagrama de casos de uso del negocio

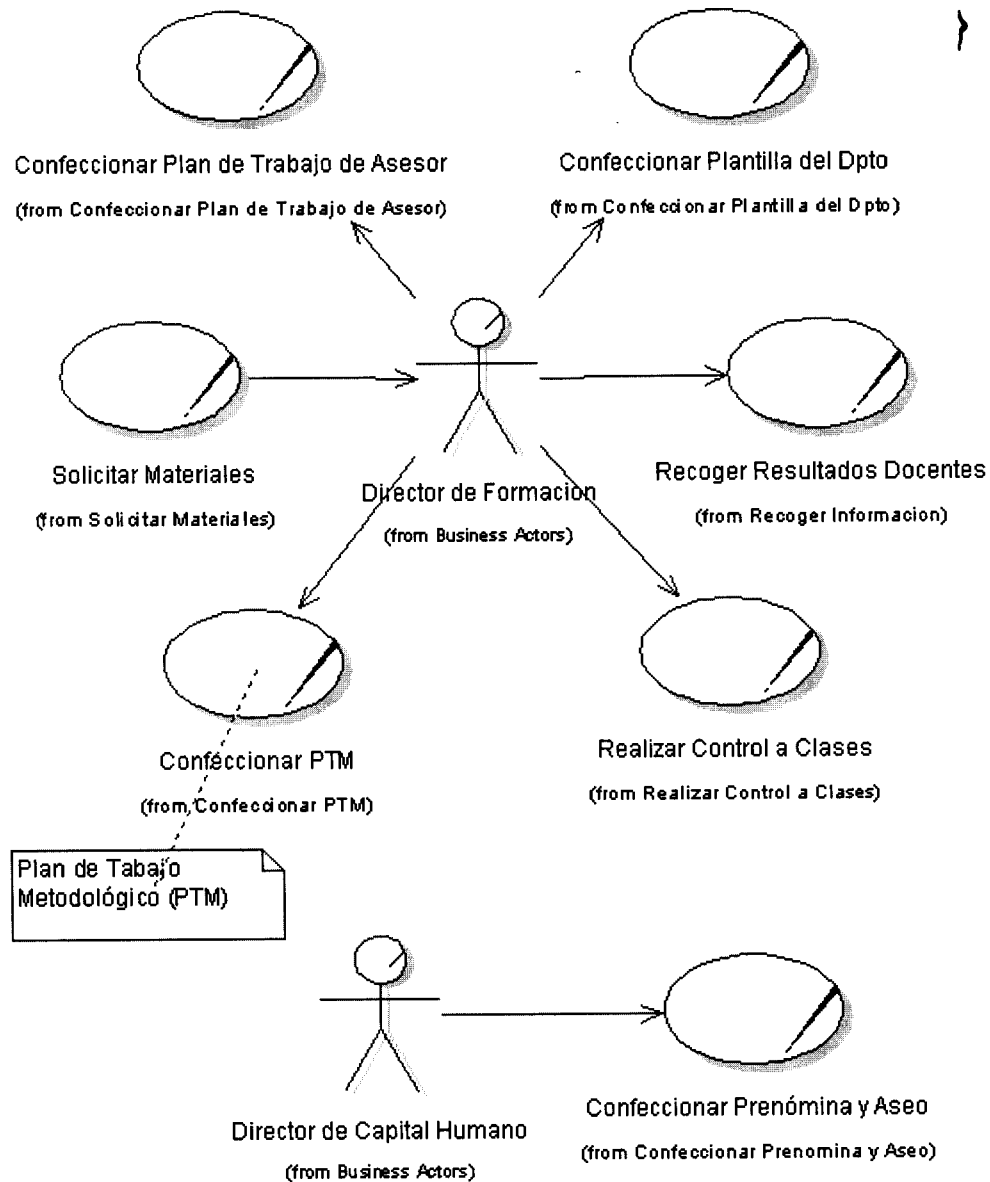


Figura 1. Diagrama de casos de uso del negocio.

2.6 Trabajadores del negocio

Tabla 2. Descripción de los trabajadores del negocio

Nombre del trabajador	Descripción
Jefe de Departamento	Es el encargado de completar la Plantilla del Departamento y Planificar el Control a Clases. Confeccionar informe semestral del departamento y Plan de Trabajo Metodológico. Además solicita, completa, y aprueba el Plan de Trabajo de Asesores.
Asesor	Es el encargado de confeccionar la Plantilla por asignatura, elaborar propuesta del Plan de Trabajo seguir, confeccionar informe semestral de la asignatura; realiza junto al Jefe de Departamento el Control a Clases y coopera en la confección del Plan de Trabajo Metodológico.
Jefe de Colectivo	Es el encargado de confeccionar la Plantilla del colectivo para confeccionar la Plantilla por asignatura, es el responsable de introducir los resultados docentes de diferentes evaluaciones.
Revisor	Es el encargado de realizar y chequear el control a clases.

2.7 Casos de uso del negocio

Tabla 3. Caso de uso: Confeccionar Plan de Trabajo Metodológico (PTM).

Caso de Uso:	Confeccionar PTM
Actores:	Director de Formación
Trabajadores:	Jefe de Departamento, Asesores
Resumen:	El caso de uso se inicia cuando el Director de Formación solicita el Plan de Trabajo Metodológico al Jefe de Departamento. Este solicita un diagnóstico por asignaturas a los asesores con el cual conjuntamente con el informe semestral define los problemas, los objetivos y las acciones, Todo esto es revisado por los asesores

	según se van definiendo y al enviársela nuevamente al jefe del departamento, este va confeccionando poco a poco el Plan de Trabajo Metodológico.
Precondiciones:	La Plantilla del Departamento debe estar terminada.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1-El Director de Formación hace una solicitud al jefe del departamento del plan metodológico.	<p>1.1-El Jefe de Departamento solicita un diagnóstico en cuanto al claustro por asignatura a los asesores.</p> <p>1.2-Los asesores elaboran un diagnóstico y se lo entregan al Jefe de Dpto.</p> <p>1.3-El Jefe de Departamento utilizando este diagnóstico y el informe semestral realiza un diagnóstico general, definiendo los problemas a solucionar.</p> <p>1.4-Define los objetivos y la estrategia</p> <p>1.5-Los asesores con el diagnostico y los objetivos definen acciones a desarrollar.</p> <p>1.6-El Jefe de Departamento aprueba y añade las acciones.</p> <p>1.7-Queda concluido el Plan de Trabajo Metodológico.</p>
2.-Recibe Plan de Trabajo Metodológico.	1.8-Se le entrega al director de formación.
Poscondiciones	Queda elaborado el Plan de Trabajo, se hace entrega del mismo.

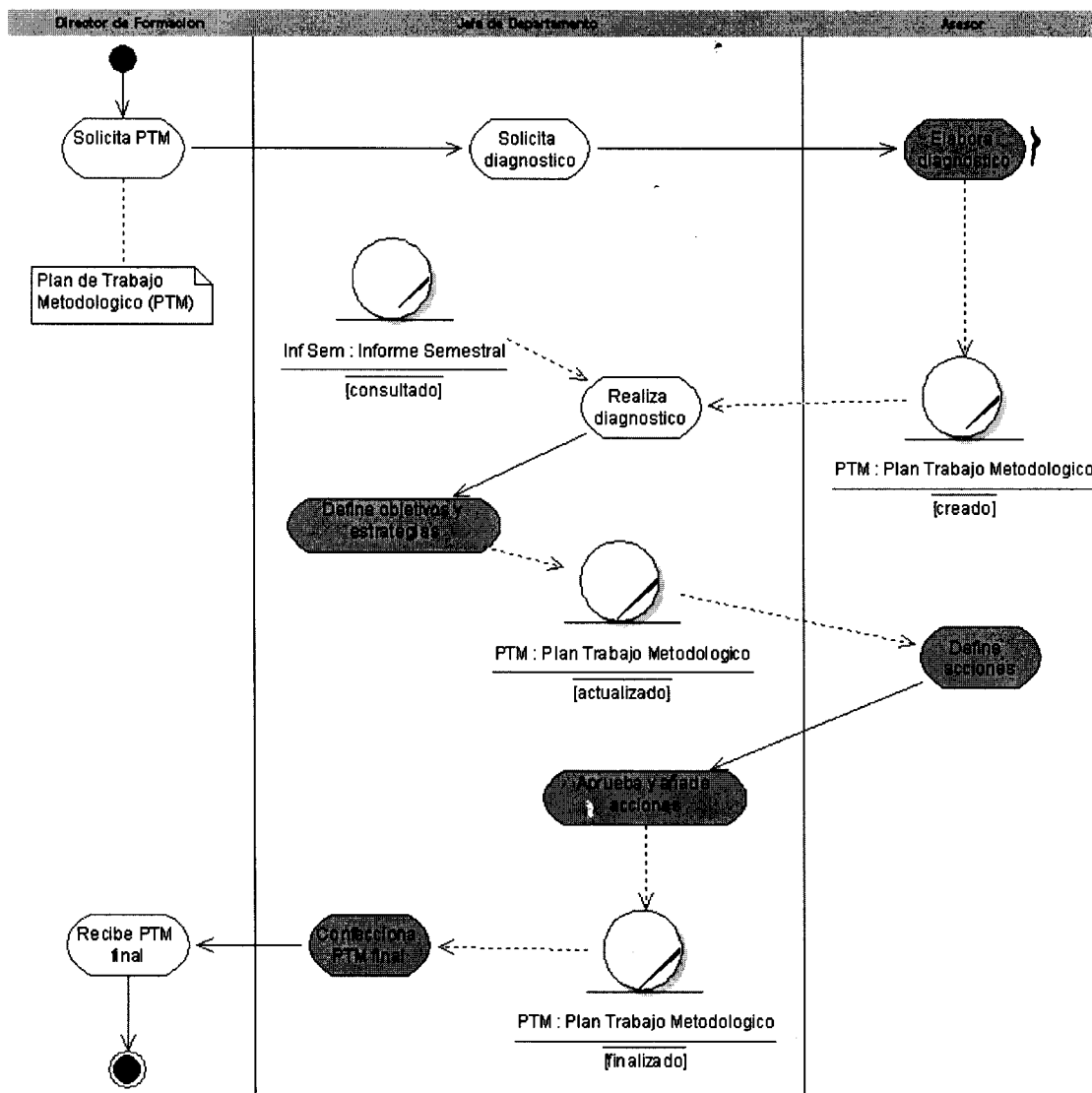


Figura 2. Diagrama de actividad del caso de uso: Confeccionar Plan de Trabajo Metodológico.

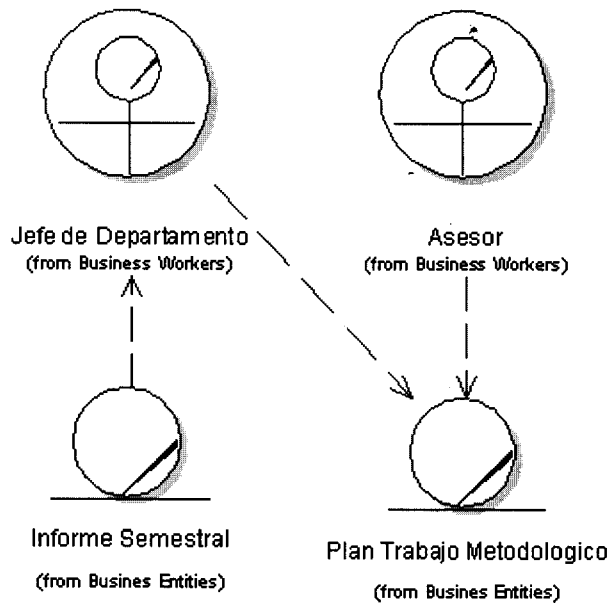


Figura 3. Modelo de objeto del caso de uso: Confeccionar Plan de Trabajo Metodológico.

Tabla 4 Caso de uso: Confeccionar Plan de Trabajo de Asesor.

Caso de Uso:	Confeccionar Plan de Trabajo de Asesor
Actores:	Director de Formación.
Trabajadores:	Jefe de Departamento, Asesores.
Resumen:	El caso de uso se inicia cuando el Director de Formación solicita el Plan de Trabajo de los Asesores al Jefe de Departamento. Este solicita un Plan de Trabajo a los Asesores, los cuales elaboran un plan de Trabajo a seguir en el período y se lo envían al Jefe de Departamento, él mismo lo revisa modifica, adiciona, y/o elimina objetivos a cumplir en el. Después de quedar conciliado queda concluido y es enviado al VRF.
Precondiciones:	
Flujo Normal de Eventos	

CAPÍTULO 2 MODELO DEL NEGOCIO

Acción del Actor	Respuesta del Negocio
<p>1- El Director de Formación hace una solicitud al Jefe de Departamento del plan de trabajo de los asesores.</p> <p>2-El Director de Formación recibe el plan de trabajo de asesor.</p>	<p>1.1-El Jefe de Departamento solicita una propuesta de Plan de Trabajo a los Asesores.</p> <p>1.2-Los asesores elaboran una propuesta y se la entregan al Jefe de Dpto.</p> <p>1.3-El Jefe de Departamento revisa la propuesta y utilizando la misma modifica el Plan de Trabajo, incluyendo o no otras actividades.</p> <p>1.5-Los asesores junto al Jefe de Departamento llegan a un concilio acerca de la propuesta de Plan de Trabajo presentado.</p> <p>1.6-El Jefe de departamento aprueba finalmente el Plan de Trabajo.</p> <p>1.7-Queda concluido el Plan de Trabajo y se archiva.</p> <p>1.8-Se le entrega al Director de Formación en la tercera semana del curso.</p>
Poscondiciones	Queda elaborado el Plan de Trabajo, se hace entrega del mismo.

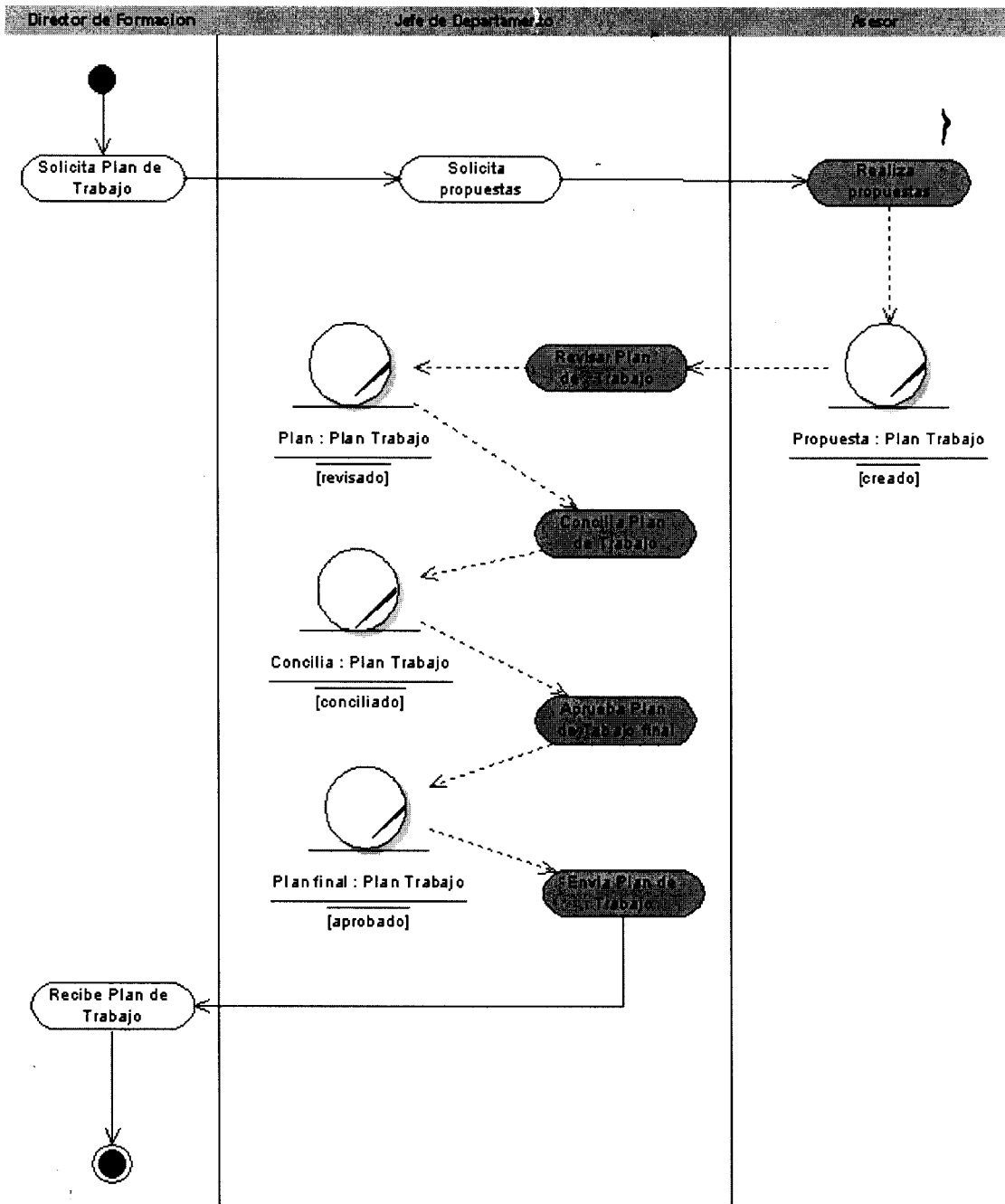


Figura 4. Diagrama del caso de uso: Confeccionar Plan de Trabajo de Asesor.

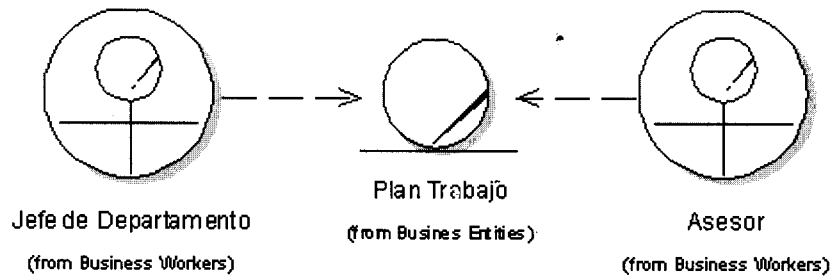


Figura 5. Modelo de objeto del caso de uso: Confeccionar Plan de Trabajo de Asesor.

Tabla 5. Caso de uso: Confeccionar Plantilla del Departamento.

Caso de Uso:	Confeccionar Plantilla del Departamento	
Actores:	Director de Formación.	
Trabajadores:	Jefe de Departamento, Asesores, Jefe de Colectivo	
Resumen:	El caso de uso se inicia cuando el Director de Formación solicita plantilla al Jefe de Dpto., el cual solicita los datos a los Asesores quienes piden datos por Facultad a los Jefe de colectivo (encargados de confeccionar plantilla de asignatura). La misma se envía al Jefe de Dpto. que elabora la plantilla general y es enviada al director de formación.	
Precondiciones:		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1- Solicitar Plantilla al Jefe del Departamento.	1.1-El Jefe de Departamento crea una Plantilla inicial 1.2-Los Jefes de Colectivo elaboran una plantilla por facultad y por asignatura, que contengan todos los datos de sus profesores. 1.4- La plantilla de los Jefes de Colectivo son agrupadas por los asesores para confeccionar la plantilla del departamento. 1.5-El Jefe del Departamento va a confeccionar una plantilla general adicionando algunos datos a los que	

<p>2.-Recibe Plantilla del Departamento.</p>	<p>son plantilla fija del departamento. 1.6-Entrega la plantilla al Director de Formación.</p>
<p>Poscondiciones</p>	<p>Plantilla del departamento confeccionada.</p>

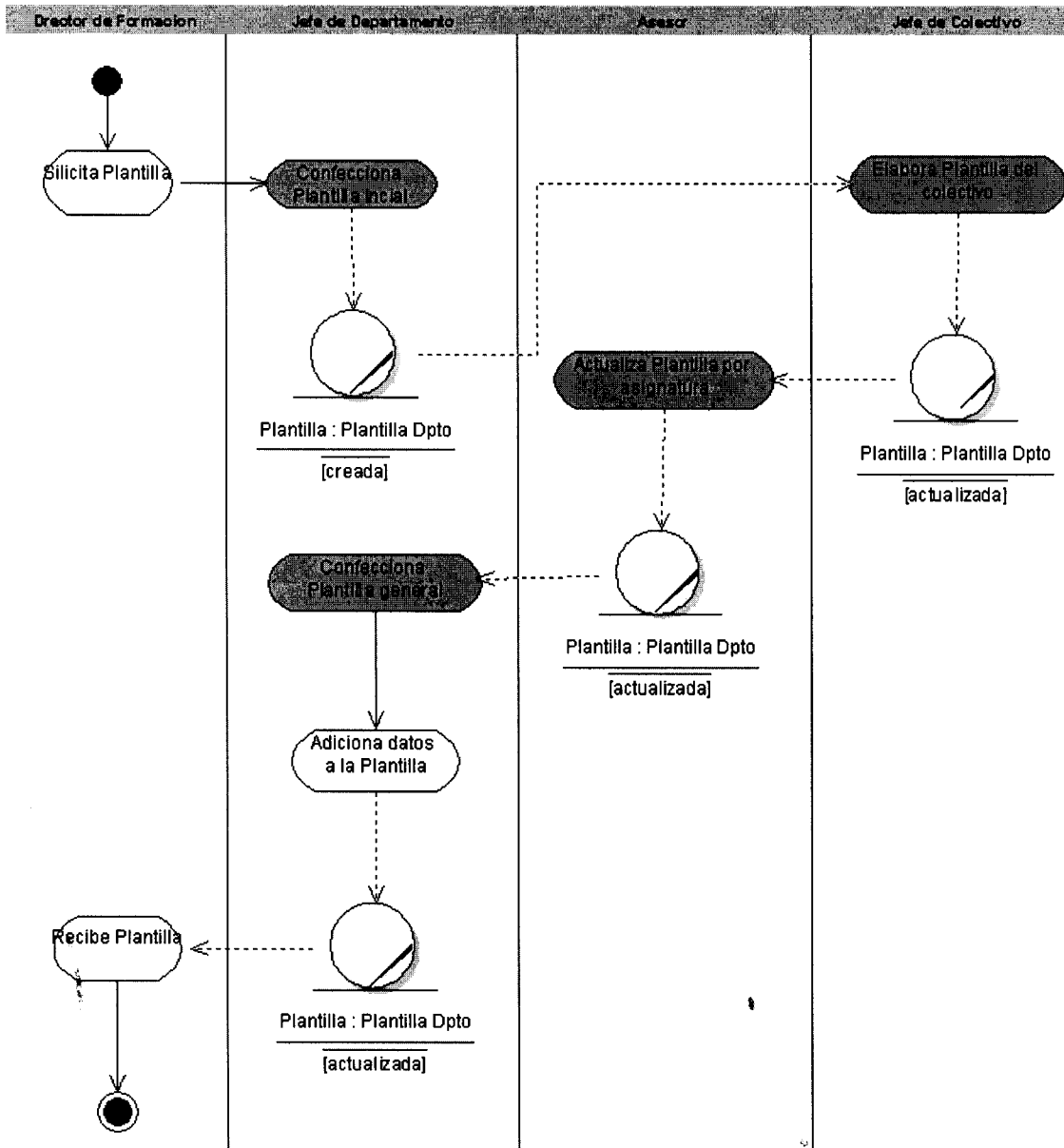


Figura 6. Diagrama de actividad del caso de uso: Confeccionar Plantilla del Departamento.

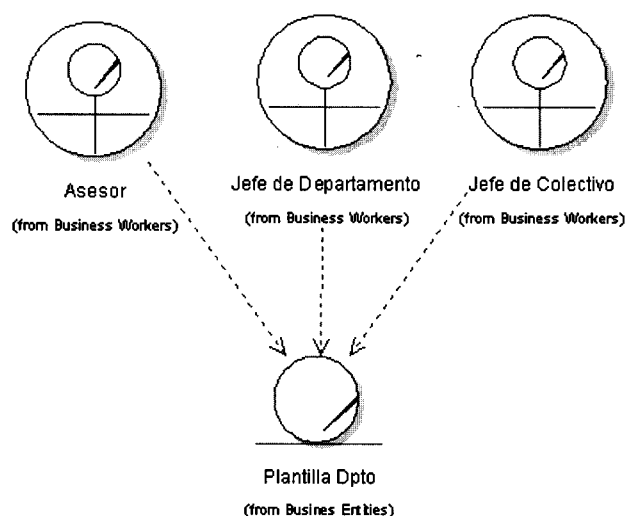


Figura 7. Modelo de objeto del caso de uso: Confeccionar Plantilla del Departamento.

Tabla 6. Caso de uso: Realizar Control a Clases.

Caso de Uso:	Realizar Control a Clases
Actores:	Director de Formación
Trabajadores:	Jefe de Departamento, Asesores
Resumen:	El caso de uso se inicia cuando el Director de Formación solicita el Control a clases al Jefe de Departamento, quien consultando el horario docente conjuntamente con los asesores planifican las actividades de controles a clases donde se especifica la facultad, grupo, asignatura, profesor y turno. Después de efectuado el control a clase se guardan los resultados del control.
Precondiciones:	La Plantilla debe estar terminada.
Flujo Normal de Eventos	

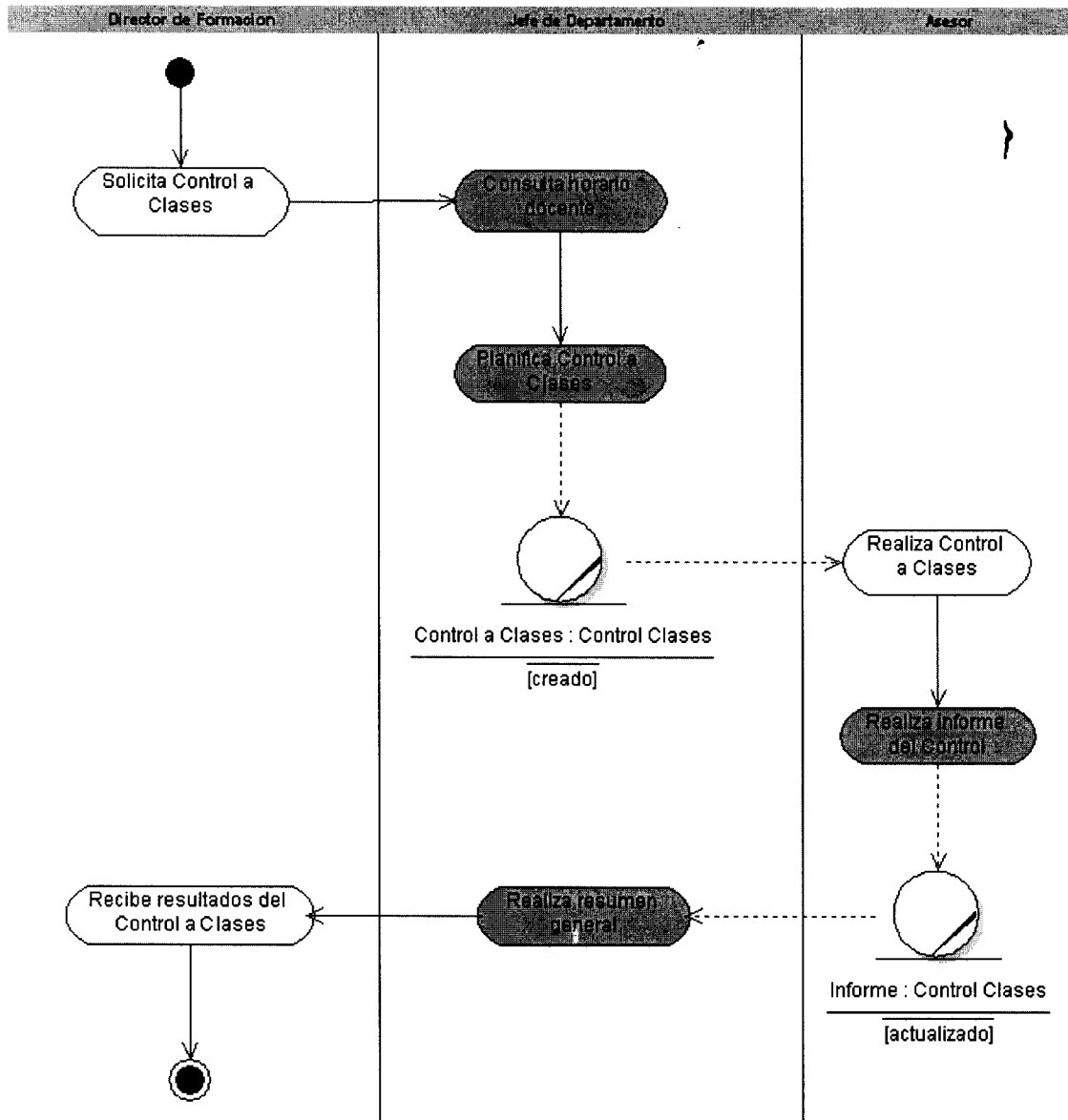


Figura 8. Diagrama de actividad del caso de uso: Realizar Control a Clases.

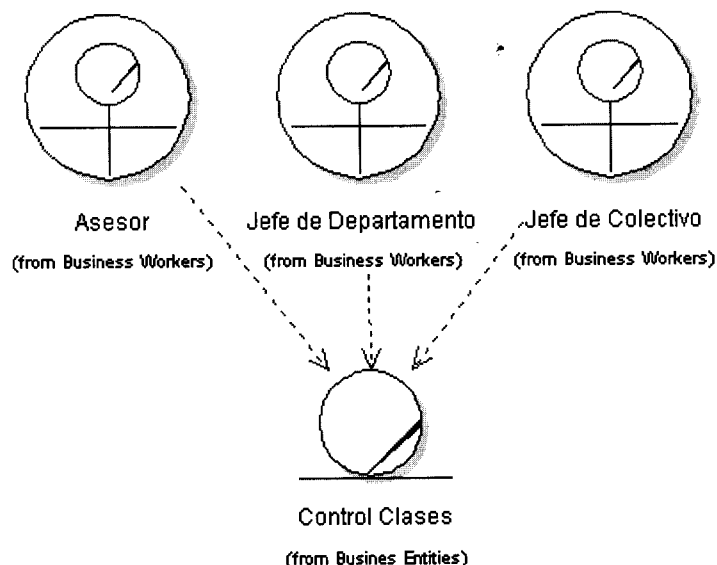


Figura 9. Modelo de objeto del caso de uso: Realizar Control a Clases.

Tabla 7. Caso de uso: Solicitar Materiales.

Caso de Uso:	Solicitar Materiales	
Actores:	Director de Formación.	
Trabajadores:	Jefe de Departamento	
Resumen:	El caso de uso se inicia cuando el Jefe de Departamento necesita solicitar los materiales, y a su vez confecciona un pedido y la justificación, y la entrega al Director de Formación, quien registra la justificación, firma el pedido y se lo entrega al Jefe de Departamento nuevamente, el que envía el pedido al almacén.	
Precondiciones:		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
	1-El Jefe del Departamento confecciona un pedido de materiales.	

<p>4. El Director de Formación al recibir los documentos enviados por el Jefe del Departamento registran la justificación, firma el pedido y se lo envía nuevamente ya firmado.</p>	<p>2-El Jefe de Departamento justifica el pedido de los materiales.</p> <p>3-Ambos documentos son enviados al Director de Formación.</p> <p>4.1 -Se envía el pedido a la Vicerrectoría Docente para su aprobación final.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
4.- Rechaza la solicitud de Materiales.	4.1.-Se archiva la solicitud rechazada.
Poscondiciones	Queda pedido de materiales y justificación lista (firmado) y archivado.

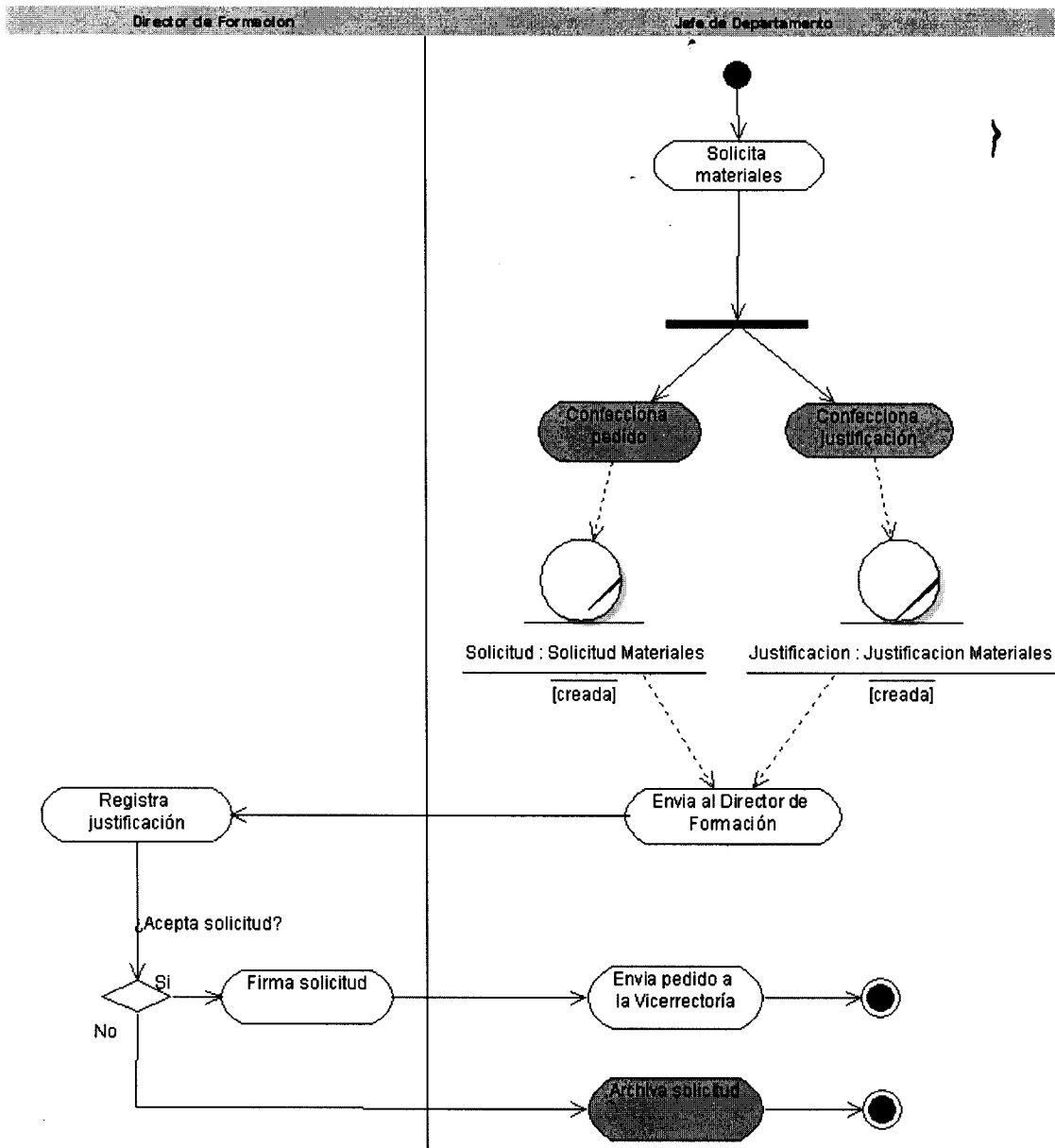


Figura 10. Diagrama de actividad del caso de uso: Solicitar Materiales.

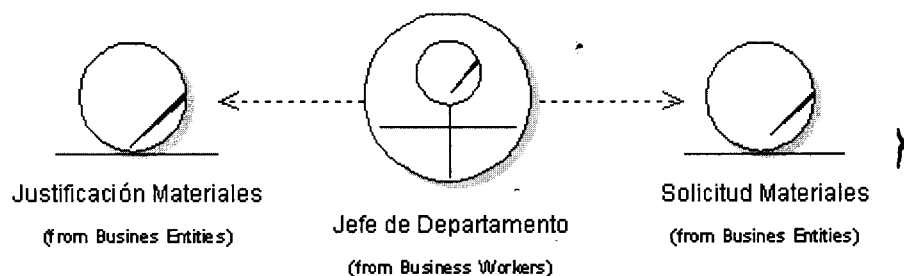


Figura 11. Modelo de objeto del caso de uso: Solicitar Materiales.

Tabla 8. Caso de uso: Confeccionar Prenómina y Aseo

Caso de Uso:	Confeccionar Prenómina y Aseo.	
Actores:	Director de Capital Humano	
Trabajadores:	Jefe de Dpto.	
Resumen:	El caso de uso se inicia cuando el Jefe de Capital Humano solicita la plantilla de Prenómina y Aseo al Jefe de Departamento. Este actualiza los datos de la misma haciendo dos copias y se las envía al Jefe de Capital Humano el cual las firma, archivando este una de las plantillas y devolviendo las copias al jefe del Dpto.	
Precondiciones:		
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Negocio
	1-El Director de Capital Humano solicita la plantilla de Predomina y Aseo al Jefe del Departamento. 2-Concilia la plantilla de Prenómina y Aseo. 3-Firma los documentos.	1.1- Llena la prenómina general. Tiene en cuenta quien falta, por que tiempo, la causa y si le toca aseo. 1.2- Realiza dos copias de la misma enviando el documento y las copias al Director de Capital Humano para que las firme.

<p>4- Archiva el original.</p> <p>5-Envía las dos copias al Jefe de Departamento.</p>	<p>5.1- Envía una copia para solicitud de aseo.</p> <p>5.2- Archiva la otra Copia.</p>
<p>Poscondiciones Queda confeccionada la prenomina.</p>	

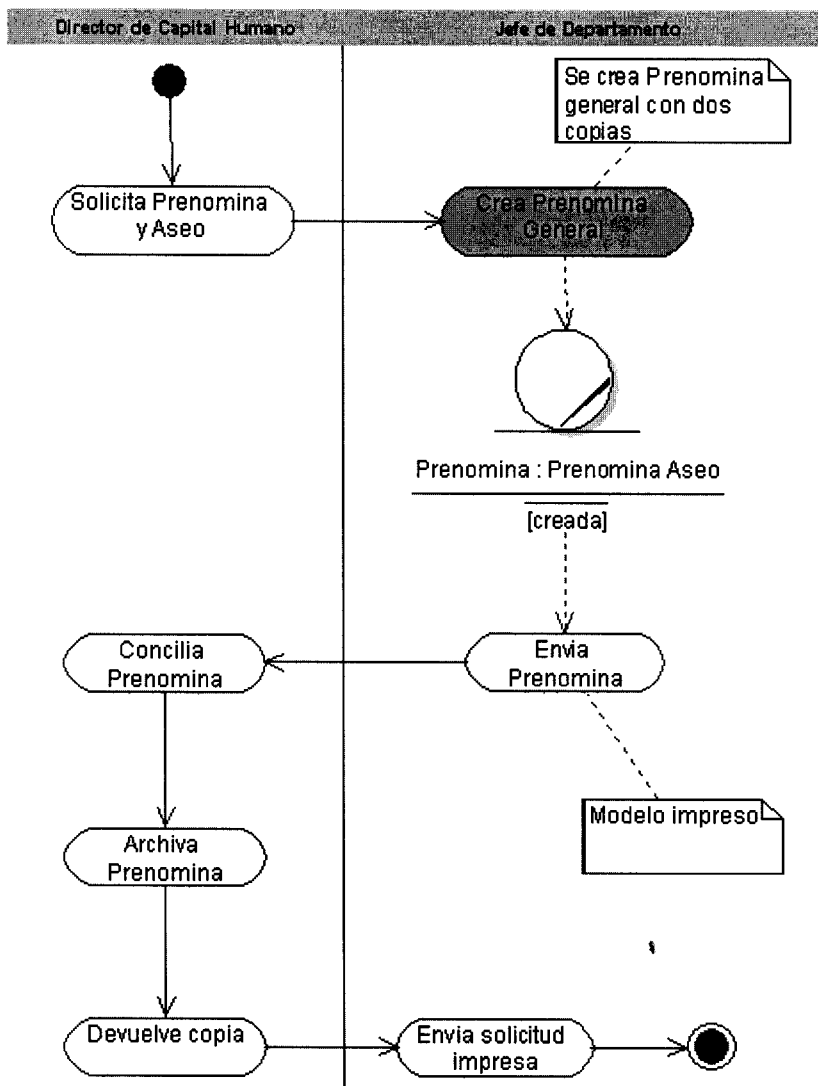


Figura 12. Diagrama de actividad del caso de uso: Confeccionar Prenomina y Aseo.

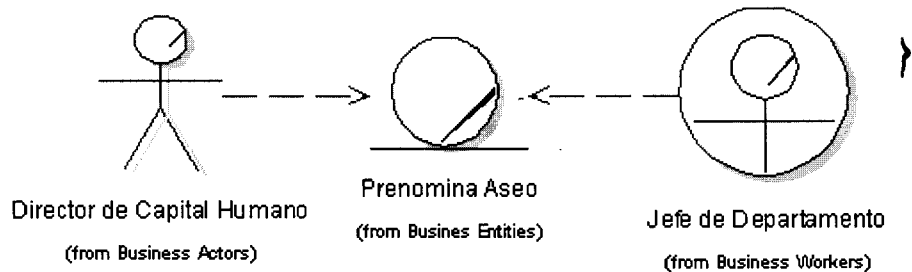


Figura 13. Modelo de objeto del caso de uso: Confeccionar Prenómina y Aseo.

Tabla 9. Caso de uso: Recoger Información.

Caso de Uso:	Recoger Información	
Actores:	Director de Formación	
Trabajadores:	Jefe de Departamento, Jefe de Colectivo	
Resumen:	El caso de uso se inicia cuando el Director de Formación solicita la consolidación de los resultados de la disciplinas al Jefe de Departamento, este solicita los resultados por asignatura a los asesores los cuales solicitan a los colectivos la información. Finalmente el Jefe de Departamento elabora la consolidación y se la entrega al Director de Formación.	
Precondiciones:		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1.-Solicita la consolidación de la disciplina.	1.1.-El Jefe de Departamento solicita la consolidación por asignatura a los asesores. 1.2.-Lo asesores solicitan resultados de su asignatura a los Jefes de Colectivos. 1.3.-El Jefe de Colectivo realiza la	

CAPÍTULO 2 MODELO DEL NEGOCIO

2.-Recibe resultado evaluativo.	recopilación de los resultados. 1.4.-Los asesores elaboran la consolidación de la asignatura y se la envían al Jefe de Departamento. 1.5.-El Jefe de Departamento elabora la consolidación de la disciplina. 1.6.-Envía consolidación al Director de Formación.
Poscondiciones	Queda confeccionada la consolidación de la disciplina.

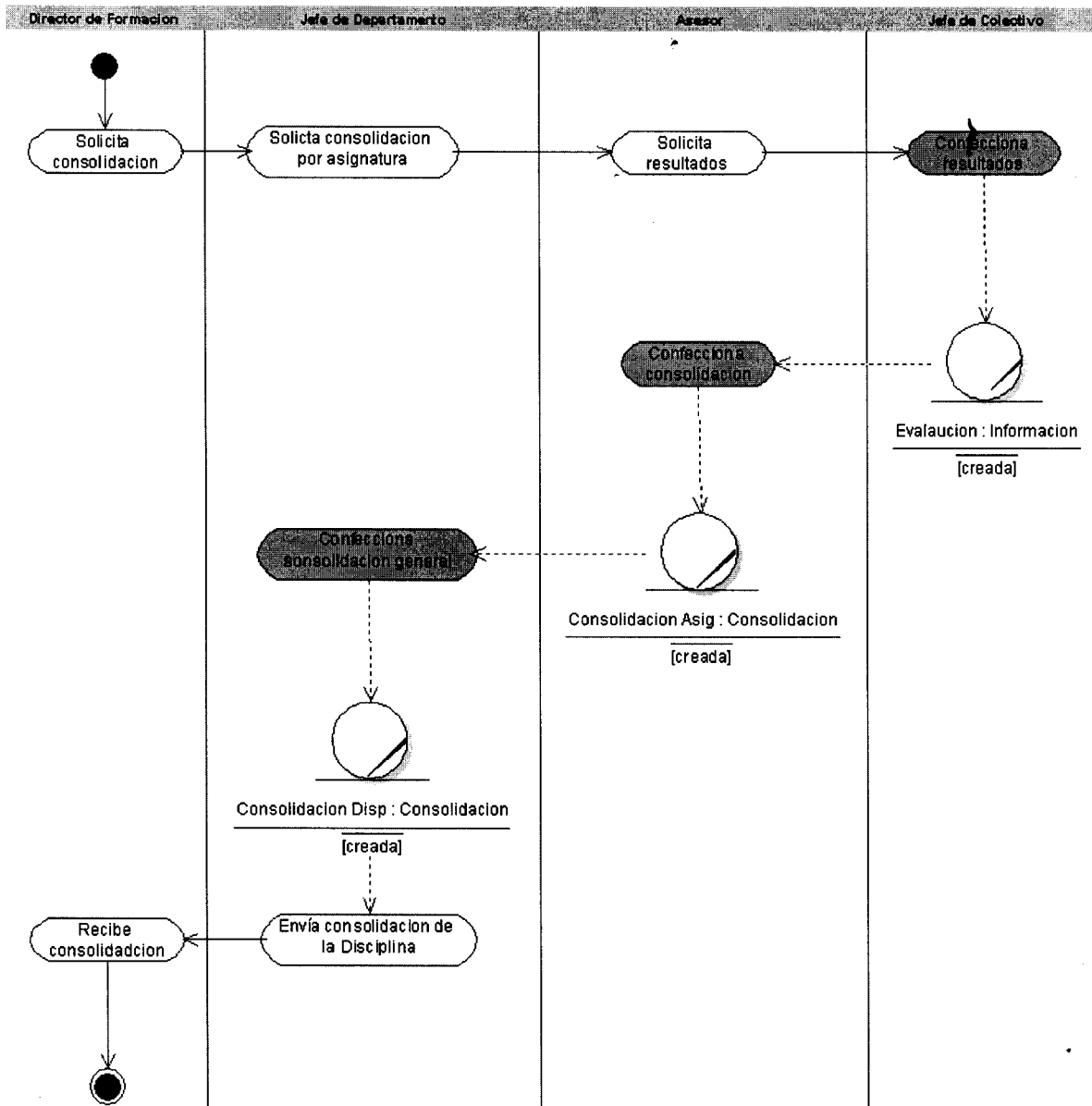


Figura 14. Diagrama de actividad del caso de uso: Recoger Información.

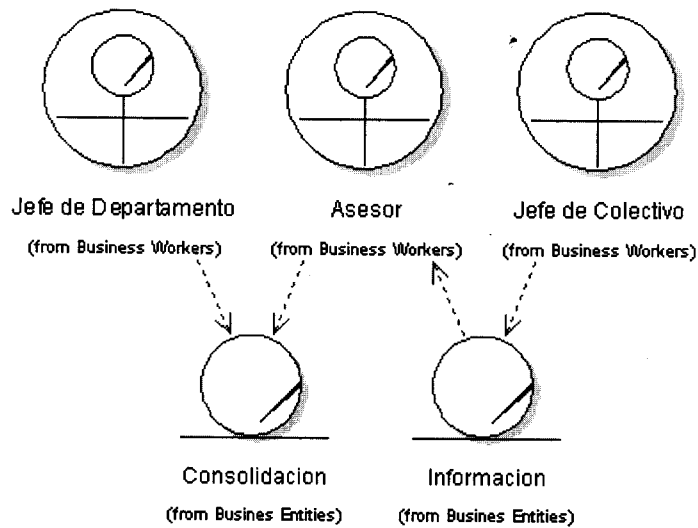


Figura 15. Modelo de objeto del caso de uso: Recoger Información.

2.8 Modelo de objetos

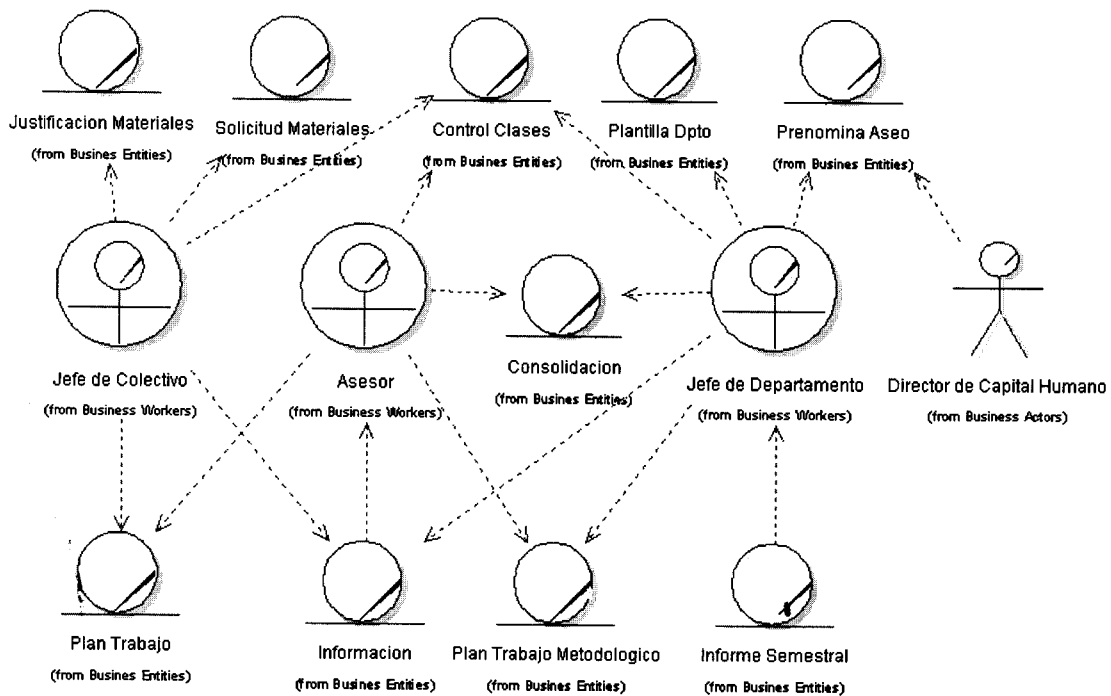


Figura 16. Modelo de objetos.

2.9 Conclusiones

En el presente capítulo se propuso explicar el funcionamiento de los procesos, siguiendo la secuencia lógica de las actividades que se llevan a cabo. Para ello se hizo una explicación textual de cada proceso, así como una representación grafica mediante diagramas de actividad en el modelado de negocio. Con esto se logra conocer el funcionamiento más en detalle del Departamento Docente Central de Ingeniería y Gestión de Software, destacando los procesos que se desean automatizar.



3 REQUISITOS

3.1 Introducción

En el presente capítulo se destacan los requisitos funcionales y no funcionales que debe tener el sistema que se propone, lo que permite hacer una concepción general de los objetivos del sistema, e identificar mediante un Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el sistema, y las secuencias de acciones con las que interactúan.

3.2 Actores del sistema a automatizar

Un actor es una idealización de una persona externa, de un proceso, o de una cosa que interactúa con un sistema, un subsistema, o una clase. Un actor caracteriza las interacciones que los usuarios exteriores pueden tener con el sistema, pueden ser definidos en jerarquías de generalización, en las cuales una descripción abstracta del actor es compartida y aumentada por una o más descripciones específicas del actor. Un actor puede ser un ser humano, otro sistema informático, o un cierto proceso ejecutable. Se dibuja a un actor como una persona pequeña con trazos lineales y el nombre debajo de él. [1]

Tabla 10. Definición de actores del sistema a automatizar

Nombre del actor	Descripción
Metodólogo	Es el encargado de confeccionar y controlar el Plan de Trabajo Metodológico, los Controles a Clases, así como acceder y modificar la plantilla de la disciplina. Las acciones de este actor las podrán ejecutar tanto el Jefe de Departamento como el Asesor.
Jefe de	Es responsable de la confección de la prenomina de pago y

Departamento	aseo, solicitud de materiales, plantilla de la disciplina, así como de chequear y evaluar el cumplimiento de las actividades metodológicas y del asesor.
Asesor	Es el encargado elaborar Plan de Trabajo Individual y darle seguimiento.
Jefe de Colectivo	Es el encargado de la confección y actualización de la plantilla de la disciplina a nivel de colectivo, así como informar acerca del proceso docente.
Usuario	Es aquel en determinado momento solicita entrar al sistema y posterior a eso puede tener acceso determinado contenido, en correspondencia con el rol que este juegue.
Administrador	Es el responsable de instalación y soporte del sistema.

3.3 Definición de los requisitos funcionales

Confeccionar Plan de Trabajo Metodológico (PTM).

- R1.** Mostrar características del claustro como diagnóstico inicial del PTM (total de profesores, total de profesores por categoría docente, cantidad de alumnos ayudantes).
- R2.** Mostrar los informes semestrales de cursos anteriores.
- R3.** Registrar resumen del PTM (objetivos y estrategias).
- R4.** Modificar resumen del PTM.
- R5.** Registrar actividades al PTM (tipo de actividad, descripción, fecha de cumplimiento, responsable, participan, y documento asociado si lo requiere).
- R6.** Notificar por e-mail a usuario involucrado en la actividad.
- R7.** Notificar por e-mail a los asesores para que elaboren propuestas del PTM.
- R8.** Modificar actividades del PTM.
- R9.** Eliminar actividades del PTM.
- R10.** Mostrar PTM.
 - Permitir imprimir.
 - Permitir enviar por e-mail.

Evaluar PTM

R11. Mostrar actividades a realizar en la semana.

R12. Mostrar resumen del cumplimiento de la semana.

R13. Mostrar resumen del cumplimiento de las actividades por tipo de actividad, mostrando total y estado actual.

R14. Verificar estado de cumplimiento de las actividades.

R15. Actualizar estado de cumplimiento de las actividades, los estados pueden ser cumplida, no cumplida, parcialmente cumplida y planificada.

R16. Registrar observación de la evaluación.

R17. Modificar observación de la evaluación.

Gestionar actividades.

R18. Crear actividad definiendo descripción, tipo, fecha de ejecución, responsable, participantes.

R19. Modificar datos de la actividad.

R20. Eliminar actividad.

R21. Asignar documento a la actividad.

R22. Cambiar documento.

Confeccionar Plan de Trabajo Individual (PTI)

R23. Mostrar PTI de un asesor dado.

R24. Agregar, modificar y eliminar aspectos del PTI.

R25. Agregar, modificar y eliminar objetivos por aspecto del PTI.

R26. Introducir y modificar observación por objetivo.

Evaluar asesor

R27. Evaluar la observación del objetivo a cumplido, no cumplido, y parcialmente cumplido.

R28. Modificar evaluación de la observación del objetivo.

R29. Agregar justificación de evaluación.

R30. Modificar justificación de evaluación.

Mostrar PTI

R31. Mostrar resumen cuantitativo del cumplimiento del PTI para un asesor determinado (aspecto, cantidad total de objetivos, cantidad de objetivos cumplidos, cantidad de objetivos no cumplidos y cantidad de objetivos parcialmente cumplidos).

R32. Mostrar resultado detallado por aspecto y persona (objetivo, justificación y estado del objetivo).

R33. Permitir imprimir resumen.

R34. Permitir enviar resumen por e-mail.

Confeccionar plantilla de la disciplina

R35. Determinar datos iniciales del departamento y modificarlos (nombre de la disciplina, curso, fecha de inicio de curso, asignaturas).

R36. Insertar Jefe de Departamento, asesores, Jefes de Colectivo y secretaria.

- Datos generales (nombre y apellidos, correo-e, número de solapín, dirección particular, teléfono, cargo)
- Datos docentes (asignatura que imparte, grupo(s), facultad, categoría docente, grado científico).

R37. Notificar a asesores y jefes de colectivo por e-mail que deben confeccionar la plantilla.

Confeccionar plantilla por colectivo

R38. Introducir, modificar y eliminar profesores al colectivo.

R39. Asignar datos de cada profesor en el colectivo.

- Datos generales.
- Datos docentes.

R40. Subir horarios docentes por facultad.

Mostrar Profesores

R41. Mostrar listado de los profesores según la asignatura que imparten.

R42. Mostrar listado de profesores según la facultad a la que pertenecen.

R43. Mostrar listado de todos los profesores, con todos los datos correspondientes a los mismos.

R44. Mostrar listado de profesores según las habilidades que poseen.

R45. Permitir Imprimir el listado obtenido a partir de lo solicitado a mostrar.

Planificar Controles a Clases.

- R46. Consultar horario docente de una facultad.
- R47. Agregar control a clase (se define fecha, facultad, turno, grupo, y profesor).
- R48. Modificar control a clase, esto incluye actualizar y eliminar.
- R49. Chequear cumplimiento de la planificación de controles a clases.
- R50. Introducir evaluación donde se destacan los aspectos positivos y negativos del control a clase.

Mostrar resultados de Controles a Clases

- R51. Mostrar cantidad de controles a clases realizados.
- R52. Mostrar resumen de controles a clases destacando aspectos positivos y negativos, por facultad.
- R53. Mostrar resumen de controles a clases destacando aspectos positivos y negativos, por facultad y asignatura.
- R54. Mostrar observaciones de los controles a clases, destacando aspectos positivos y negativos en un periodo dado.

Realizar Solicitud de Materiales.

- R55. Confeccionar planilla de solicitud de materiales según clasificación de de los materiales (de oficina o técnicos) donde cada material tenga descripción, unidad de medida, cantidad, y justificación.
 - Agregar materiales.
 - Eliminar materiales.
 - Actualizar materiales y demás datos de la solicitud.
- R56. Confeccionar planilla de justificación de los materiales la cual contendrá la fecha, quien la elabora, a quien va dirigida, así como materiales con su justificación.
- R57. Mostrar planilla de solicitud además de permitir imprimir y mandar por e-mail.
- R58. Mostrar planilla de justificación además de permitir imprimir y mandar por e-mail.
- R59. Guardar el real de materiales aprobados de la solicitud.
- R60. Mostrar reporte de materiales solicitados contra aceptados, por solicitud y periodo.
- R61. Eliminar registros de solicitudes y justificaciones.

Confeccionar Prenómina y Aseo.

R62. Confeccionar la planilla de pre Nómina y aseo conteniendo la siguiente información: quien la elabora, quien la aprueba, dependencia, entidad, organismo, fecha (periodo); de los trabajadores solapín, nombre, cargo, horas de trabajo extra, horas de doble turnos, días feriados, si recibe aseo o no, y la cantidad de hombre y cantidad de mujeres que reciben aseo mas el total.

R63. Modificar relación de trabajadores en la pre Nómina (adicionar y eliminar).

R64. Mostrar planilla de pre Nómina.

R65. Permitir imprimir la planilla de pre Nómina.

R66. Permitir enviar la planilla de pre Nómina por e-mail

R67. Permitir guardar la planilla.

Gestionar noticias.

R68. Publicar noticias.

R69. Modificar noticias.

R70. Eliminar noticias publicadas.

R71. Reemplazar noticias.

Autenticar usuario

R72. Verificar que el usuario este registrado en el sistema.

R73. Verificar usuario y contraseña en el dominio UCI.

R74. Verificar usuario y contraseña en el sistema.

3.4 Diagrama de casos de uso del sistema a automatizar

Un caso de uso es una unidad coherente de funcionalidad, externamente visible, proporcionada por una unidad del sistema y expresada por secuencias de mensajes intercambiados por la unidad del sistema y uno o más actores. El propósito de un caso de uso es definir una pieza de comportamiento coherente, sin revelar la estructura interna del sistema. La definición de un caso de uso incluye todo el comportamiento que implica: las líneas principales, las diferentes variaciones sobre el comportamiento normal, y todas las condiciones excepcionales, que pueden ocurrir con tal comportamiento, junto con la respuesta deseada. [1]

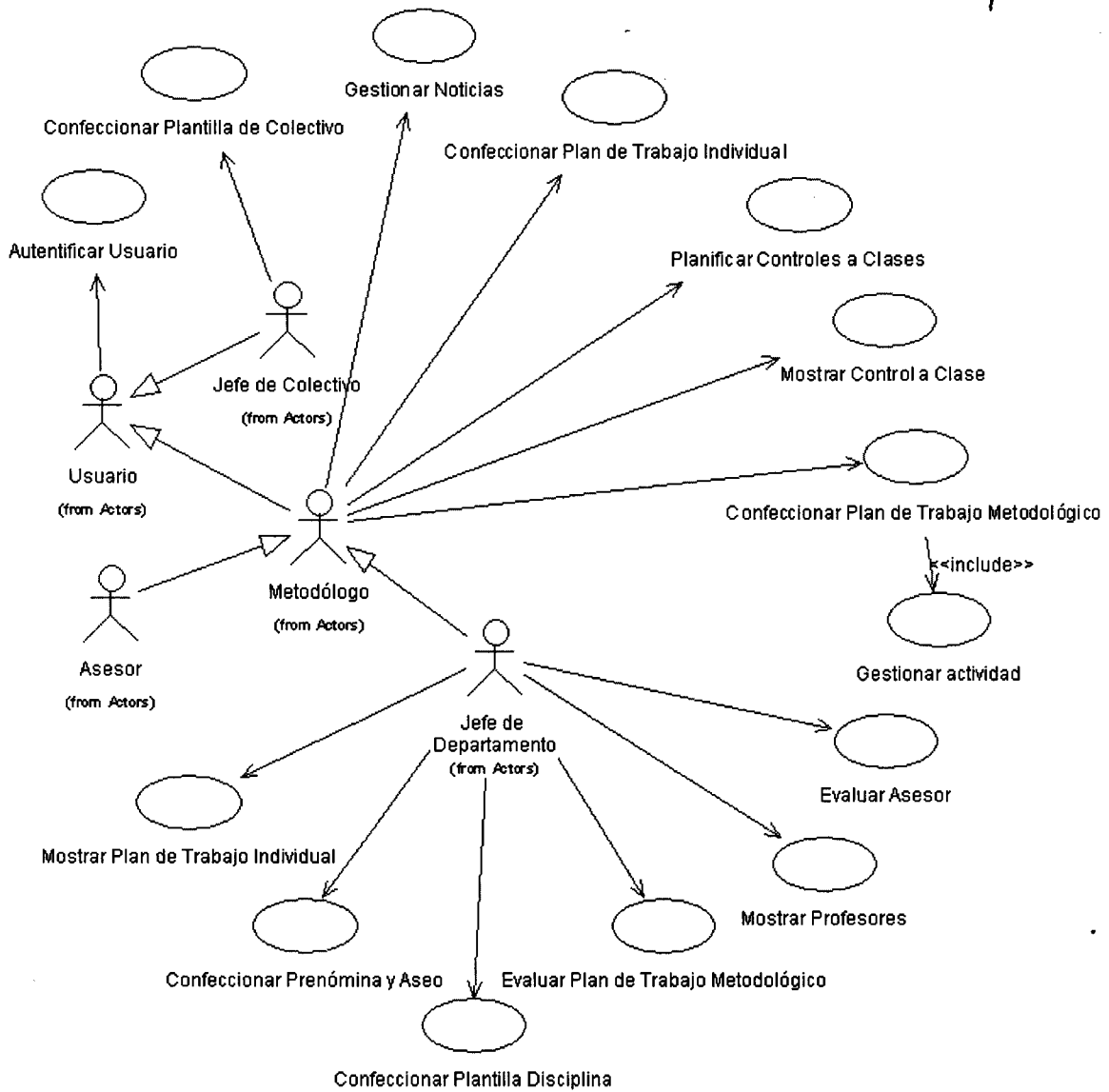


Figura 17. Diagrama del casos de uso del sistema.

3.5 Definición de los requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Apariencia o interfaz externa:

- Diseño orientado a llamar la atención del usuario y con una navegación sencilla.
- Construcción de enlaces rápidos o anclas para los documentos muy largos.

Usabilidad:

- El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

Rendimiento:

- Tiempos de respuestas no mayor de 2 segundos, al igual que la velocidad de procesamiento de la información.

Soporte:

- Se requiere un servidor de bases de datos con las siguientes características:
 - Soporte para medianos volúmenes de datos y velocidad de procesamiento.
 - Tiempo de respuesta de no más de 3 segundos en accesos concurrentes.
- Versión de PHP 5.0.x o superior.
- Servidor Web Apache 2.0.x o superior
- Por parte del cliente se requiere un navegador capaz de interpretar JavaScript.

Portabilidad:

- Necesidad de que el sistema sea multiplataformas.

Seguridad:

- Identificar al usuario antes de que pueda realizar cualquier acción sobre el contenido del sistema.
- Garantizar que la información sea editada únicamente por quien tiene derecho a editarla.

- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que este activo.
- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- Verificación sobre acciones irreversibles (eliminaciones).

Legales:

- La plataforma escogida para el desarrollo de la aplicación, está basada en la licencia GNU/GPL.

Confiabilidad:

- La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.

Funcionalidad:

- Reducir al mínimo el tiempo en que carga el sistema.
- Guardar en caché páginas de contenido para agilizar la navegación del portal.

Software:

- Navegador compatible o superior con Internet Explorer 4, o NetsCape Navegador.
- Macromedia Dreamweaver MX 2004.
- MySQL 2.3.x.
- Apache 2.0.x
- PHP 5.0.x

3.6 Descripción de los casos de uso

Tabla 11. Confeccionar Plan de Trabajo Metodológico

CU-1	Confeccionar Plan de Trabajo Metodológico
Actor	Metodólogo
Resumen	El caso de uso se inicia cuando el Metodólogo, en este caso el Jefe de Departamento consulta información del claustro para confeccionar objetivos y estrategias del PTM. Define actividades notificando a involucrados para que participen en la planeación y ejecución. Finalmente el PTM se imprime o envía por e-mail.

Referencia	R1, R2, R3, R4, R5, R6, R7, R8, R9 y R10 Casos de uso asociados: <ul style="list-style-type: none"> • Gestionar Actividad (Include).
Precondiciones	Debe estar conformada la Plantilla de la Disciplina.
Poscondiciones	Las actividades van a estar en un estado de planificada mientras no se cumpla la fecha de ejecución, luego pueden pasar a cumplidas, parcialmente cumplidas o no cumplidas.
Requisitos especiales	

Tabla 12. Evaluar Plan de Trabajo Metodológico

CU-2	Evaluar Plan de Trabajo Metodológico
Actor	Jefe de Departamento
Resumen	El caso de uso se inicia cuando el Jefe de Departamento decide verificar el estado de cumplimiento de las actividades del Plan de Trabajo Metodológico, chequeando resultados de las actividades asta el momento y emitiendo una observación que justifica el estado asignado.
Referencia	R11, R12, R13, R14, R15, R16 y R17
Precondiciones	Debe haber pasado la fecha de ejecución de alguna actividad.
Poscondiciones	Cambia el estado de las actividades después de pasado el tiempo de ejecución.
Requisitos especiales	

Tabla 13. Gestionar actividad.

CU-3	Gestionar Actividad
Actor	Metodólogo
Resumen	El caso de uso se inicia cuando se introduce alguna actividad al Plan de Trabajo Metodológico, el cual permite crear, modificar y eliminar actividades.
Referencia	R18, R19, R20, R21 y R22
Precondiciones	Debe estar confeccionada la plantilla de la disciplina.

Poscondiciones	
Requisitos especiales	

Tabla 14. Confeccionar plan de trabajo individual.

CU-4	Confeccionar Plan de Trabajo Individual
Actor	Metodólogo
Resumen	El caso de uso se inicia cuando el Metodólogo, que puede ser un Asesor o el Jefe de Departamento, consulta el Plan de Trabajo Individual de algún asesor y decide modificarle los objetivos y observaciones. Después de periodo de confección este no puede ser modificado.
Referencia	R23, R24, R25 y R26
Precondiciones	Se debe constar con la plantilla de la disciplina confeccionada.
Poscondiciones	--
Requisitos especiales	--

Tabla 15. Evaluar asesor.

CU-5	Evaluar Asesor
Actor	Jefe de Departamento
Resumen	El caso de uso se inicia cuando el Jefe de Departamento consulta el cumplimiento de los objetivos del PTI de algún asesor, evaluando el cumplimiento y emitiendo una observación.
Referencia	R27, R28, R29 y R30
Precondiciones	Para efectuar esta evaluación debe existir algún PTI previamente planificado.
Poscondiciones	Esto puede cambiar el estado de cumplimiento de los objetivos plasmados en la planificación del PTI.
Requisitos especiales	

Tabla 16. Mostrar plan de trabajo individual.

CU-6	Mostrar Plan de Trabajo Individual
Actor	Jefe de Departamento
Resumen	El caso de uso se inicia cuando el Jefe de Departamento visualiza resúmenes del estado final del PTI, donde finalmente puede imprimir o enviar por e-mail a otros interesados.
Referencia	R31, R32, 33 y R34
Precondiciones	Se debe haber planificado algún PTI previamente.
Poscondiciones	
Requisitos especiales	

Tabla 17. Confeccionar plantilla de la disciplina.

CU-7	Confeccionar plantilla de la disciplina
Actor	Jefe de Departamento
Resumen	El caso de uso se inicia cuando el Jefe de Departamento define los datos iniciales del departamento (nombre disciplina, curso, fecha inicio curso, asignaturas), asesores, y Jefes de Colectivos para conformar una estructura de roles.
Referencia	R35, R36 y R37
Precondiciones	El sistema o un administrador deben haber definido al Jefe de Departamento, que es el usuario de mayor responsabilidad y peso en el sistema.
Poscondiciones	
Requisitos especiales	

Tabla 18. Confeccionar plantilla de Colectivo.

CU-8	Confeccionar Plantilla de Colectivo
Actor	Jefe de Colectivo
Resumen	El caso de uso se inicia cuando el Jefe de Colectivo registra los profesores del colectivo, definiendo sus datos personales si no esta registrado, profesionales y docentes.
Referencia	R38, R39 y R40

Precondiciones	El Jefe de Colectivo debe haber sido previamente notificado de su rol en el sistema.
Poscondiciones	
Requisitos especiales	

Tabla 19. Mostrar profesores.

CU-9	Mostrar Profesores
Actor	Metodólogo
Resumen	El caso de uso se inicia cuando el Metodólogo intenta mostrar reportes con información filtrada de los profesores, la que puede ser impresa o enviada por e-mail.
Referencia	R41, R42, R43, R44 y R45
Precondiciones	Debe estar confeccionada la Plantilla de la Disciplina.
Poscondiciones	
Requisitos especiales	

Tabla 20. Planificar controles a clases.

CU-10	Planificar Controles a Clases
Actor	Metodólogo
Resumen	El caso de uso se inicia cuando el Metodólogo agrega una nueva planificación de control a clases, también puede iniciarse al actualizar planificaciones después del cumplimiento, donde se señalan aspectos positivos y negativos.
Referencia	R46, R47, R48, R49 y R50
Precondiciones	Necesita que la plantilla de la disciplina este confeccionada.
Poscondiciones	
Requisitos especiales	

Tabla 21. Mostrar resultado de controles a clases.

CU-11	Mostrar resultados de Controles a Clases
Actor	Metodólogo
Resumen	El caso de uso se inicia cuando el Metodólogo desea visualizar resultados de los controles a clases, estos son mostrados por criterios que el actor debe seleccionar como por facultad, asignatura, en un periodo y combinaciones entre estos, siempre destacando aspectos positivos y negativos.
Referencia	R51, R52, R53 y R54
Precondiciones	Para poder mostrar algún resultado de control a clase deben existir controles planificados.
Poscondiciones	
Requisitos especiales	

Tabla 22. Realizar solicitud de materiales.

CU-12	Realizar Solicitud de Materiales
Actor	Jefe de Departamento
Resumen	El caso de uso se inicia cuando el Jefe de Departamento hace solicitud de materiales a la Vicerrectoría de Formación. Se elabora una justificación por cada material que se solicite, la cual es enviada con la solicitud.
Referencia	R55, R56, R57, R58, R59, R60 y R61
Precondiciones	
Poscondiciones	
Requisitos especiales	

Tabla 23. Confeccionar prenómina y aseo.

CU-13	Confeccionar Prenómina y Aseo
Actor	Jefe de Departamento
Resumen	El caso de uso se inicia cuando el Jefe de Departamento confecciona

	la pre Nómina de pago y aseo, a la que puede modificar los trabajadores que la componen así como imprimir el modelo, enviarlo por correo, y guardarlo en el sistema.
Referencia	R62, R63, R64, R65, R66 y R67
Precondiciones	Debe estar confeccionada la plantilla del departamento.
Poscondiciones	Queda archivada la pre Nómina de pago y aseo.
Requisitos especiales	A este caso de uso solo puede acceder el Jefe de Departamento.

Tabla 24. Gestionar noticias.

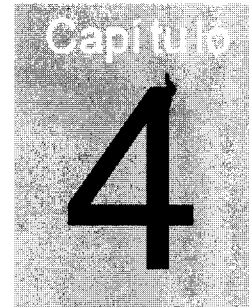
CU-15	Gestionar noticias
Actor	Jefe de Departamento, Asesor, Jefe de Colectivo.
Resumen	Este caso de uso se inicia cuando uno de los actores concebido como Metodólogo, edita o publica noticias en el sistema, esto incluye actualización y eliminación. Permite además mostrar las noticias publicadas al mes correspondiente.
Referencia	R68, R69, R70 y R71
Precondiciones	
Poscondiciones	

Tabla 25. Autenticar usuario.

CU-16	Autenticar usuario
Actor	Usuario
Resumen	El caso de uso se inicia cuando el usuario solicita entrar al sistema, introduciendo nombre y contraseña, verificando sus credenciales en el dominio uci.cu para permitir el acceso. En caso de que no pueda autenticar en el dominio entonces lo intenta en el sistema, consultando su perfil de usuario.
Referencia	R72, R73 y R74
Precondiciones	El usuario debe haber sido adicionado previamente al sistema.
Poscondiciones	
Requisitos especiales	Solo tiene acceso a la información y recursos definido para el rol que juega en el sistema.

3.7 Conclusiones

En este capítulo se determinaron los requisitos que debe cumplir el sistema, se representaron mediante un Diagrama de Casos de Uso, y finalmente se describieron paso a paso todas las acciones de los actores del sistema con los casos de uso con los que interactúan. Con este análisis ahora se puede continuar con el diseño que se propondrá, tratando de que se cumplan todos los requerimientos y las funciones que han sido consideradas necesarias.



4 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

4.1 Introducción

En el presente capítulo se realiza el diseño de la propuesta de solución, seleccionado una arquitectura y modelándose los artefactos que contribuyen al desarrollo de aplicaciones Web, con el uso de Frameworks Manipuladores de Contenidos flexibles.

Se elabora el modelo de datos adecuado, se especifican los principios de diseño gráfico, los estándares de la interfaz de aplicación, así como una concepción general del tratamiento de excepciones que se ofrece al usuario.

4.2 Descripción de la arquitectura

El Sistema de Gestión Metodológica Central ha sido desarrollado sobre la plataforma Drupal. La arquitectura de Drupal está altamente compartimentada en módulos, esto le permite una gran flexibilidad y extensibilidad. A continuación se representa una vista de global de la arquitectura.

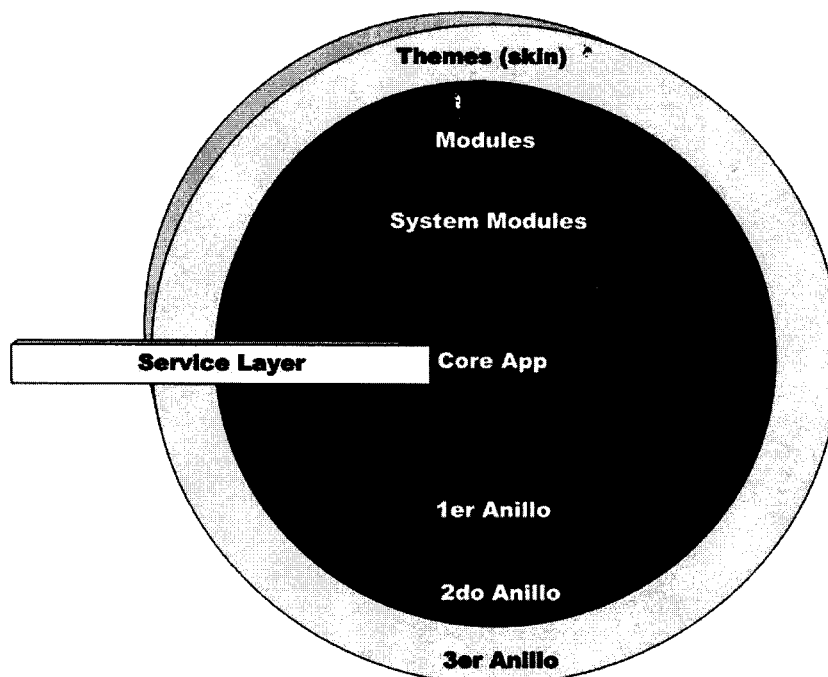


Figura 18. Arquitectura de Drupal.

El framework cuenta con cinco paquetes fundamentales:

1. **Core Application** (Aplicación del núcleo)
2. **System Modules** (Módulos de sistema)
3. **Modules** (Módulos adicionales)
4. **Themes** (Plantillas o Skins)
5. **Service Layer** (Capa de servicio)

Aunque se les diga capas no cumple justamente con la arquitectura en capas, sino que se comporta como anillos con diferentes niveles de prioridad. Esta arquitectura adoptada por el equipo de desarrollo de Drupal tiene un alto prestigio entre las aplicaciones al estilo de drupal en las comunidad de software libre, puesto que ha demostrado al lo largo del tiempo que propicia una elevada flexibilidad y fácil extensibilidad al sistema.

Core Application: La aplicación del núcleo es como tal el corazón de Drupal pero que por si sola no es capas de realizar ninguna función. La razón de esto es la arquitectura orientada a componentes que caracteriza al sistema, llevando todas las funcionalidades del sistema a componentes manejables que no forman parte del núcleo

de la aplicación. El objetivo del núcleo es más bien la orquestación de todos los módulos que componen el sistema y de su integración con la capa de servicios y de presentación.

System Modules: Conjunto de módulos que son requeridos por el sistema y que está en el primer anillo que rodea el núcleo de la aplicación. Estos módulos brindan funcionalidades básicas al sistema como son:

1. **Control de acceso:** Gestión de usuarios, roles y permisos sobre roles. Este módulo es completamente extensible por terceros módulos, por ejemplo existen implementaciones para la integración con LDAP.
2. **Gestión de contenido Web:** Permite cubrir completamente las necesidades de gestión de información de cualquier tipo gracias a un sistema único por su extensibilidad y posibilidad de categorización.
3. **Registro de eventos:** De forma automática se registran y clasifican todos los eventos que ocurren en el sistema.

Modules: Los representa el segundo anillo de la aplicación y se encargan de extender el sistema e implementar nuevas funcionalidades que no estaban previstas en el framework. Para ello cuentan con la completa colaboración de la capa de servicios con su poderoso API y de los módulos del primer anillo que les permiten abstraerse de los temas de seguridad, control de acceso, registro de eventos y otros.

Service Layer: La capa de servicio no implementa lógica de negocio alguna, sin embargo es de vital importancia en un sistema Drupal. El objetivo de la capa de servicios es proveer a los módulos de un completo y poderoso conjunto de APIs que les permitirán abstraerse de dificultades técnicas y poder concentrarse en la implementación de la lógica de negocio.

Componentes de la capa de servicio más significativos:

1. **Theme:** Brinda un API completo de funciones para la generación de código XHTML compatible a partir de datos estructurados.
2. **Form:** Brinda un API para la generación, validación y control de formularios complejos.
3. **File-Management:** API para el manejo de ficheros.

4. **Image:** API para el trabajo con imágenes.
5. **XML-RPC y SOAP:** API completo para construir o usar servicios web (Web Services).
6. **Database-Bridge:** API para el acceso a bases de datos. Funciona como punto de acceso único a la base de datos permitiendo hacer puentes a diferentes implementaciones para diferentes gestores de base de datos. Asta el momento para MySQL y PostgresSQL.

Themes: El sistema de temas separa el contenido de la presentación permitiendo controlar o cambiar fácilmente el aspecto del sitio web, incluyendo entre otras muchas cosas, los colores, los tipos de letra y la ubicación del contenido mostrado en las páginas. Se pueden crear plantillas con HTML y/o con PHP. []

Este documento se limitará a describir los módulos que fueron desarrollados específicamente para la aplicación, los cuales se encuentran dentro del paquete "Modules". Para una mejor organización y entendimiento de las partes del sistema se da una vista de paquetes de la arquitectura, centrando la atención en el paquete Gestión Metodológica Central (GMC), donde esta contenida la solución propuesta por este trabajo.

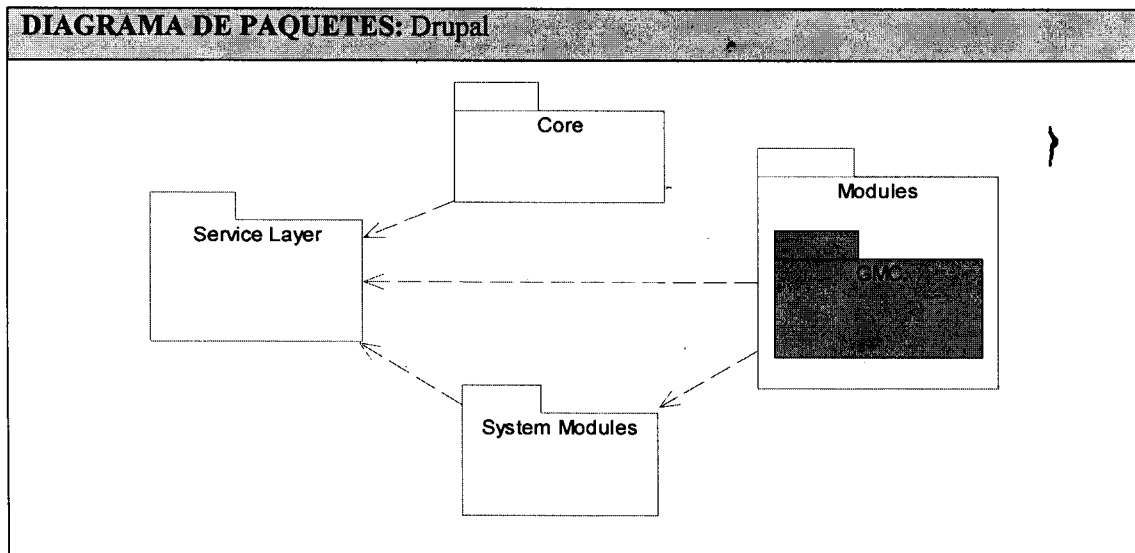


Figura 19. Estructura de paquetes del sistema.

Se representan en color mas fuerte los artefactos que componen el sistema propuesto, además se incluyen en el diagrama los paquetes de los que depende el subsistema para facilitar la comprensión del mismo.

4.3 Diagrama de clases del diseño

El diagrama de clases para las Aplicaciones Web, difiere un poco del resto de las aplicaciones que estamos acostumbrados a construir, puesto que en ellas son más importantes la modelación de la lógica y estado del negocio que los detalles de presentación. Para obtener un nivel correcto de abstracción y detalle que nos permita obtener un resultado final, es mejor modelar los artefactos del sistema, es decir, modelar las páginas, los enlaces entre estas, así como el contenido dinámico de estas, una vez que estén en el navegador del cliente; estos son los artefactos que necesitamos modelar para la implementación del producto final.

De acuerdo a la forma en que hemos organizado el contenido del trabajo, consideramos que se debe presentar los modelos organizados por paquetes.

Subsistema Gestión Metodológica Central (GMC)

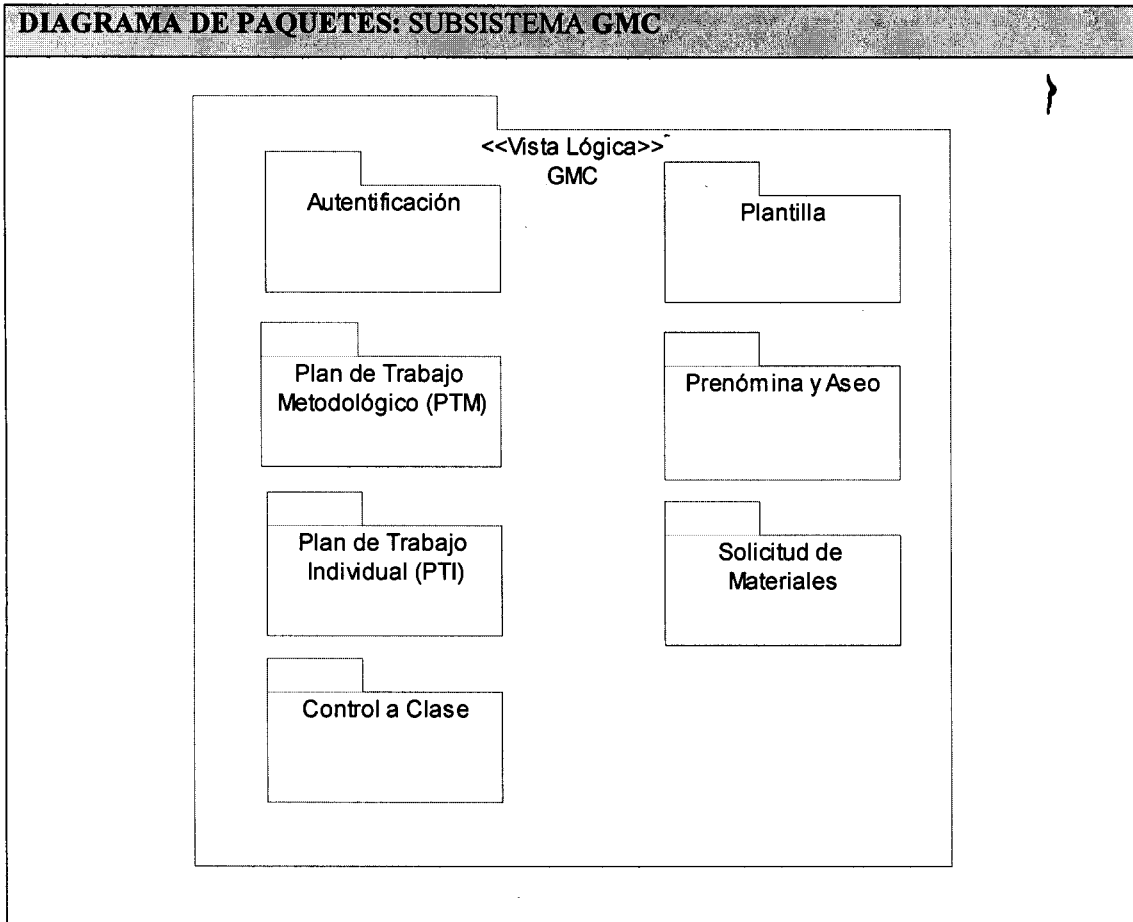


Figura 20. Diagrama de paquetes del subsistema GMC.

4.3.1 Paquete <Autenticación >

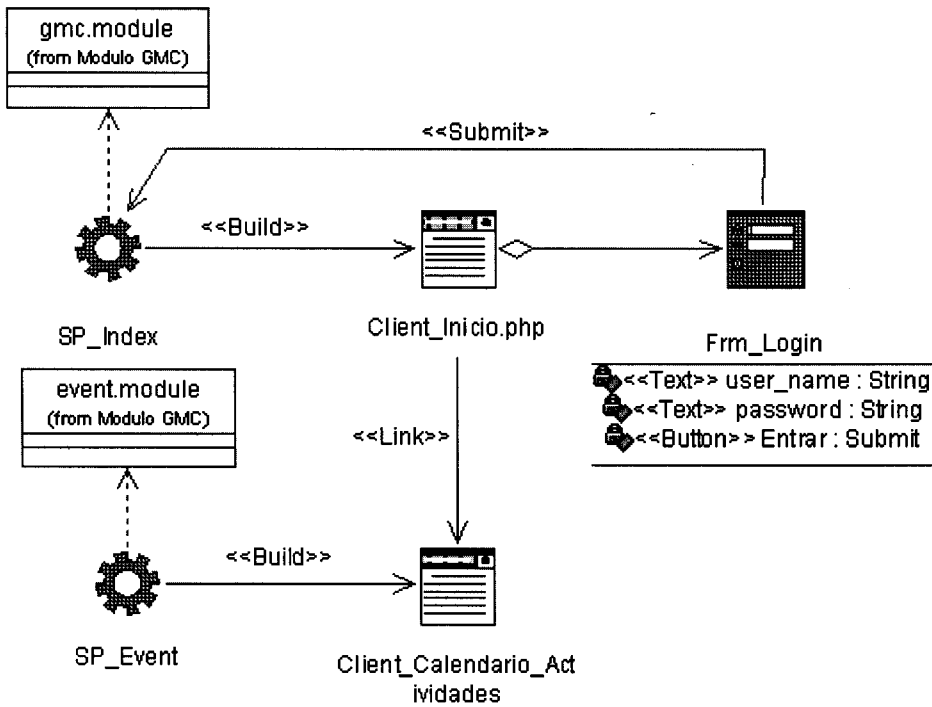


Figura 21. Diagrama de clases web del paquete <Autenticación>.

4.3.2 Paquete <Prenómina y Aseo>

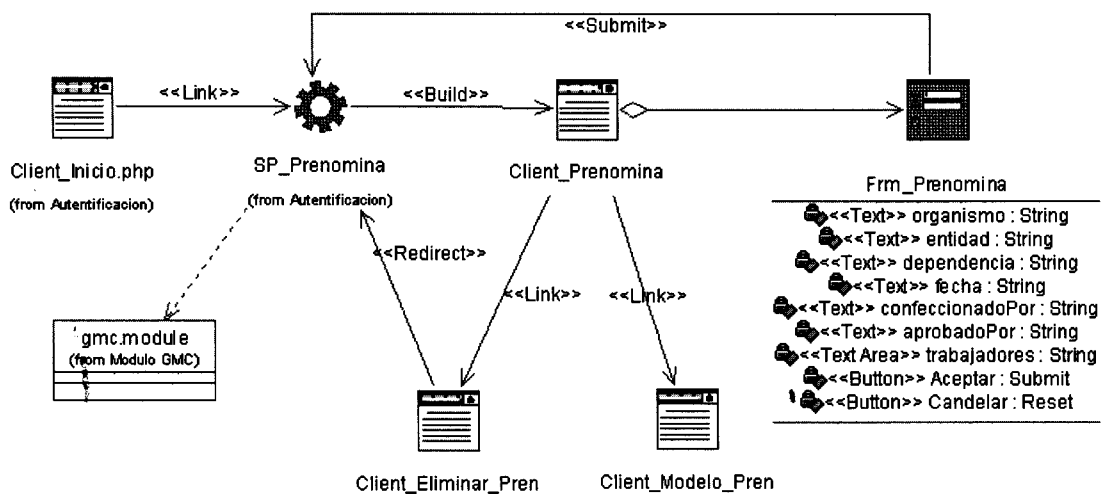


Figura 22. Diagrama de clases web del paquete <Prenómina y Aseo>.

4.3.3 Paquete <Solicitud de Materiales>

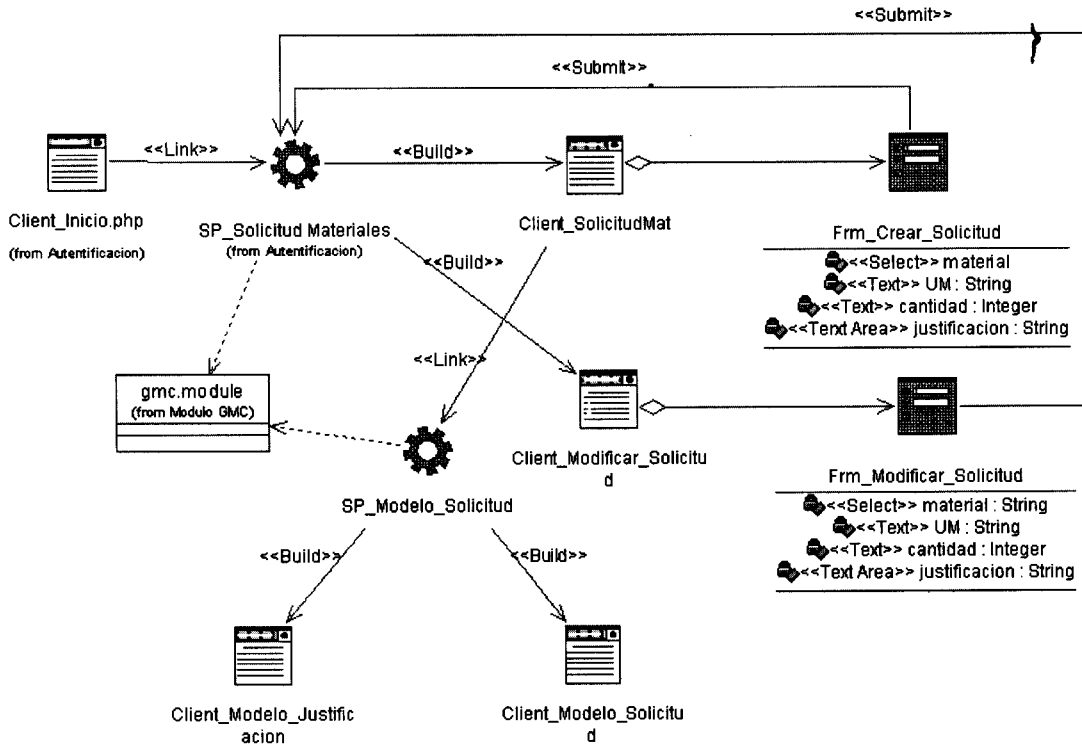


Figura 23. Diagrama de clases web del paquete <Solicitud de Materiales>.

4.3.4 Paquete <Control a Clases>

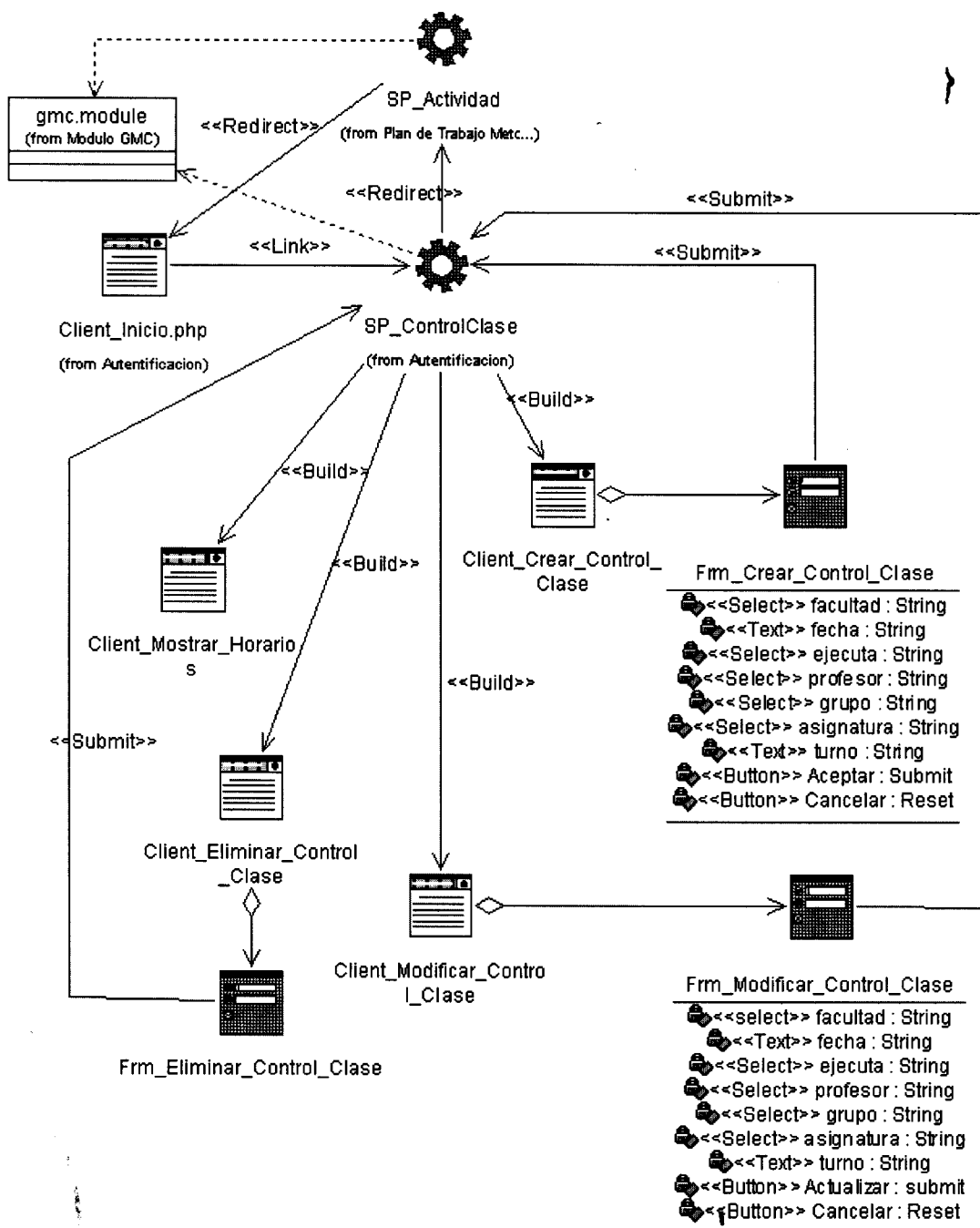


Figura 24. Diagrama de clases web del paquete <Control a Clases>

4.3.5 Paquete <Plan de Trabajo Metodológico>

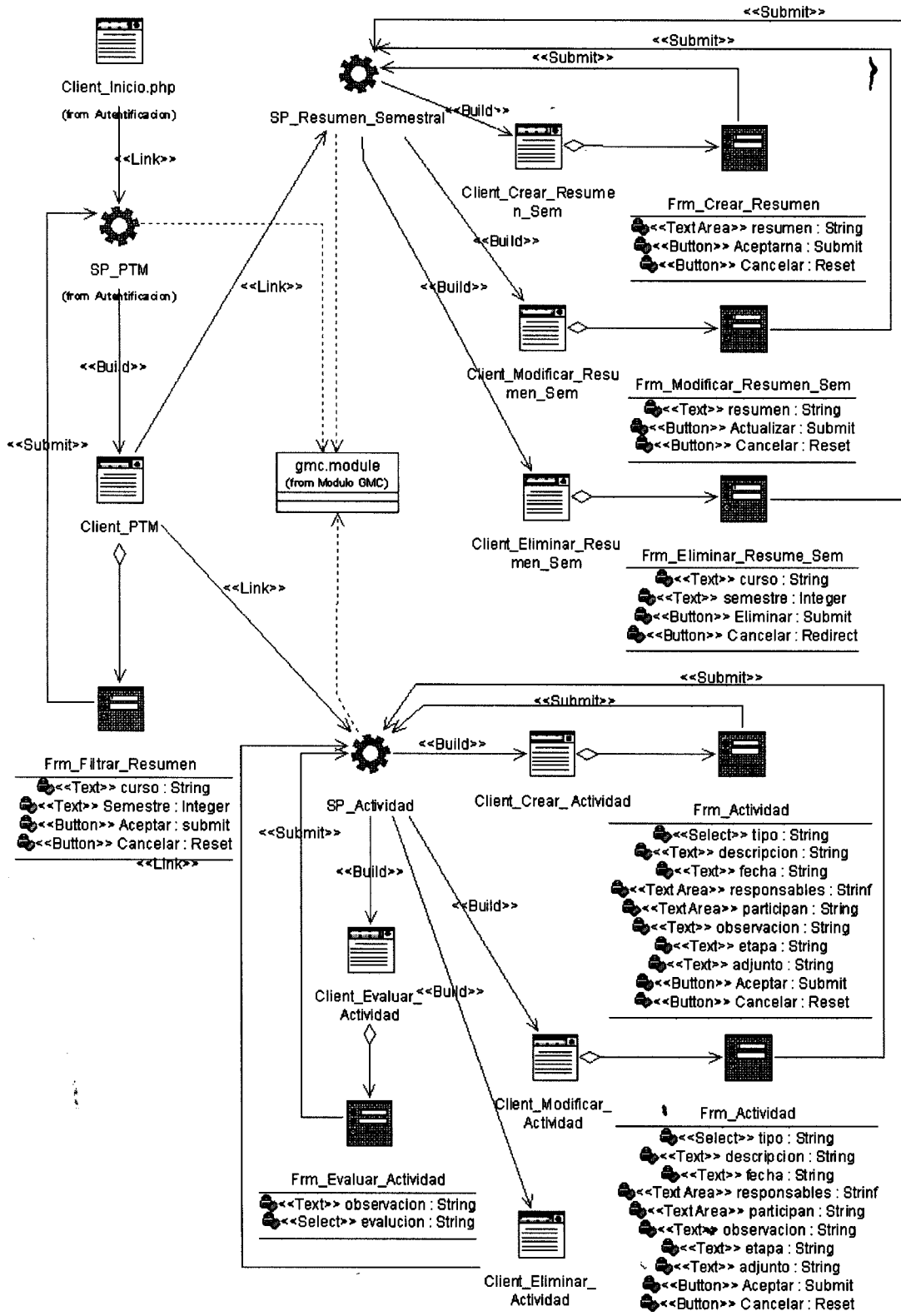


Figura 25. Diagrama de clases web del paquete <Plan de Trabajo Metodológico>

4.3.6 Paquete <Plan de Trabajo Individual>

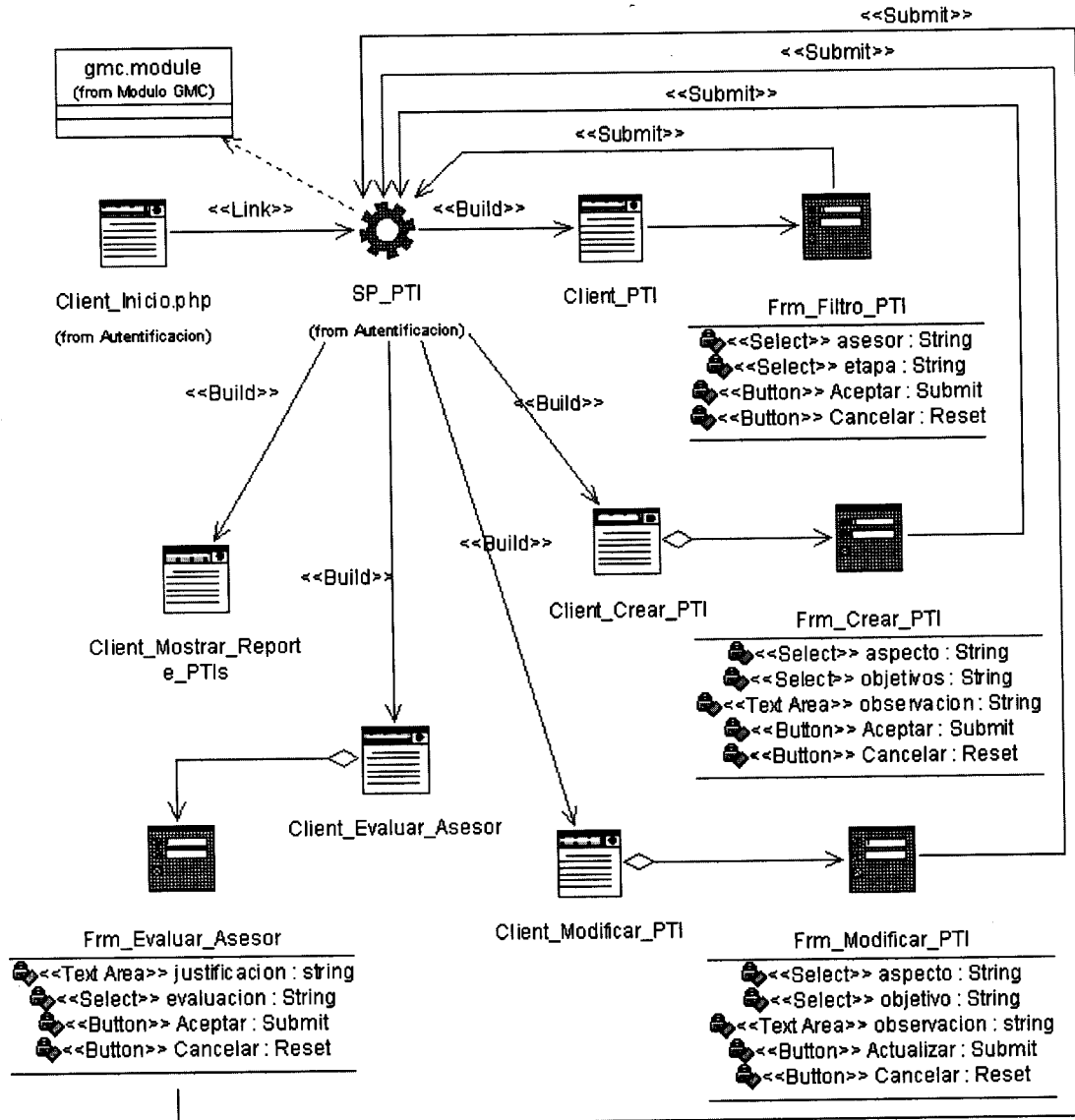


Figura 26. Diagrama de clases web del paquete < Plan de Trabajo Individual >.

4.3.7 Paquete <Plantilla >

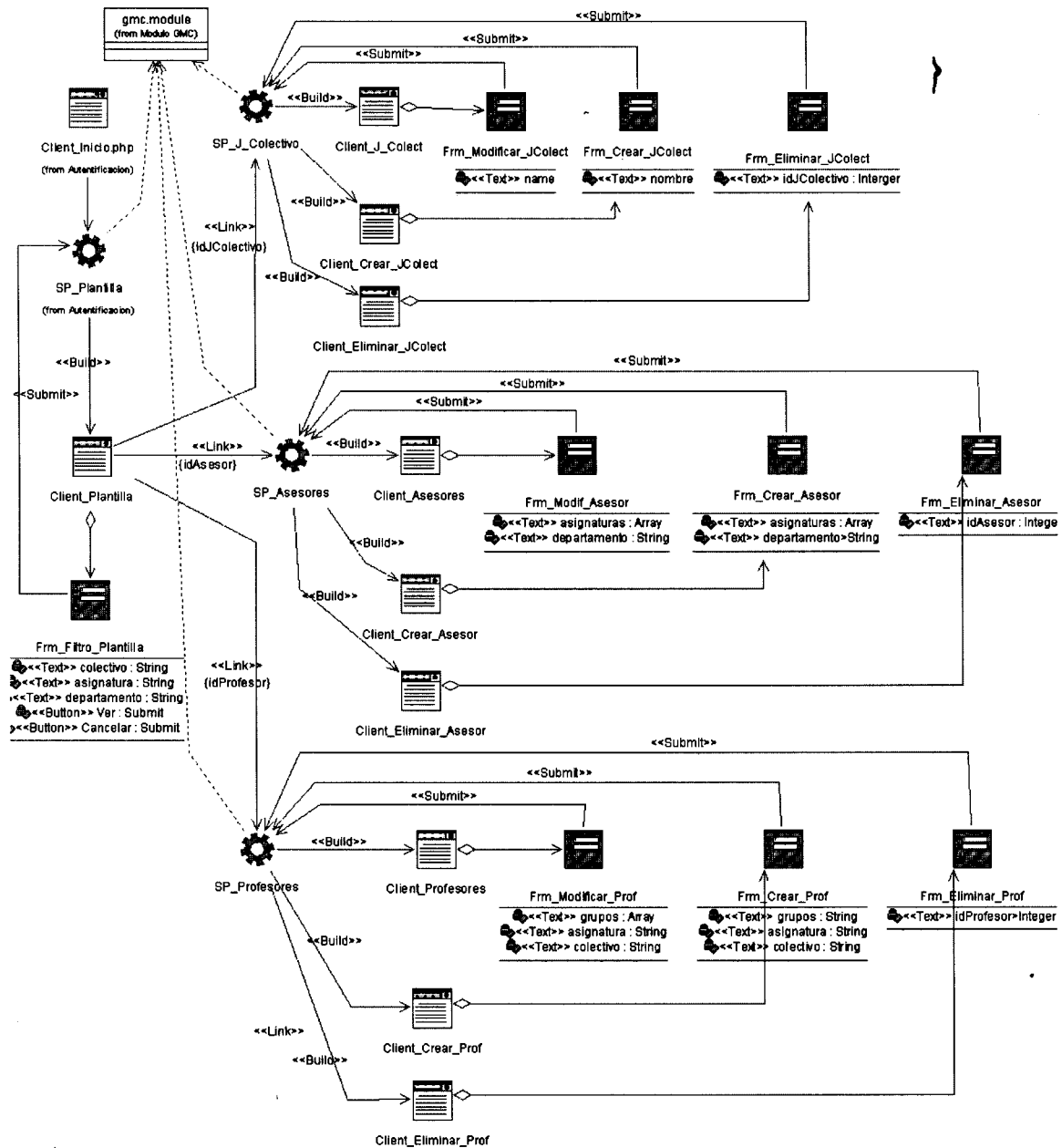


Figura 27. Diagrama de clases web del paquete <Plantilla>.

4.4 Principios de diseño

El diseño de la interfaz es uno de los puntos fundamentales a tratar a la hora de la presentación de la aplicación teniendo en cuenta que es lo que ve el usuario y por lo tanto debe ser lo más amigable y comprensible posible. Una aplicación con una interfaz bien diseñada debe tener, además de un buen diseño gráfico, una buena navegabilidad, usabilidad y distribución de los contenidos.

Para el diseño de la interfaz de usuario de este sistema se han seguido los siguientes principios:

- Permitir al usuario acceder solamente a las opciones a las que, dado su rol, puede ejecutar.
- Permitir su utilización por cualquier persona que tenga conocimientos básicos de informática.
- Los objetos que componen la interfaz den a entender al usuario de su existencia y su funcionalidad, donde el usuario puede claramente darse cuenta si es un objeto de interacción con la interfaz o es sencillamente información.
- Brindar a los usuarios una retroalimentación de las acciones que ejecutan, mostrando mensajes de confirmación, explicación, y navegación entre otros, claramente redactados de forma tal que el usuario tenga dominio de donde esta y de lo que está sucediendo.
- Evitar la sobrecarga de imágenes para favorecer la navegabilidad haciendo uso de imágenes pequeñas (thumbnails) vinculadas.

4.4.1 Interfaz de usuario

Con el objetivo de lograr un diseño consistente de la interfaz de la aplicación, se respetó en todas las páginas el esquema Cabecera-Navegador-Contenido. La cabecera contiene el nombre de la aplicación en la esquina superior izquierda y el logo de la organización que puede ser fácilmente configurable. En el navegador se incluyen uno o varios bloques, que se muestran en correspondencia con el rol del usuario, que soportan de forma jerárquica el sistema de menú. En el área del contenido se muestran los restantes contenidos (formularios de entrada, las salidas, los reportes, etc).

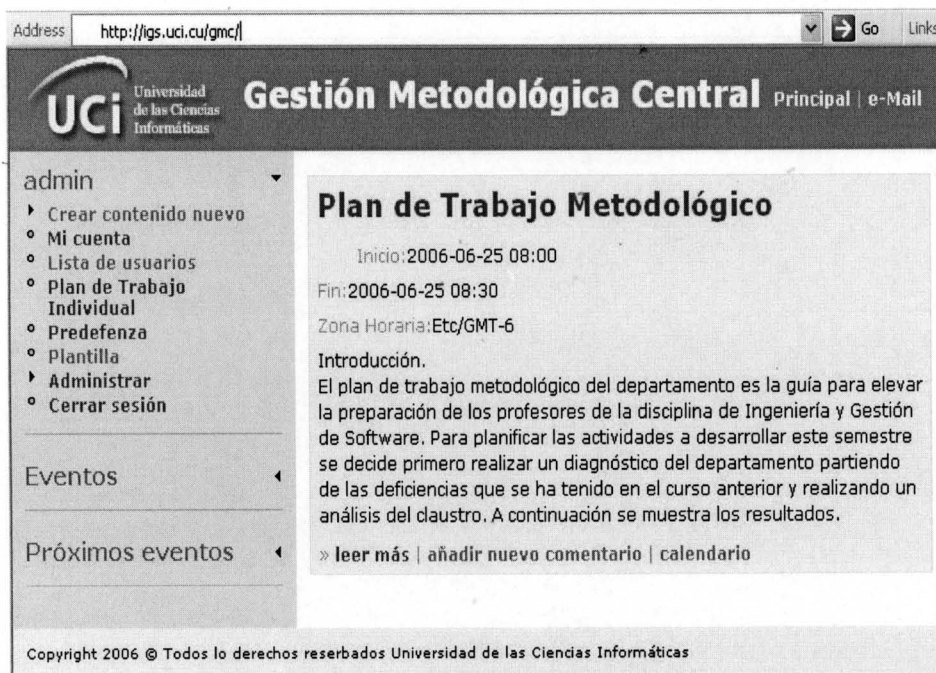


Figura 28. Esquema general del diseño.

Para el diseño se utilizan las tablas y plantillas, se utiliza también una hoja de estilos para guardar la configuración del diseño de todas las páginas. Esta hoja de estilos establece el tipo y tamaño de fuente de los distintos elementos de cada página. Se utiliza en general la fuente Arial, de tamaño entre 10 y 16 píxeles, según la importancia de la información mostrada. La hoja de estilos también establece el color de fondo, el formato de las tablas y bloques, entre otros.

4.5 Tratamiento de errores

El tratamiento de errores se realiza con el sistema de captura de errores de Drupal, una vez que ocurre una excepción el cliente es redireccionado a una página de error con el mensaje correspondiente.

Por otra parte cada formulario se encarga de la validación de sus datos para evitar errores de concepto. Y se utilizan mensajes de confirmación, para acciones que son irreversibles como es el caso de las eliminaciones. Por último, se utilizan errores en forma de mensajes de texto en la misma página donde se ejecutó la acción, de forma que el usuario pueda corregir más fácilmente y continuar.

CAPÍTULO 4 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

The screenshot shows the 'Gestión Metodológica Central' interface. On the left, there is a navigation menu with 'Inicio de sesión de usuario' expanded, showing fields for 'Nombre de usuario: *' (containing 'yunieve') and 'Contraseña: *', and an 'Iniciar sesión' button. Below this is a 'Solicitar nueva contraseña' link. The main content area shows a 'Principal' header, a message 'El campo Contraseña es necesario.' in a red box, and a 'Plan de Trabajo Metodológico' section. This section includes dates for 'Inicio' (2006-06-25 08:00) and 'Fin' (2006-06-25 08:30), a 'Zona Horaria: Etc/GMT-6', and an 'Introducción' paragraph. At the bottom of the page, there is a copyright notice: 'Copyright 2006 © Todos los derechos reservados Universidad de las Ciencias Informáticas'.

Figura 29. Validación de formularios.

The screenshot shows the 'Gestión Metodológica Central' interface with the 'admin' menu expanded. The main content area displays a confirmation dialog titled '¿Está seguro de que quiere eliminar la cuenta yamilisf?'. The dialog text states: 'Todos los envíos realizados por este usuario se atribuirán a la cuenta de usuario anónimo. Esta acción no se puede deshacer.' Below the text are two buttons: 'Eliminar' and 'Cancelar'. The navigation menu on the left includes 'Crear contenido nuevo', 'Mi cuenta', 'Lista de usuarios', 'Plan de Trabajo Individual', 'Predefenza', 'Plantilla', 'Administrar', and 'Cerrar sesión'. The footer contains the same copyright notice as Figure 29: 'Copyright 2006 © Todos los derechos reservados Universidad de las Ciencias Informáticas'.

Figura 30. Confirmación de eliminación.

4.6.2 Modelo físico de datos

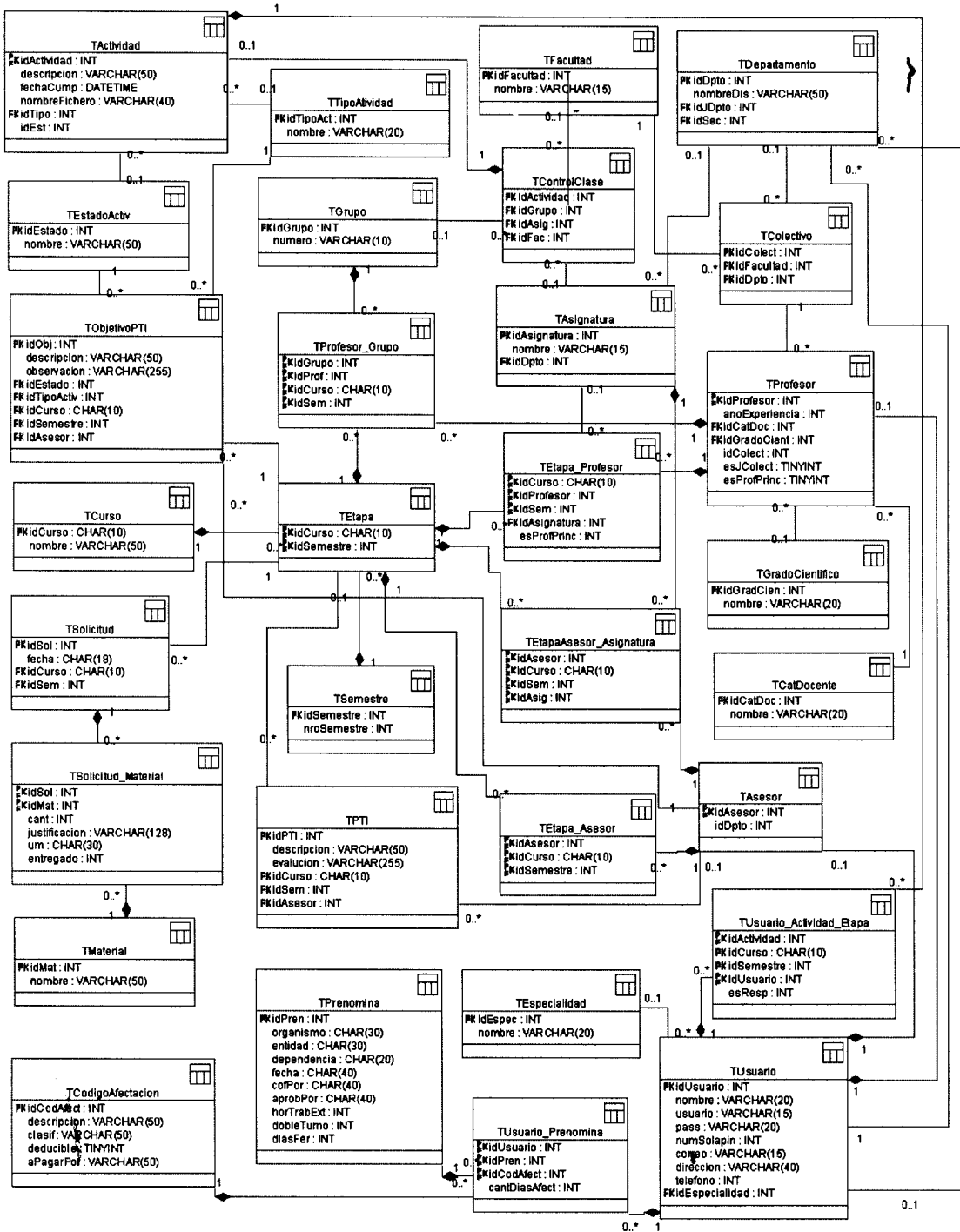


Figura 32. Diagrama de Base de Datos.

4.7 Diagrama 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, donde también radica el servidor de base de datos (MySQL).

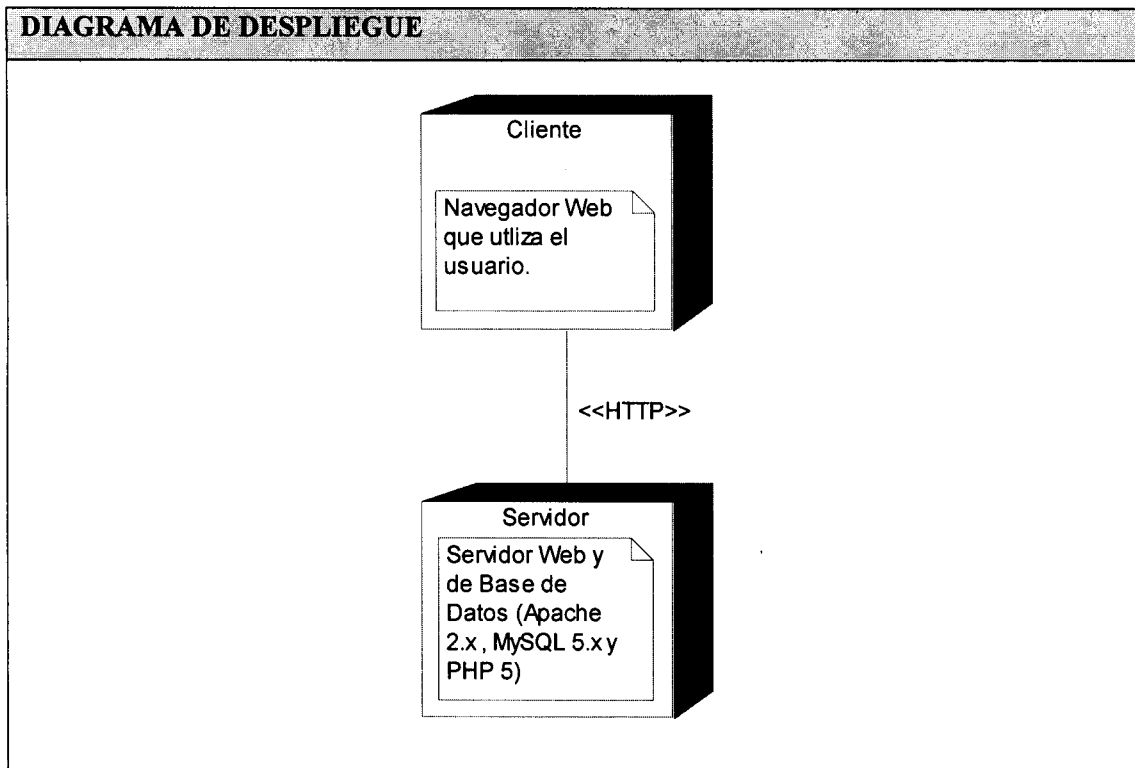


Figura 33. Diagrama de despliegue

4.8 Conclusiones

En el presente capítulo se obtuvo la solución al sistema propuesto, describiendo la arquitectura utilizada. Se mostraron los resultados de la etapa de diseño, para ello se utilizó un modelo de paquetes que brindó una mejor comprensión del sistema. Se modelaron los componentes como clases utilizando las extensiones del UML, y se

CAPÍTULO 4 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

presentaron a través de diagramas de clases Web. A partir del refinamiento del diagrama conceptual se obtuvo un modelo físico de datos en correspondencia con los requerimientos planteados en el capítulo anterior.

Se realizó el diseño del diagrama de despliegue donde se describen los nodos de procesamiento en tiempo real donde se ejecutará la aplicación y los vínculos entre ellos. Todos estos elementos obtenidos son claves para la correcta implementación del sistema propuesto.



5 ESTUDIO DE FACTIBILIDAD.

5.1 Introducción

En este capítulo se evalúa la factibilidad, beneficios y costo del sistema propuesto. Se obtendrán valores de importantes indicadores como son: esfuerzo y tiempo de desarrollo. Los cálculos parten de factores como entradas y salidas externas, ficheros internos, interfaces externas y el lenguaje utilizado.

5.2 Estimación de costo.

Existen varios modelos para estimar los costos de los proyectos de software. Entre ellos se encuentra COCOMO v2 (del inglés *Constructive Cost Model*), uno de los más aceptados internacionalmente en la actualidad. Este modelo expresa el esfuerzo de desarrollo en términos de Personas Mes.

En el caso del Sistema de Gestión Metodológica Central, al ser desarrollado en la Universidad de las Ciencias Informáticas, no se hace necesario calcular el costo monetario, pues se cuenta con el capital humano necesario (estudiantes y profesores), sin necesidad de contratar personal ajeno a la institución.

Se realizó un estudio del esfuerzo necesario para el desarrollo del sistema, arrojando un resultado de 20 Personas Mes, estimándose además, el tiempo de desarrollo del proyecto.

5.2.1 Planificación

Tabla 26. Entradas Externas.

Nombre de la entrada externa	Cantidad de Ficheros	Cantidad de Elementos de datos	Clasificación (Simple, Media y compleja)
Registrar resumen del PTM	1	2	Simple
Registrar actividades al PTM	1	6	Simple
Eliminar actividades del PTM	1	6	Simple
Registrar observación de la evaluación	1	1	Simple
Crear actividad	1	5	Simple
Eliminar actividad	1	5	Simple
Agregar aspectos del PTI	1	4	Simple
Eliminar aspectos del PTI	1	4	Simple
Agregar justificación de evaluación	1	3	Simple
Insertar datos de usuarios con permisos	1	12	Simple
Introducir profesores al colectivo	1	2	Simple
Eliminar profesores del colectivo	1	2	Simple
Agregar control a clases	1	5	Simple
Agregar materiales a Planilla de solicitud	1	5	Simple
Eliminar materiales	1	5	Simple
Confeccionar planilla de justificación de materiales	1	3	Simple
Confeccionar planilla de prenomina y aseo	2	15	Medio
Eliminar noticias publicadas	1	1	Simple

Tabla 27. Salidas Externas.

Nombre de la salida externa	Cantidad de Ficheros	Cantidad de Elementos de datos	Clasificación (Simple, Media y compleja)
Listado de características del claustro como diagnóstico inicial del PTM.	1	6	Simple
Informes semestrales de cursos anteriores	1	1	Simple
Mostrar PTM	1	1	Simple
Listado de las actividades a realizar en la semana	1	1	Simple
Listado de resumen del cumplimiento de la semana	1	1	Simple
Mostrar PTI de un asesor dado	1	1	Simple
Mostrar resumen del cumplimiento del PTI para un asesor determinado	2	5	Simple
Listado de profesores según la asignatura que imparten	1	1	Simple
Listado de profesores según la facultad a la que pertenecen	1	1	Simple
Listado de todos los profesores	1	2	Simple
Mostrar cantidad de controles a clases realizados	1	1	Simple
Resumen de controles a clases	1	4	Simple
Mostrar planilla de solicitud de materiales	1	4	Simple
Mostrar planilla de justificación de materiales	1	4	Simple

CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD

Mostrar reportes de materiales solicitados contra aceptados	1	1	Simple
---	---	---	--------

Tabla 28. Peticiones.

Nombre de la petición	Cantidad de Ficheros	Cantidad de Elementos de datos	Clasificación (Simple, Media y compleja)
Buscar características del claustro	1	3	Simple
Buscar informes semestrales	1	1	Simple
Buscar actividades del PTM	1	1	Simple
Buscar actividades por semana	1	1	Simple
Buscar resumen del cumplimiento de la semana	1	1	Simple
Buscar PTI de un asesor	2	2	Simple
Buscar evaluación de un asesor	1	2	Simple
Buscar profesores según asignaturas	2	2	Simple
Buscar profesores según facultad	2	2	Simple
Buscar profesores	1	1	Simple
Buscar horario docente de una facultad	1	2	Simple
Buscar cantidad de controles realizados	1	1	Simple
Buscar planilla de solicitud	1	1	Simple
Buscar planilla de justificación	1	1	Simple
Buscar planilla de prenomina	1	1	Simple

Tabla 29. Ficheros Internos.

Nombre del fichero interno	Cantidad de registros	Cantidad de Elementos de datos	Clasificación (Simple, Media y compleja)
Usuario	1	9	Simple
Profesor	1	9	Simple
Facultad	1	2	Simple
Colectivo	1	3	Simple
Asignatura	1	4	Simple
Grupo	1	2	Simple
CategoríaDoc	1	2	Simple
Curso	1	2	Simple
Especialidad	1	2	Simple
Actividad	1	8	Simple
TipoActiv	1	2	Simple
EstadoActiv	1	2	Simple
Usuario_Actividad	1	3	Simple
ControlClase	1	4	Simple
ObjetivoPTI	1	6	Simple
PTI	1	6	Simple
Departamento	1	6	Simple
Prof_Grupo	1	2	Simple
Asesor	1	2	Simple
Prenómina	1	8	Simple
Usuario_Pren	1	3	Simple
Prenomina_CodAfect	1	3	Simple
CodigoAfect	1	5	Simple
Solicitud	1	2	Simple
Material	1	2	Simple
Solicitud_Material	1	6	Simple

Tabla 30. Interfaces Externas.

Elementos	Simples		Medios		Complejos		Subtotal de puntos de función
	No	X Peso	No	X Peso	No	X Peso	
Ficheros lógicos internos	26	7	0	10	0	15	182
Entradas externas	17	3	1	4	0	6	55
Salidas externas	15	4	0	5	0	7	60
Peticiones	15	3	0	4	0	6	45
Total							342

5.2.2 Costos

Tabla 31. Cantidad de instrucciones fuentes.

Características	Valor
Puntos de función desajustados	342
Lenguaje	PHP (80 %) SQL (20%)
Instrucciones fuentes por puntos de función	15 39
Instrucciones fuentes por lenguaje (miles de instrucciones)	(4,104) (2,667)
Instrucciones fuentes (miles de instrucciones)	6,771

Tabla 32. Factores de escala.

Factores	Valor	Justificación
PREC	2.48	Se han desarrollado proyectos similares en el pasado.
FLEX	2.03	Debe haber buen cumplimiento de los requerimientos del sistema y de las especificaciones de interfaz externa.
TEAM	2,19	El equipo que va desarrollar el software es cooperativo.
RESL	2.83	La herramienta de desarrollo seleccionada posee gran

		tolerancia ante fallos y capacidad de resolución de riesgos que pudieran ocurrir. Se estima que la reducción del riesgo es entorno a un 75%.
PMAT	6.24	El proceso tiene un bajo nivel de madurez.

Tabla 33. Multiplicadores de escala.

Nombre	Valor	Justificación
RUSE	1.00	Existe concordancia entre la documentación y las necesidades del ciclo de vida.
PDIF	1.00	La plataforma es estable
PREX	1.18	El equipo de trabajo no tiene experiencia previa con la plataforma y las herramientas de desarrollo.
FCIL	0.87	Se hace un uso notable de herramientas case para documentar e implementar el sistema.
SCED	1.00	Planificación normal
RCPX	1.00	Alta confianza requerida, Base de datos mediana, documentación básica y complejidad normal
PERS	0.83	ACAP = ALTO PCAP = ALTO PCON = ALTO

La ecuación que plantea COCOMO para calcular el esfuerzo de desarrollo es la siguiente:

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

donde $E = B + 0.01 \times \sum_{j=1}^5 SF_j$

$$E = 0,91 + 0,01 * 15,77 = 1,0677$$

$$PM = 2,94 * 6,771^{1,0677} * 0,852 = 19,31 \approx 20 \text{ hombres-mes.}$$

La ecuación que plantea COCOMO para calcular el tiempo de desarrollo es la siguiente:

$$TDEV = C \times PM^F$$

donde $F = D + 0.2 \times (E - B)$

$$F = 0,28 + 0,2 \times (1,0677 - 0,91) = 0,31$$

$$TDEV = 3,67 \times 20^{0,31} = 9,29 \approx 10 \text{ meses}$$

La ecuación que plantea COCOMO para calcular la cantidad de hombres es la siguiente:

$$CH = PM / TDEV$$

$$CH = 20/10 = 2 \text{ personas}$$

Como el sistema es desarrollado por 1 hombre se obtiene el siguiente tiempo de desarrollo:

$$TDEV = 20 \text{ meses.}$$

$$CH = 1$$

5.3 Beneficios tangibles e intangibles

El Sistema de Gestión Metodológica Central de la UCI es un software que diseñado para el Departamento Docente Central de Ingeniería y Gestión de Software, puede adaptarse a otros entornos, por lo que puede utilizarse en otros departamentos ya sea de la UCI o cualquier otra universidad que necesite gestionar el proceso docente metodológico; haciéndolo en cierto modo comercializable, aunque no fue elaborado con este fin.

Su principal objetivo es ayudar en el proceso metodológico del Departamento Docente Central de Ingeniería y Gestión de Software, por tanto, los beneficios inmediatos son mayormente intangibles:

1. Ahorro de tiempo y recursos en la toma de decisiones del Jefe de Departamento.
2. Planeación más efectiva de las actividades del Plan de Trabajo del departamento.

3. Posibilidad de mantener informado al Jefe de Departamento y Asesores sobre las actividades planificadas.

5.4 Análisis de costos y beneficios

El desarrollo de este sistema no conlleva a grandes gastos de recursos, ni de tiempo. Las tecnologías y herramientas que se utilizaron para su desarrollo son totalmente libres, por lo que no hay que incurrir en gastos en el pago de licencias de uso. El sistema es completamente portable, por lo que un cambio de plataforma es viable y factible, y no hay que incurrir en muchos cambios; debido a la estructuración modular del diseño.

CONCLUSIONES

Luego del estudio realizado y del correspondiente diseño del sistema, se logró la definición de la arquitectura para la aplicación propuesta, con el fin de la gestión, planeación y control de las actividades metodológicas del Departamento Docente Central de Ingeniería y Gestión de Software de la UCI, tales como planes de trabajo, controles a clases, evaluación de las actividades, y entre otras, logrando establecer pautas con respecto a la seguridad de la información almacenada del departamento. Además se logra la implementación de un prototipo funcional, provisto de un ambiente cómodo, fácil de entender, y que cumple los estándares de diseño.

Para el desarrollo de la propuesta se realizó el análisis de la tecnología adecuada en favor de la política de Software Libre en que se encuentra la UCI, como en favor de los desarrolladores; concluyéndose en la utilización de un Framework Gestor de Contenido (CMF) como Drupal, específicamente PHP como lenguaje de programación, el gestor de bases de datos MySQL, y Apache como servidor Web junto a otras tecnologías como las hojas de estilo CSS.

El sistema se desarrolló siguiendo la metodología RUP, y se utilizaron representaciones UML para la modelación de todas las fases del proyecto. Se siguió una arquitectura modular, orientada a componentes con el uso de varios paradigmas de programación. Se modeló el negocio, se definieron los requerimientos del sistema, tanto funcionales como no funcionales, estructurándose además, el modelo de casos de uso del sistema.

Se realizó el análisis y diseño del sistema a través de diagramas de clases del análisis, diagramas de clases del diseño, de interacción, de clases persistentes, entre otros. Se elaboró el modelo de datos, el de despliegue y el de implementación, analizándose finalmente el costo y los beneficios que genera el sistema.

Por todo lo anterior se concluye que los objetivos propuestos para el presente proyecto en una primera iteración han sido cumplidos satisfactoriamente, incluyendo una serie de recomendaciones que deben tenerse en cuenta para el trabajo futuro.

RECOMENDACIONES

Al concluir este trabajo se recomienda lo siguiente para versiones futuras:

- Profundizar en el estudio de los procesos y tareas de los Departamentos Docentes Centrales de la UCI, creándose un grupo de desarrollo en esta dirección.
- Continuar el desarrollo de este sistema, adicionándole nuevas funcionalidades, adecuándolo a los cambios que puedan surgir en etapas posteriores, en el proceso de enseñanza de la UCI.
- Permitir la implantación de futuras versiones a nivel central, donde directivos a una mayor escala, en el proceso de formación de la UCI, puedan formar parte del grupo de usuarios directos del sistema, y tener un control más exacto de los resultados en el proceso docente metodológico.
- Implantar el sistema en la Intranet de la Universidad de las Ciencias Informáticas para prestar los servicios implementados dándole solución inmediata a los problemas detectados en la etapa de estudio preliminar.

REFERENCIAS BIBLIOGRÁFICAS

- [1]. Sæther Bakken, Stig. Schmid, Egon. Manual de Php. PHP Documentation Group. 2003. <http://www.php.net/docs.php>.
- [2]. Wikipedia. Artículo: "Framework" (05-2006).
<http://es.wikipedia.org/wiki/Framework>
- [3]. Sistemas de Gestión de Contenidos: "About". Drupal.org (2006).
<http://drupal.org/features>.
- [4]. Sistemas de Gestión de Contenidos: "Feature Overview". Drupal.org (2006).
<http://drupal.org/features>.
- [5]. Desarrollo de Web: Manual: "Tutorial de SQL". (02-2006).
<http://www.desarrolloweb.com/manuales/9/>.
- [6]. Freeman, P., "The Context of Design", en Software Design Techniques, IEEE Computer Society Press, 3ra. ed., 1980.
- [7]. RUMBAUGH J, JACOBSON I, BOOCH G. El Lenguaje Unificado de Modelado. Manual de Referencia. [PDF]. Addison Wesley, 1999.
- [8]. Bass, L., P. Clements y R. Kazman, "*Software Architecture in Practice*", Addison-Wesley, 1998.
- [9]. Model-View-Controller (MVC): Artículo: "*Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*", (2006)
http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/style.asp#10

BIBLIOGRAFÍA

- Framework: Artículo: "Framework" (05-2006).
<http://es.wikipedia.org/wiki/Framework>
- Freeman, P., "The Context of Design", en Software Design Techniques, IEEE Computer Society Press, 3ra. ed., 1980.
- Bass, L., P. Clements y R. Kazman, "Software Architecture in Practice", Addison-Wesley, 1998.
- Desarrollo de Web: Manual: "Tutorial de SQL". (02-2006).
<http://www.desarrolloweb.com/manuales/9/>.
- Sæther Bakken, Stig. Schmid, Egon. Manual de Php. PHP Documentation Group. 2003. <http://www.php.net/docs.php>.
- Sistemas de Gestión de Contenidos: "About". Drupal.org (2006).
<http://drupal.org/features>.
- Sistemas de Gestión de Contenidos: "Feature Overview". Drupal.org (2006).
<http://drupal.org/features>.
- Model-View-Controller (MVC): Artículo: "Estilos y Patrones en la Estrategia de Arquitectura de Microsoft", (2006).
http://www.microsoft.com/spanish/msdr/arquitectura/roadmap_arq/style.asp#10
- BOOCH Grady, RUMBAUGH James, JACOBSON Ivar. "El proceso unificado de desarrollo de software". 2000. Addison Wesley.
- BOOCH Grady, RUMBAUGH James, JACOBSON Ivar. "El Lenguaje Unificado de Modelado". Manual de referencia. 2000. Addison Wesley.
- Pressman, R. "Software Engineering. A Practitioner's Approach". Fourth Edition. McGraw – Hill. USA, 1999.
- Larman, C. "Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design". Prentice-Hall, Inc. 1998.
- Joseph, Schmuller. "Aprendiendo UML en 24 horas", Prentice-Hall, Inc. 2001.

GLOSARIO DE TÉRMINOS

CMF: Content Management Framework (Plataforma Gestión de Contenidos), son sistemas usados para la construcción de aplicaciones que gestionan contenido.

CSS: (Hoja de Estilo en Cascada) Dentro del diseño de páginas de Internet se presenta esta como la vanguardia en cuanto a definición de estilos dentro de las plantillas de diseño.

Código Abierto: Es una tendencia internacional del desarrollo de software que profesa la distribución del código junto a las aplicaciones, se rigen por licencias tales como GNU/GPL.

HTTP: HyperText Transfer Protocol (Protocolo de transferencia de hipertexto). Es el protocolo usado para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la World Wide Web.

IDE: Ambiente de desarrollo integrado. Es como se le llama al ambiente que proporciona al usuario una determinada herramienta de desarrollo.

Zend: Compañía líder de infraestructuras para web; está reconocida internacionalmente como la autoridad actualmente en PHP. Sus fundadores son los diseñadores del PHP v.4 en adelante, actualmente es una compañía líder dentro de la comunidad Open Source.

Hook: Del inglés gancho, palabra usada en la comunidad de Drupal para nombrar los métodos que deben implementar los módulos desarrollados para este CMS.