

**Instituto Superior Politécnico “José Antonio
Echeverría”
Facultad de Ingeniería Industrial
Centro de Estudios de Ingeniería y Sistemas (CEIS)**



Universidad de las Ciencias Informáticas

**SISTEMA DE GESTIÓN DE CURSOS OPTATIVOS Y DE SEGUNDO
PERFIL (SIGESCOSP)**

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

Autor: *Alejandro Antonio Sánchez Cué*

Tutor: *Lic. William Martínez Cortes*

Universidad de las Ciencias Informáticas

williamm@uci.cu.

Ciudad de La Habana, Cuba

Junio, 2006

DECLARACIÓN DE AUTORÍA

Declaro que yo, Alejandro Antonio Sánchez Cué, soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) y al Centro de Estudios de Ingeniería y Sistemas (CEIS) para que hagan el uso que estimen pertinente con este trabajo.

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

Firma del autor

Alejandro Antonio Sánchez Cué

Nombre completo del tutor

Lic. William Martínez Cortes

OPINIÓN DEL TUTOR

Título: Sistema de Gestión de Cursos Optativos y de Segundo Perfil.

Autor: Alejandro Antonio Sánchez Cué

Desde que al estudiante Alejandro Antonio Sánchez Cué se le asignó como Trabajo de Diploma el diseño de un sistema para la gestión de los cursos optativos de la Facultad 3 de la Universidad de las Ciencias Informáticas, este se enfrentó a dificultades como la ausencia de un cliente fijo, que supiera exactamente los requerimientos que debía cumplir el sistema. A pesar de esto, supo extraer requisitos de clientes sin conocimientos de informática, y diseñar por si solo un sistema que cumpliera con las necesidades de los clientes, aun cuando la facultad no tenía claro de cómo debía realizarse este.

Su diseño, a pesar de no ser perfecto, pues nadie hace nada perfecto la primera vez, contiene la esencia de lo que se quiere lograr, siendo fácilmente modificable y extensible. Este servirá de punto de partida para que un equipo de desarrolladores de la propia universidad pueda implementar el sistema de gestión de cursos optativos, que facilitará una tarea que, a medida que crece nuestra universidad, se vuelve cada vez más tediosa y burocrática. Vale destacar que aunque la idea original era automatizar la gestión de cursos optativos de la mencionada facultad, el estudiante siempre tuvo en mente la posibilidad de extensión de su sistema a toda la universidad, haciendo su diseño consecuente con estas ideas.

Los documentos entregados, dígame diagramas UML, diseño de la base de datos y el propio informe del trabajo de diploma, están muy bien estructurados, y servirán de material de consulta para el equipo de desarrollo que acometerá la tarea de implementar el sistema.

Como se evidencia de la lectura de los párrafos anteriores, el estudiante a demostrado una alta originalidad, creatividad, laboriosidad, y responsabilidad ante el trabajo, y sobre todo una muy alta independencia.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de 5 puntos.

Firma

Fecha

OPINIÓN DEL USUARIO

El Trabajo de Diploma titulado "Sistema de Gestión de Cursos Optativos y de Segundo Perfil", fue desarrollado en la Universidad de las Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

Totalmente _____

Parcialmente en un _____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____

Representante de la entidad Cargo

Firma Cuño

DEDICATORIA

*A mis padres con todo mi amor,
por haber dedicado toda sus vidas a mi formación,
y no haber escatimado esfuerzos para ver realizado este
que es nuestro gran sueño...*

Alejandro

AGRADECIMIENTOS

Quisiera agradecer a todas las personas que de una forma u otra han confiado en mí y me han ayudado en el desarrollo de este trabajo, de forma especial a:

- ◆ *A William Martínez Cortés, tutor de este proyecto, por su ayuda y orientación.*
- ◆ *A mis profesores a lo largo de los cinco años de mi carrera por inculcarme los conocimientos que me fueron muy útiles.*
- ◆ *A Julio y a mis compañeros de estudio por compartir todos los momentos.*
- ◆ *A Janet por su disposición a ayudarme en todo momento y contribuir en gran medida al éxito de este proyecto.*
- ◆ *A Yamy por su apoyo y preocupación durante todos estos meses.*
- ◆ *A Ridosbey porque sin su ayuda inicial no se hubiera podido lograr nada.*

RESUMEN

El presente trabajo consiste en el diseño de una aplicación Web para mejorar la gestión de los procesos existentes en la Universidad de las Ciencias Informáticas (UCI) relacionados con los cursos optativos y de segundo perfil, mediante la automatización de los mismos, con la creación del Sistema de Gestión de Cursos Optativos y de Segundo Perfil (**SiGesCOSP**), de manera que estudiantes y docentes obtengan un servicio de mayor calidad, con un alto nivel de fiabilidad y rapidez. Esta herramienta propuesta, contribuye al desarrollo de una eficiente organización de la universidad y sus servicios.

Con este sistema trataremos de solucionar gran parte de los problemas actuales relacionados con la gestión de los cursos optativos y de segundo perfil en la universidad, donde no existe ninguna aplicación con este fin. La aplicación muestra funcionalidades básicas de un sistema de gestión, entre las que se incluyen: autenticación y personalización de usuarios, subir programas analíticos de cursos, sugerencia de temas, publicación de temas para cursos, aprobación y oferta de cursos abiertos para matrícula, realización de matrículas en los cursos, entre otras.

Como parte del trabajo se entrevistaron a directivos y especialistas en el tema. Los resultados fundamentales son el diseño de la aplicación y el uso más eficiente de los recursos materiales y humanos.

Este trabajo se recoge toda la documentación relacionadas con el análisis y diseño del **SiGesCOSP**.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO # 1: FUNDAMENTACIÓN DEL TEMA	6
1.1 INTRODUCCIÓN	6
1.2 OBJETO DE ESTUDIO Y FLUJO ACTUAL DE LOS PROCESOS	6
1.3 PROCESOS OBJETO DE AUTOMATIZACIÓN	7
1.4 SISTEMAS AUTOMATIZADOS EXISTENTES VINCULADOS AL CAMPO DE ACCIÓN	8
1.5 FUNDAMENTACIÓN DE LOS OBJETIVOS	10
1.6 TENDENCIAS Y TECNOLOGÍAS ACTUALES	10
1.6.1 <i>Introducción</i>	10
1.6.2 <i>Software libre</i>	11
1.6.3 <i>Arquitectura cliente/servidor</i>	11
1.6.4 <i>Lenguajes de programación para la Web</i>	12
1.6.4.1 ASP.NET	12
1.6.4.2 PHP (Hypertext Preprocessor)	14
1.6.4.3 Selección del lenguaje a utilizar	16
1.6.5 <i>Sistemas gestores de Bases de Datos (SGBD)</i>	17
1.6.5.1 MySQL	17
1.6.5.2 SQL Server	18
1.6.5.3 Selección del SGBD a utilizar	18
1.6.6 <i>Metodologías de desarrollo de software</i>	19
1.6.6.1 RUP (Proceso de Desarrollo Racional)	19
1.6.6.2 XP (Programación Extrema)	20
1.6.6.3 FDD (Desarrollo guiado por la funcionalidad)	21
1.6.6.4 Selección de la metodología a utilizar	22
1.7 CONCLUSIONES	22
CAPÍTULO # 2: MODELO DE NEGOCIOS	23
2.1 INTRODUCCIÓN	23
2.2 MODELO DEL NEGOCIO ACTUAL	23
2.3 ACTORES Y TRABAJADORES DEL NEGOCIO	24
2.4 REGLAS DEL NEGOCIO	24
2.5 DIAGRAMA DE CASOS DE USO DEL NEGOCIO	26
2.6 CASOS DE USO DEL NEGOCIO	27
2.6.1 <i>CU1 Proponer curso</i>	27
2.6.2 <i>CU2 Sugerir tema</i>	28
2.6.3 <i>CU3 Solicitar matrícula</i>	29
2.7 MODELO DE OBJETO	30
2.8 CONCLUSIONES	30
CAPÍTULO # 3: REQUISITOS	31
3.1 INTRODUCCIÓN	31
3.2 ESPECIFICACIÓN DE REQUERIMIENTOS	31
3.2.1 <i>Requisitos funcionales</i>	31
3.2.2 <i>Requisitos no funcionales</i>	32
3.3 DESCRIPCIÓN DEL SISTEMA PROPUESTO	34
3.4 MODELO DE CASOS DE USO DEL SISTEMA	35
3.4.1 <i>Actores del sistema</i>	35
3.4.2 <i>Diagrama de Casos de Uso</i>	37
3.5 DESCRIPCIÓN DETALLADA DE LOS CASOS DE USO DEL SISTEMA	37
3.5.1 <i>CU1 Autenticar usuario</i>	38
3.5.2 <i>CU2 Solicitar matrículas</i>	39
3.5.3 <i>CU3 Gestionar cursos propuestos</i>	40

3.5.4 CU4 Gestionar cursos publicados.....	42
3.5.5 CU5 Proponer curso.....	44
3.5.6 CU6 Sugerir tema.....	45
3.5.7 CU7 Realizar matrícula.....	46
3.5.8 CU8 Gestionar temas.....	47
3.5.9 CU9 Listar cursos disponibles.....	49
3.5.10 CU10 Gestionar usuarios.....	50
3.6 CONCLUSIONES.....	52
CAPÍTULO # 4: DESCRIPCIÓN DE LA SOLUCIÓN.....	53
4.1 INTRODUCCIÓN.....	53
4.2 DIAGRAMAS DE INTERACCIÓN.....	53
4.2.1 CU1 Autenticar usuario.....	54
4.2.2 CU2 Solicitar matrícula.....	54
4.2.3 CU3 Gestionar cursos propuestos.....	55
4.2.4 CU4 Gestionar cursos publicados.....	55
4.2.5 CU5 Proponer curso.....	56
4.2.6 CU6 Sugerir tema.....	56
4.2.7 CU7 Realizar matrícula.....	57
4.2.8 CU8 Gestionar tema.....	58
4.2.9 CU9 Listar cursos disponibles.....	58
4.2.10 CU10 Gestionar usuarios.....	59
4.3 DIAGRAMA DE CLASES DE DISEÑO.....	59
4.4 PRINCIPIOS DE DISEÑO.....	59
4.5 TRATAMIENTO DE ERRORES.....	60
4.6 DISEÑO DE LA BASE DE DATOS.....	62
4.6.1 Modelo Lógico de Datos.....	62
4.6.2 Modelo Físico de Datos.....	63
4.7 MODELOS DE IMPLEMENTACIÓN.....	65
4.7.1 Diagrama de Despliegue.....	65
4.7.2 Diagrama de Componentes.....	66
4.8 CONCLUSIONES.....	71
CAPÍTULO # 5: ESTUDIO DE FACTIBILIDAD.....	72
5.1 INTRODUCCIÓN.....	72
5.2 PLANIFICACIÓN POR PUNTO DE FUNCIÓN.....	72
5.2.1 Planificación.....	73
5.2.2 Costos.....	76
5.3 BENEFICIOS TANGIBLES E INTANGIBLES.....	78
5.4 ANÁLISIS DE COSTO Y BENEFICIOS.....	79
5.5 CONCLUSIONES.....	80
CONCLUSIONES.....	81
RECOMENDACIONES.....	82
REFERENCIAS BIBLIOGRAFICAS.....	83
BIBLIOGRAFIA.....	84
GLOSARIO DE TÉRMINOS.....	87
ANEXOS.....	88

ÍNDICE DE FIGURAS

Fig. 1.1 Arquitectura del Framework de .NET.....	13
Fig. 1.2 Esquema de funcionamiento de PHP.....	15
Fig. 2.1 Diagrama de Casos de Uso del negocio.....	26
Fig. 2.2 Diagrama de Actividades del Caso de Uso Proponer curso.....	27
Fig. 2.3 Diagrama de Actividades del CU2 Sugerir tema.....	28
Fig. 2.4 Diagrama de Actividades del CU3 Solicitar matrícula.....	29
Fig. 2.5 Diagrama del Modelo de Objeto.....	30
Fig. 3.1 Diagrama de Casos de Uso del sistema.....	37
Fig. 4.1 Diagrama de Clases de Análisis CU1 Autenticar usuario.....	54
Fig. 4.2 Diagrama de Clases de Análisis CU2 Solicitar matrícula.....	54
Fig. 4.3 Diagrama de Clases de Análisis CU3 Gestionar cursos propuestos.....	55
Fig. 4.4 Diagrama de Clases de Análisis CU4 Gestionar cursos publicados.....	55
Fig. 4.5 Diagrama de Clases de Análisis CU5 Proponer curso.....	56
Fig. 4.6 Diagrama de Clases de Análisis CU6 Sugerir tema.....	56
Fig. 4.7 Diagrama de Clases de Análisis CU7 Realizar matrícula.....	57
Fig. 4.8 Diagrama de Clases de Análisis CU8 Gestionar temas.....	58
Fig. 4.9 Diagrama de Clases de Análisis CU9 Listar cursos disponibles.....	58
Fig. 4.10 Diagrama de Clases de Análisis CU10 Gestionar usuarios.....	59
Fig. 4.11 Diagrama de Clases Persistentes.....	62
Fig. 4.12 Diagrama del Modelo Físico de Datos [parte I].....	63
Fig. 4.13 Diagrama del Modelo Físico de Datos [parte II] (A continuación a la derecha de parte I).....	64
Fig. 4.14 Diagrama del Modelo Físico de Datos [parte III] (A continuación debajo de parte I y II).....	65
Fig. 4.15 Diagrama de Despliegue.....	66
Fig. 4.16 Diagrama de Componentes principal.....	67
Fig. 4.17 Diagrama de Componentes del modulo Autenticar.....	68
Fig. 4.18 Diagrama de Componentes del modulo Funcionalidades usuarios.....	69
Fig. 4.19 Diagrama de Componentes del modulo Administración [parte I].....	70
Fig. 4.20 Diagrama de Componentes del modulo Administración [parte II] (A continuación a la derecha de parte I).....	71

ÍNDICE DE TABLAS

Tabla 2.1 Actores del negocio.....	24
Tabla 2.2 Trabajadores del negocio.....	24
Tabla 2.3 CU1 Proponer curso.....	27
Tabla 2.4 CU2 Sugerir tema.....	28
Tabla 2.5 CU2 Sugerir tema.....	29
Tabla 3.1 Actores del sistema.....	36
Tabla 3.2 CU1 Autenticar usuario.....	38
Tabla 3.3 CU2 Solicitar matrícula.....	39
Tabla 3.4 CU3 Gestionar cursos propuestos.....	40
Tabla 3.5 CU4 Gestionar cursos publicados.....	42
Tabla 3.6 CU5 Proponer curso.....	44
Tabla 3.7 CU6 Sugerir tema.....	45
Tabla 3.8 CU7 Realizar matrícula.....	46
Tabla 3.9 CU8 Gestionar temas.....	47
Tabla 3.10 CU9 Listar cursos publicados.....	49
Tabla 3.11 CU10 Gestionar usuarios.....	50
Tabla 5.1 Entradas externas.....	73
Tabla 5.2 Salidas externas.....	74
Tabla 5.3 Peticiones.....	74
Tabla 5.4 Ficheros internos.....	75
Tabla 5.5 Interfaces externas.....	75
Tabla 5.6 Instrucciones fuentes estimadas.....	76
Tabla 5.7 Factores de escala.....	76
Tabla 5.8 Multiplicadores de esfuerzo.....	77

INTRODUCCIÓN

En la Universidad de las Ciencias Informáticas existe un sistema diseñado para que los estudiantes aumenten su caudal de conocimientos y desarrollen habilidades en temas variados (programación en diferentes lenguajes, redes, diseño, etc.) o herramientas complementarias, relacionados con el perfil de la Informática y las Telecomunicaciones y que no se encuentran concebidos dentro del programa obligatorio que deben cumplir y aprobar para graduarse; así como en temas de segundo perfil, que generalmente son temas de desarrollo específico que tiene cada facultad (turismo, bioinformática, etc.). Todo esto se lleva a cabo a través de cursos que se le imparte denominados cursos optativos y de segundo perfil. Las diferentes facultades de la universidad elaboran los programas analíticos para los cursos sobre los temas que son publicados por el departamento central de Práctica Profesional y sobre temas relacionados con su perfil. Estos programas son elaborados guiándose por un formato predeterminado confeccionado por dicho departamento de Práctica Profesional. Los cursos elaborados por las facultades puede ser propuestas echas por los profesores o pueden ser directamente elaborados en la facultad (son generalmente de 24 a 40 horas lectivas). Estas propuestas echas por las facultades son enviadas al departamento central de Práctica Profesional para su análisis y su posible aprobación para ser impartido.

Como requisito necesario para poder obtener su diploma de graduado de ingeniero en Informática, a todos los estudiantes de la UCI se les exige en su currículo tener cursado cierta cantidad de cursos entre cursos optativos y de segundo perfil a lo largo de sus 5 años en la universidad. Cada facultad establece la distribución de los cursos que deben cursar sus estudiantes entre ambas categorías.

Actualmente todos estos procesos de gestión de los cursos se realizan de forma manual. Cuando el departamento de Práctica Profesional aprueba las propuestas de los programas analíticos para cursos echas por las facultades, se determina cuándo va a ser la convocatoria y se publican todos esto resultados

en los murales de las facultades, con toda la información referente a cada curso; y se le envía por correo a los estudiantes la lista de cursos que están abiertos para matrícula con sus correspondientes datos. Estos entonces tienen la posibilidad de matricularse en los cursos en que haya capacidad y en los que cumplan con las condiciones expuestas anteriormente sobre los cursos obligatorios y opcionales que deben cumplir.

Por todo esto se nos presenta el siguiente **problema a resolver**: ¿Cómo diseñar una aplicación Web donde se pueda automatizar los procesos relacionados con la gestión de la información relacionada con los cursos optativos y de segundo perfil en la Universidad de las Ciencias Informáticas?

Con este trabajo esperamos recopilar y analizar información relacionada con la gestión de los cursos optativos y de segundo perfil en la UCI, investigar sobre las deficiencias del proceso y llegar a conclusiones sobre las mejores opciones con las que contamos para así culminar con el diseño de la aplicación que servirá para automatizar los procesos que en la actualidad se realizan de forma manual.

Como **objeto de estudio** tenemos los procesos relacionados con la gestión de los cursos optativos y de segundo perfil como son: realizar propuestas por las facultades de programas analíticos para cursos al departamento de Práctica Profesional, análisis de los programas analíticos propuestos en el departamento de Práctica Profesional y publicación de los cursos aprobados para impartirse, posibilidad de matrícula y datos sobre los cursos para los estudiantes y profesores.

En cuanto al **campo de acción** esta constituido por la automatización de todos los procesos relacionados a la gestión de los cursos optativos y de segundo perfil, facilitando así la realización de procesos que hoy en día se realizan de forma manual cuyo funcionamiento no es el más óptimo. Esto se pretende lograr mediante el diseño de una aplicación Web lo más flexible, rápida y eficaz posible.

Como principales **objetivos** tenemos los siguientes:

Objetivo general

Desarrollar un análisis de los temas relacionados a la gestión de los cursos optativos y de segundo Perfil en la UCI, llegando a diseñar una aplicación Web eficiente y adaptable a las necesidades existentes, que permita a todos los usuarios y administradores relacionados conocer los temas sobre los que se necesitan programas analíticos, permitir subir programas analíticos para cursos al departamento central de Práctica Profesional, publicar los cursos que se aprueben con todos los datos de los mismos, matricular a los alumnos interesados.

Objetivos específicos:

- Diseñar un módulo que permita publicar los temas que constituyen necesidad de cursos optativos para el departamento central de Práctica Profesional. Aquí se desarrollara una interfaz donde las facultades puedan subir sus sugerencias de temas de cursos y de programas analíticos para cursos sobre temas publicados para ser analizados.
- Diseñar un módulo que permita sólo a las personas autorizadas del departamento de Práctica Profesional acceder a los cursos y temas que han sido propuestos por las facultades para su análisis para una posible publicación.
- Diseñar un módulo que permita publicar los cursos después de haber pasado por el proceso de revisión y haber sido aprobados por el departamento de Práctica Profesional, permitiendo la publicación en dos modos: abierto para matrícula, que es cuando ya se encuentra definido todos los pormenores (fecha de la convocatoria, disponibilidad de aula, profesor que lo va a impartir, período lectivo, etc.); y en espera de planificación, que es cuando no se han determinado algunos de los detalles antes expuestos.
- Diseñar un módulo que permita a los estudiantes ver los cursos que están en oferta y hacer la solicitud de matrícula, y permitiendo la realización de la matrícula, brindando la facilidad de notificar a los usuarios si finalmente han sido aprobados como matrícula del curso en cuestión.

- Diseñar un módulo que permita brindar datos estadísticos de los cursos optativos y de segundo perfil, manteniendo los distintos niveles de acceso para los diferentes tipos de usuarios.

Con vistas a dar cumplimiento a los objetivos trazados se propone la realización de las siguientes tareas:

- Analizar el funcionamiento actual de los cursos optativos y de segundo perfil en la universidad.
- Elaborar la fundamentación teórica del objeto de estudio.
- Caracterizar sitios Web internacionales relacionados con cursos optativos
- Seleccionar la Metodología de Análisis y Diseño de Sistemas Informáticos, que garantice la calidad del sistema.
- Seleccionar la plataforma y herramientas en la que se desarrollará la aplicación, fundamentando dicha elección.
- Diseñar una Base de datos que soporte las funcionalidades necesarias al sistema.

El desarrollo de esta aplicación y su posterior explotación, minimizará el tiempo de trabajo necesario para llevar a cabo cualquier operación, así como las pérdidas de datos. Se logrará una mejor organización en los procesos que ella automatiza. Además, la interfaz amigable hará posible que cualquier persona sin necesidad de una alta calificación pueda interactuar con el sistema, con riesgos mínimos en cuanto a introducción de errores.

El presente documento está compuesto por 5 capítulos:

En el Capítulo # 1 se describe la situación actual. Se abordan los principales problemas que motivan el desarrollo de este trabajo, se describe los procesos que se proponen automatizar y se fundamentan los objetivos se plantean.

También se hace referencia a las tendencias y tecnologías existentes en la actualidad que se deben considerar para hacer la selección de aquellas que se van a utilizar en el proyecto. Finalmente se plantea dicha selección a modo de propuesta.

En el Capítulo # 2 se describen los procesos actuales a través de un modelo de negocio, el cual sirve de base para determinar qué es lo que se va a desarrollar.

En el Capítulo # 3 se determinan las funcionalidades del sistema propuesto y se describen las mismas, formulándose el modelo del sistema, el cual respalda todos los requerimientos funcionales anteriormente delimitados.

El Capítulo # 4 aborda aspectos relacionados con la construcción de la solución propuesta, se modelan los diagramas de clases de análisis y diseño, se plantea el modelo de datos, se especifican principios de diseño, así como el tratamiento de los errores.

El Capítulo # 5 y último, contiene el estudio de factibilidad realizado para este sistema, se enumeran los beneficios tangibles e intangibles y se analizan los costos que representa la elaboración de la propuesta de solución, así como los beneficios que aporta dicha propuesta.

Capítulo # 1: FUNDAMENTACIÓN DEL TEMA

1.1 Introducción.

En el presente capítulo se describen los procesos actuales de la gestión de los cursos optativos y de segundo perfil en la UCI y se identifican los principales problemas que nos conllevan a la investigación. Se analiza la forma que en que se realizan los procesos, la inconveniencia de la misma y se plantea una variante de solución y se fundamentan los objetivos propuestos. Además se analizan de forma detallada las tecnologías con las que contamos para elaborar el producto de software y se escogen las más convenientes de acuerdo a nuestras necesidades y posibilidades.

1.2 Objeto de estudio y flujo actual de los procesos.

Como objeto de estudio tenemos todos los procesos relacionados con la gestión de los cursos optativos y de segundo perfil como son: publicación de temas para cursos, propuesta de programas analíticos para cursos por los jefes de departamento de las distintas facultades sobre los temas publicados, aprobación y publicación de los programas analíticos para los cursos por las personas acreditadas del departamento central de Práctica Profesional, posibilidad de matrícula para los estudiantes en los cursos publicados, acceso a los datos sobre los cursos abiertos, sugerencia de nuevos temas para cursos, actualización de nuevas versiones de los cursos publicados, etc.

El flujo actual de los procesos involucrados en el campo de acción de este proyecto se inicia con la publicación de temas para cursos optativos y de segundo perfil por el departamento de Práctica Profesional de acuerdo a las necesidades que tenga cada facultad y que posean ellos mismos. Las facultades entonces, en la persona de los jefes de departamento autorizados, proponen los programas analíticos para los cursos de los temas seleccionados, además también le sugieren temas para nuevos cursos al departamento de Práctica Profesional. Este se da a la tarea de revisar las propuestas que le han sido

enviadas por las facultades y aprueba los que crea conveniente según su valoración. Cuando ya están los cursos que han sido aprobados, estos se publican con todos los datos necesarios (fecha de la convocatoria, cantidad de horas lectivas, horario, profesor, lugar, etc.) para que los estudiantes interesados hagan su solicitud de matrícula en los cursos que desean cursar y aún tengan capacidad. De aquí ya salen conformadas las matrículas y queda listo el curso para comenzar a impartirse.

1.3 Procesos objeto de automatización.

A continuación se describen los procesos que serán objeto de automatización por nuestro sistema:

Publicar temas para cursos optativo y de segundo perfil:

El departamento de Práctica Profesional en sus análisis realizados puede determinar un grupo de temas para los cursos optativos y de segundo perfil sobre los que ellos necesitan recibir propuestas de programas analíticos. Estos temas son necesidades que le han enviado las distintas facultades o que ellos mismos han determinado. Después se publican los temas que finalmente queden seleccionados para que sean accesibles a los usuarios y estos puedan realizar sus propuestas.

Proponer programa analítico para un curso:

Dados los temas que se encuentran publicados, las facultades pueden proponer al departamento central de Práctica Profesional un programa analítico para un tema de curso en específico, enviando la propuesta en el formato predeterminado con anterioridad por el departamento, en los temas donde haya capacidad para proponer un nuevo programa analítico, ya que no puede exceder el límite de propuestas antes fijado por el departamento central.

Publicar cursos abiertos para matrículas:

Los programas para los cursos que son propuestos por las facultades llegan a un banco de propuestas en el departamento central de Práctica Profesional, los encargados de revisar estos materiales realizan los análisis correspondientes de cada documento y aprueban los que cumplan con los requerimientos que ellos

exigen y que se estime conveniente que están en condiciones de ser aplicados a los estudiantes. Estos son publicados en dos modos: pendiente, que son los que no se han determinado todos los detalles; y abiertos para la matrícula, que es cuando ya se pueden empezar a recibir las solicitudes de matrículas por parte de los estudiantes.

Solicitud de matrículas en un curso:

Cuando los cursos optativos y de segundo perfil son publicados y abiertos para matrícula, los estudiantes pueden hacer su solicitud en los cursos que deseen y que a su vez estén en correspondencia con las restricciones entre obligatorios y opcionales que ellos deben cumplir de acuerdo al perfil de cada facultad.

Análisis de las solicitudes de matrículas:

La facultad correspondiente realiza el análisis de las solicitudes de matrícula de los estudiantes en los cursos y realiza la aprobación de las mismas que ellos consideren y que sean viables, enviándoles una notificación a los matriculados y a los no matriculados que fue denegada su solicitud.

1.4 Sistemas automatizados existentes vinculados al campo de acción.

La razón de existencia de un software de gestión esta basada en la digitalización de todo tipo de procesos de manipulación de información que se realizan de forma manual o no automatizada. Existe gran variedad de estos como pueden ser de gestión bibliotecaria, de gestión docente, gestión de recursos humanos, gestión de cursos, y otros. El sistema que nos hemos propuesto diseñar esta relacionado con la gestión de los cursos optativos y de segundo perfil en la Universidad de las Ciencias Informáticas.

Después de una búsqueda exhaustiva en Internet y de consultar amplia bibliografía no se pudo hallar software alguno en Cuba relacionado, pero se encontraron algunas referencias de sitios Web internacionales que tienen que ver con la problemática que nos atañe. A continuación algunos ejemplos:

http://www.ecologia.edu.mx/posgrado/cursos_optativos.htm

Este sitio radica en México. La colección de cursos optativos para cada estudiante se decide con la supervisión de cada Comité Tutorial, considerando los más apropiados al tipo de investigación de tesis (maestría y doctorado) y para lograr una educación integral y adecuada en al menos una de las cuatro fortalezas reconocidas. En este sitio se cuenta con las siguientes funcionalidades: muestra el nombre del curso, los profesores participantes, una descripción breve del curso, la última vez que se ofreció el curso, y la frecuencia.

http://licbq.fcien.edu.uy/menu/submenu_planes/optativas.html

En Montevideo, Uruguay, se tiene un sitio Web sobre cursos optativos para planes de estudio, donde muestran al usuario tres tipos de cursos: optativas semestre impar, optativas semestre par y futuras optativas.

<http://cursos.puc.cl/catalogo/sitio/aap.html>

Es un sitio Web con cursos optativos de Licenciatura en los Agrorrecursos, donde muestran el nombre del curso, la sigla, los pre-requisitos, descripción, programa del curso, y la Web del curso.

http://mamcyp.unam.edu.ar/index.php?option=com_content&task=view&id=12&Itemid=15

Es una Web de cursos optativos sobre orientación de celulosa y papel. En la sección de cursos optativos brinda el nombre del curso, la cantidad de horas que tiene, y descripción acerca de él.

http://www.quimica.izt.uam.mx/posgrado-quimica/cursos_optativos.htm

En el sitio Web de cursos optativos sobre química. En la sección de cursos optativos, le muestran al usuario la clave, el nombre, el trimestre, y unas especificaciones propias del curso.

Después de analizar todos estos ejemplos vemos que casi la totalidad consta de características similares, poseen escasas funcionalidades en la gestión de los cursos.

Nuestro sistema es único en el sentido que surge como una necesidad de la universidad y va destinado a darle solución a problemáticas muy específicas que existen en nuestro entorno y que hasta ahora no se le ha dado solución

alguna. Además muestra grandes avances comparado con los sitios analizados como son la posibilidad de enviar nuevos curso y temas, realizar matriculas online, etc.

1.5 Fundamentación de los objetivos.

Como principal objetivo de nuestro proyecto esta desarrollar un análisis de todos los procesos relacionados con la gestión de la información relacionada con los cursos optativos y de segundo perfil en la Universidad de las Ciencias Informáticas, para culminar con el diseño de una aplicación Web que permita a los usuarios y administradores relacionados de alguna forma con los cursos optativos y de segundo perfil acceder a la información de una forma rápida y eficiente.

Todo esto fue necesario dado el auge que esta teniendo en la universidad el movimiento de los cursos optativos y de segundo perfil, y la carencia de un sistema automatizado que gestionara toda la información y así se agilizaran todos los procesos, por lo que se nos dio la tarea de diseñar el software que le diera respuesta a todos los objetivos y necesidades planteadas.

1.6 Tendencias y tecnologías actuales.

1.6.1 Introducción.

Nos proponemos hacer un análisis de las tecnologías y las tendencias que existen actualmente en el mundo, que nos pudieran ser de utilidad en el diseño, implementación y puesta a punto de la propuesta de solución brindada. Analizaremos los lenguajes de programación, los gestores de bases de datos, así como las metodologías de desarrollo de software más utilizadas a escala nacional e internacional. Finalmente se seleccionan las opciones más convenientes, teniendo en cuenta que estas deben permitir cumplir a cabalidad con los objetivos propuestos en el proyecto, así como ajustarse completamente a los intereses, posibilidades y políticas de la Universidad de las Ciencias Informáticas.

1.6.2 Software libre.

El concepto de *Software Libre* se refiere al derecho irrestricto de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Comprende las siguientes libertades:

- La libertad de ejecutar el programa con cualquier propósito.
- La libertad de estudiar cómo funciona el programa y adaptarlo a las necesidades propias.
- La libertad de redistribuir copias del programa y de ese modo ayudar a otros.
- La libertad de mejorar el programa y liberar esas mejoras al público y de ese modo beneficiar a toda la comunidad.

Para ejercer estas libertades, el acceso al código fuente es una precondition. Los sistemas operativos y programas de aplicación de software libre son múltiples y variados y abarcan las más corrientes necesidades de los usuarios. Están siendo permanentemente perfeccionados a través de la interacción de quienes los usan, quienes pueden modificarlos y adaptarlos fácilmente a sus necesidades [1].

Recientemente se ha venido observando una tendencia en la Universidad de las Ciencias Informáticas, y en nuestro país en general, hacia la utilización en grado creciente de este tipo de software. El presente trabajo parte de esa premisa y se propone el diseño de un sistema que satisfaga las necesidades que lo originaron, haciendo uso de herramientas y tecnologías libres.

1.6.3 Arquitectura cliente/servidor.

La arquitectura cliente/servidor es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realice se efectúe con la mayor eficiencia, y permita simplificarlas. En esta arquitectura la capacidad de procesamiento está repartida entre el servidor y los clientes. De acuerdo con la distribución de la lógica de la aplicación hay dos posibilidades:

1. Cliente liviano: si el cliente solo se hace cargo de la presentación.

2. Cliente pesado: si el cliente asume también la lógica del negocio.

Esta arquitectura posee dos ventajas fundamentales: el servidor no necesita tanta potencia de procesamiento, porque parte del proceso se reparte con los clientes y reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre para otra conexión [2].

1.6.4 Lenguajes de programación para la Web.

Existen muchos lenguajes que se emplean para el desarrollo de aplicaciones Web pero los más utilizados en la actualidad por sus grandes ventajas y potencialidades son el **ASP.NET** de la Microsoft y el **PHP** (*Hypertext Preprocessor*). A continuación se exponen algunos elementos sobre estos dos lenguajes y al final llegamos a conclusiones sobre cual es más factible utilizar para el desarrollo de nuestra aplicación.

1.6.4.1 ASP.NET.

.NET es un marco de trabajo de programación generado en Common Language Runtime que puede utilizarse en un servidor para generar eficaces aplicaciones Web. **.NET** ofrece varias ventajas importantes acerca de los modelos de programación Web anteriores [3]:

- Mejor rendimiento.
- Compatibilidad con herramientas de primer nivel.
- Eficacia y flexibilidad.
- Simplicidad.
- Facilidad de uso.
- Escalabilidad y disponibilidad.
- Posibilidad de personalización y extensibilidad.
- Seguridad.

El Framework de **.NET** es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de

aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables. Los principales componentes de este entorno son:

- Lenguajes de compilación.
- Biblioteca de clases de **.NET**.
- CLR (Common Language Runtime)..

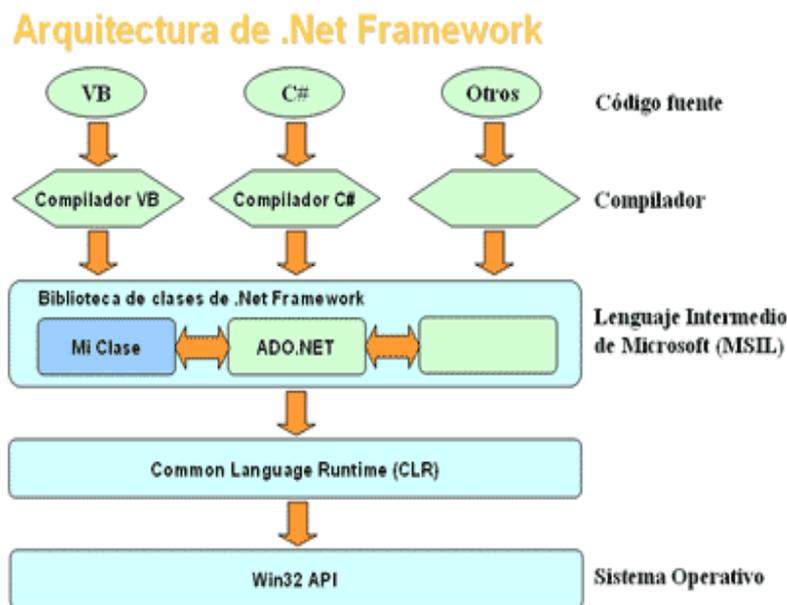


Fig. 1.1 Arquitectura del Framework de .NET.

.NET Framework soporta múltiples lenguajes de programación y aunque cada lenguaje tiene sus características propias, es posible desarrollar cualquier tipo de aplicación con cualquiera de estos lenguajes. Existen más de 30 lenguajes adaptados a **.NET**, desde los más conocidos como C# (C Sharp), Visual Basic o C++ hasta otros lenguajes menos conocidos como Perl o Cobol.

Las páginas de formularios Web de **ASP.NET** consisten en archivos de texto con una extensión de nombre de archivo .aspx. Pueden implementarse por todo un árbol de directorio raíz virtual IIS. Cuando un explorador cliente solicita recursos .aspx, el motor en tiempo de ejecución de **ASP.NET** analiza y compila el archivo de destino en una clase de **.NET** Framework. Esta clase puede utilizarse, a continuación, para procesar de forma dinámica las solicitudes entrantes. (Debe observarse que el archivo .aspx sólo se compila la primera que se tiene acceso al mismo; la instancia de tipo compilada se vuelve a utilizar en múltiples

solicitudes). Una página de **ASP.NET** puede crearse tomando simplemente un archivo HTML existente y cambiando la extensión del nombre de archivo a .aspx (no se necesita ninguna modificación del código) [4].

1.6.4.2 PHP (Hypertext Preprocessor).

PHP es el acrónimo de Hypertext Preprocessor (Preprocesador de Hipertexto). Se trata de un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Está muy orientado al desarrollo de aplicaciones Web y permite insertar contenidos dinámicos en las páginas.

Características:

- **Multiplataforma:** **PHP** funciona tanto en sistemas Unix o Linux con servidor Web Apache como en sistemas Windows con Microsoft Internet Information Server, de forma que el código generado por cualquiera de estas plataformas no debe ser modificado al pasar a la otra. Sin embargo, las mejores prestaciones del lenguaje se obtienen trabajando en un entorno Unix o Linux con servidor Web Apache. La principal ventaja que aporta el uso conjunto de **PHP** y Apache es que, de esta manera, se puede compilar el intérprete **PHP** como un módulo de Apache, consiguiendo así que la velocidad de ejecución de una página **PHP** sea elevada y que el consumo de recursos sea bajo ya que el intérprete **PHP** se carga una sola vez en memoria.
- **Ejecución en Servidor:** Un lenguaje del lado del servidor es aquel que se *ejecuta en el servidor Web* justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página **PHP**. Dado que la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

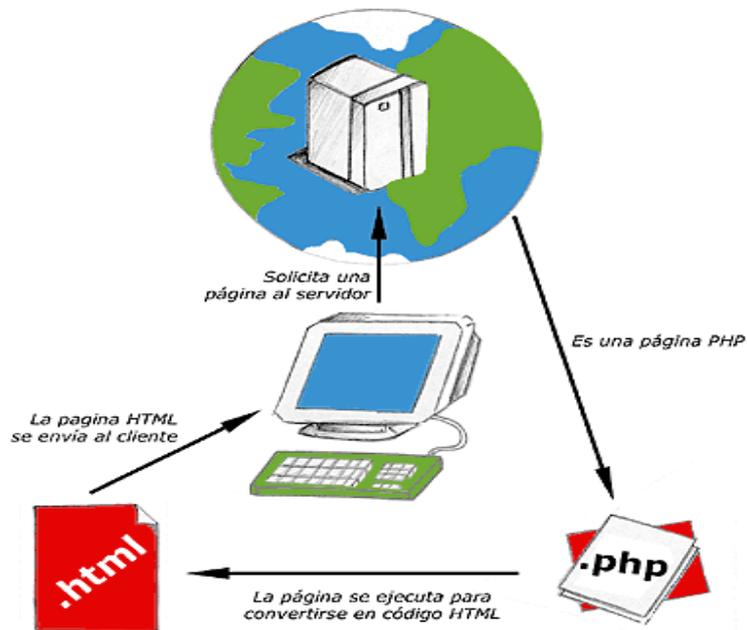


Fig. 1.2 Esquema de funcionamiento de PHP.

Cuando un navegador solicita al servidor Web una página **PHP**, antes de enviar dicha página al cliente, se la pasa al intérprete de **PHP**. Éste la interpreta y es el resultado de esta interpretación del programa **PHP**, contenido en la página **PHP**, lo que termina llegando al cliente.

Otras características:

- Licencia de software libre.
- Sintaxis cómoda.
- Soporta objetos y herencia.
- Extensa librería de funciones.
- Compatibilidad con bases de datos.
- Expansión.

Utilidades que aporta:

- Funciones de correo electrónico.
- Gestión de bases de datos.
- Gestión de archivos.
- Tratamiento de imágenes.

Además de las utilidades antes expuestas, la extensa librería de **PHP** cuenta con muchos más grupos de funciones entre los que cabe destacar: funciones *para Internet* (tratamiento de cookies, accesos restringidos, comercio electrónico, etc.) o funciones de *propósito general* (funciones matemáticas, funciones de fecha, funciones de cadenas, corrección ortográfica, compresión de archivos, etc.).

A esta inmensa librería hay que añadir todas las funciones personales que cada uno va creando en respuesta a sus propias necesidades y que luego pueden ser reutilizadas en otros sitios y todas aquellas intercambiadas u obtenidas en foros o sitios especializados.

Podemos afirmar, por lo tanto, que se trata de un lenguaje ideal tanto para las personas que empiezan a desarrollar como para desarrolladores experimentados [5] [6] [7].

1.6.4.3 Selección del lenguaje a utilizar.

Después de analizado lo anteriormente expuesto hemos decidido utilizar **PHP**, ya que es un lenguaje de código abierto que puede ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de Unix, Windows y Macs. Generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz. Está escrito en C, por lo que se ejecuta rápidamente utilizando poca memoria, optimizando así nuestros recursos de hardware. Es completamente expandible y consta de un conjunto de módulos y una gran variedad de extensiones de código. Puede interactuar con muchos motores de bases de datos y dispone de ODBC para situaciones que lo requieran.

También otras características que nos hicieron escoger este lenguaje para nuestro proyecto fueron su velocidad, estabilidad, seguridad y simplicidad; además de que pertenece al grupo de los softwares libres que tanto auge tienen hoy en la universidad y en el país.

1.6.5 Sistemas gestores de Bases de Datos (SGBD).

Rapidez, efectividad en los procesos y los grandes flujos de información están como primera necesidad a la hora de optimizar servicios y productos. Ante esta notable demanda de soluciones informáticas han surgido multitud de gestores de bases de datos, siendo estos programas los que permiten manejar la información de modo sencillo y que prestan servicios para el desarrollo y el manejo de bases de datos.

1.6.5.1 MySQL.

MySQL Server es la base de datos de código fuente abierto más usada del mundo desarrollado y proporcionado por MySQL AB. MySQL AB es una empresa cuyo negocio consiste en proporcionar servicios en torno al servidor de bases de datos **MySQL**. Su origen se debió a la búsqueda por parte de los fundadores de crear un manejador de bases de datos que fuera "rápido", todavía más rápido que mSQL. Así surgió **MySQL**, primero como un producto de la empresa y después como software de dominio público [8].

Características:

- Escrito en C y C++.
- Trabaja bajo diferentes plataformas.
- Desarrollo de APIs para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl..
- Procesos MultiHilo. Capacidad de trabajar servidores con varios procesadores.
- Provee sistema transaccional con la tabla Innodb.
- Velocidad cuando se manipula datos con el tipo de tabla Myisam.
- Velocidad en la utilización de joins y procesos de optimización.
- Soporta muchos tipos de columnas para las tablas: FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM y OpenGIS (Modelo Geométrico).
- Manejo de la memoria a través de manejo del buffer y cache.

1.6.5.2 SQL Server.

SQL Server es el sistema de gestión de base de datos representativa de la Microsoft. El rendimiento, la escalabilidad y la confiabilidad son esenciales. **SQL Server** proporciona agilidad a las operaciones de análisis y administración de datos. **SQL Server** es un producto de base de datos totalmente habilitado para Web que proporciona una compatibilidad fundamental con el lenguaje de marcado extensible (XML, Extensible Markup Language) y la capacidad para realizar consultas en Internet y por encima del servidor de seguridad [9].

Características:

- Seguridad.
- Alojamiento de aplicaciones.
- Escalabilidad.
- Administración de bases de datos simplificada.
- Alta Disponibilidad.

1.6.5.3 Selección del SGBD a utilizar.

Basado en lo antes expuesto y de acuerdo con el lenguaje de programación antes seleccionado, se va a utilizar **MySQL**, porque es el SGBD que se nos hace más compatible y eficiente para usar en este caso, además de que la UCI está abogando por la utilización del software libre, ya que éste tiene las mismas características que el software comercial. **MySQL** es rápido y eficiente, capaz de trabajar con servidores con varios procesadores, cuenta con una variedad de herramientas que hace más fácil la administración, su bajo consumo en recursos de hardware lo hacen apto para ser ejecutado en una máquina con mínimas características sin ningún problema. Además el conjunto de aplicaciones **Apache-PHP-MySQL** es uno de los más utilizados en ambiente Web.

1.6.6 Metodologías de desarrollo de software.

Uno de los principales problemas en la actualidad en el desarrollo de software es seleccionar la metodología más adecuada que posibilite obtener los resultados óptimos que se desean; o sea, cómo trabajar eficientemente evitando las catástrofes que conllevan al fracaso de un gran porcentaje de proyectos. Una metodología tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo, por medio de una mayor transparencia y control sobre el proceso. Analizaremos seguidamente tres de las más conocidas, sus características, ventajas y desventajas.

1.6.6.1 RUP (Proceso de Desarrollo Racional).

RUP es uno de los procesos más generales de los existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier proyecto, y no tan solo de software.

Un proyecto realizado siguiendo **RUP** se divide en cuatro fases:

1. Intercepción (puesta en marcha).
2. Elaboración (definición, análisis, diseño).
3. Construcción (implementación).
4. Transición (fin del proyecto y puesta en producción).

En cada fase se ejecutarán una o varias iteraciones (de tamaño variable según el proyecto), y dentro de cada una de ellas seguirá un modelo de cascada o *waterfall* para los flujos de trabajo que requieren las nuevas actividades anteriormente citadas.

RUP define nueve actividades a realizar en cada fase del proyecto:

1. Modelado del negocio.
2. Análisis de requisitos.
3. Análisis y diseño.
4. Implementación.
5. Test.
6. Distribución.
7. Gestión de configuración y cambios.

8. Gestión del proyecto.

9. Gestión del entorno.

RUP se basa en casos de uso para describir lo que se espera del software y está muy orientado a la arquitectura del sistema, documentándose lo mejor posible, basándose en UML (*Unified Modeling Language*) como herramienta fundamental para preparar planos de software. La notación UML se fundamenta en principios de modelado. En la construcción de los modelos se tiene en cuenta los detalles más importantes, por lo que se necesita más de un modelo para comprender el sistema desde varios frentes. UML recomienda la utilización de distintos diagramas, donde cada uno representa un aspecto del sistema. Además, establece una serie de requerimientos y estructuras necesarias, que permiten plasmar un sistema de software previo al proceso intensivo de escribir código [10]. La herramienta CASE de modelación visual que provee el modelado basado en UML es Rational Rose. Esta se utiliza para la etapa de análisis y diseño de sistemas. Posee plataforma independiente que ayuda a la comunicación entre los miembros de equipo y propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico.

1.6.6.2 XP (Programación Extrema).

XP, como toda metodología ágil, intenta reducir la complejidad del software por medio de un trabajo orientado al objeto, basado en las relaciones interpersonales y la velocidad de reacción. Intenta minimizar el riesgo de fallo del proceso por medio de la disposición permanente de un representante competente del cliente a disposición del equipo de desarrollo. Este representante debe estar en condiciones de contestar rápida y correctamente a cualquier pregunta del equipo de desarrollo de forma que no se retrase la toma de decisiones. Define *UserStories* como bases del software a desarrollar. Estas historias las escribe el cliente y describe escenarios sobre el funcionamiento del software, que no solo se limitan a los GUI sino también puede describir el

modelo, dominio, etc. A partir de los *UserStories* y de la arquitectura perseguida se crea un plan de *releases* entre el equipo de desarrollo y el cliente.

Para cada *release* se discutirán los objetivos de la misma con el representante del cliente y se definirán las iteraciones (de pocas semanas de duración) necesarias para cumplir con los objetivos de la *release*. El resultado de cada iteración es un programa que se transmite al cliente para que lo juzgue. En base a su opinión se definen las siguientes iteraciones del proyecto y si el cliente no está contento se adaptará el plan de *releases* e iteraciones hasta que el cliente de su aprobación y el software este a su gusto.

UserStories y casos de pruebas son la base sobre la que se asienta el trabajo del desarrollador. Se sigue un diseño evolutivo con la siguiente premisa: conseguir la funcionalidad deseada de la forma más sencilla posible. Este diseño evolutivo hace que no se le de apenas importancia al análisis como fase independiente, puesto que se trabaja exclusivamente en función de las necesidades del momento [10].

1.6.6.3 FDD (Desarrollo guiado por la funcionalidad).

FDD se puede considerar a medio camino entre **RUP** y **XP**, aunque al seguir siendo un proceso ligero, se asemeja más a este último. Está pensado para proyectos con tiempo de desarrollo relativamente cortos (menos de un año). Se basa en un proceso iterativo con iteraciones cortas (2 semanas). Las iteraciones de deciden en base a *features* (funcionalidades), que son pequeñas partes del software con significado para el cliente.

Un proyecto con metodologías **FDD** consta de las siguientes 5 fases:

1. Desarrollo de un modelo general.
2. Construcción de una lista de funcionalidades.
3. Plan de releases en base a las funcionalidades a implementar.
4. Diseñar en la base a las funcionalidades.
5. Implementar en base a las funcionalidades.

Las primeras tres fases ocupan gran parte del tiempo en las primeras iteraciones, siendo las dos ultimas las que absorben la mayor parte del tiempo

según va avanzando el proyecto, limitándose las primeras a un proceso de refinamiento. Las funcionalidades a implementar en un release se dividen entre los distintos subgrupos del equipo, y se procede a implementarlas [10].

1.6.6.4 Selección de la metodología a utilizar.

Después de haber analizado estas tres metodologías llegamos a la conclusión de que se va a utilizar en nuestro proyecto la metodología de desarrollo **RUP** puesto que es la más completa y abarcadora, ya que como señalan algunos autores, las otras metodologías son casos particulares de esta. Además **XP** y **FDD** presentan algunas debilidades, lo que representa riesgos considerables, como dificultades a la hora de una buena obtención de requisitos para el sistema.

1.7 Conclusiones.

En este capítulo se abordaron los aspectos teóricos de la gestión de los cursos optativos y de segundo perfil en las Universidad de las Ciencias Informáticas, realizándose un análisis crítico de los procesos, del cual se obtendrá el modelo de negocio a tener en cuenta para el **SiGesCOSP**. También se realiza un estudio sobre las diferentes tendencias y tecnologías actuales con las que contamos para la elaboración de nuestro proyecto, llegando a la conclusión que el lenguaje de programación **PHP**, combinado con el SGBD **MySQL**, sobre una plataforma Windows 2003 Advances Server o cualquier versión de Unix o Linux, se perfila como una propuesta tentadora para ser utilizada en la programación de nuestra aplicación Web, teniendo en cuenta que está en correspondencia con nuestras posibilidades y le da respuesta total a nuestras necesidades en este caso.

2.1 Introducción.

En el presente capítulo se describe y se modela todo lo referente al negocio propuesto, plasmándose todas las reglas y restricciones que rigen el proceso, identificándose los actores, trabajadores y los casos de uso correspondientes.

2.2 Modelo del negocio actual.

El movimiento de cursos optativos y de segundo perfil ha tomado un gran auge en la Universidad de las Ciencias Informáticas. El proceso se realiza actualmente de la siguiente forma: en el departamento central de Práctica Profesional se publican un conjunto de temas de cursos sobre los que se necesitan obtener programas analíticos, a partir de aquí las facultades envían sus propuestas de programas para estos temas, de ahí el departamento selecciona las propuestas que se ajusten más a sus necesidades y los publica para ser convocados a matrícula, además así enriquece el stock de programas de cursos que ellos poseen. También las facultades envían sus sugerencias de temas de acuerdo a sus necesidades de cursos optativos, y también de acuerdo a su perfil, para que el departamento los analice y los apruebe. De esta manera los cursos que son aprobados pueden ser del departamento central o de una facultad en específico, atendiendo a esto la matrícula se realiza en la citada facultad o en el departamento central. Se les envían las ofertas de cursos que están abiertos para matrícula a los estudiantes por el vía del correo electrónico y se publican en murales, pudiendo los estudiantes solicitar la matrícula en los cursos que deseen y se ajusten a sus necesidades, teniendo prioridad los estudiantes de la facultad que oferta el curso y si el curso es del departamento central todos los estudiantes tienen la misma prioridad.

2.3 Actores y Trabajadores del negocio.

Los actores del negocio son aquellas personas o sistemas que obtienen un resultado de valor de uno o varios procesos del negocio. Los actores del negocio estudiado se definen en la siguiente tabla:

Tabla 2.1 Actores del negocio.

Actor	Justificación
JefesDpto	Sugiere temas para los cursos y programas analíticos sobre los temas de cursos que se encuentran publicados esperando propuestas.
Estudiante	Realiza la solicitud para matricularse en un curso de los que se encuentran publicados y abiertos para matrícula.

Los trabajadores del negocio son roles representado una persona (o grupo de ellas) o un sistema automatizado que actúa en el negocio realizando una o varias actividades, interactuando con los actores y con otros trabajadores del negocio y manipulando las entidades del negocio. Los trabajadores que participan en nuestro negocio se relacionan en la siguiente tabla:

Tabla 2.2 Trabajadores del negocio.

Trabajador	Justificación
Administrador	Se responsabiliza de aprobar o denegar los programas analíticos de los cursos y los temas que se necesita desarrollar. Publica los temas sobre los que se necesitan programas de cursos optativos.
Secretaria	Divulga los cursos que oferta su área y tiene la responsabilidad de realizar las matrículas.

2.4 Reglas del negocio.

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. Después

de analizar detenidamente el proceso que nos ocupa se identificaron las siguientes reglas que debe cumplir nuestro sistema:

- La *Plantilla de Temas* es el documento que contiene la relación de todos los temas que proporciona el departamento central de Práctica Profesional sobre los cuales se necesitan programas analíticos. Es creada y modificada por el Administrador, el cual es el encargado de agregar y eliminar temas en la misma.
- El *Programa Analítico* de un curso es el documento que contiene toda la documentación del mismo y se elabora a partir de un formato que es proporcionado por el departamento central de Práctica Profesional. Es creado por los JefesDpto.
- La *Planilla de Curso* es el documento oficial de un curso que contiene información general de este, donde están incluidos el programa analítico, el registro de matriculados y los demás elementos como horario, lugar, profesor, cantidad de matrícula, entre otros datos necesarios. Es creada por el Administrador y modificada por la Secretaria al matricular los estudiantes interesados en el curso dado.
- La cantidad de propuestas de programas analíticos que se aceptan para un curso es un número que determina el departamento central de Práctica Profesional para cada tema de la *Plantilla de Temas*.
- Un tema solo puede tener un solo programa analítico. Este se selecciona de la variedad de propuestas que se reciben.
- Sobre un tema solo puede haber un curso abierto para matrícula y este puede tener varias convocatorias.
- El Administrador es la persona autorizada del departamento central de Práctica Profesional que puede publicar temas para recibir propuesta de programas analíticos y cursos que ya han sido aprobado su programa analítico.
- Los JefesDpto son los profesores autorizados de los distintos departamentos de las facultades para enviar los programas analíticos y

sugerir temas nuevos para cursos al departamento central de Práctica Profesional.

- Los cursos tienen dos clasificaciones de acuerdo a quien los ofrece: puede ser del departamento central de Práctica Profesional o de una facultad en específico. En los cursos centrales pueden solicitar la matrícula los estudiantes de cualquier facultad, mientras que los cursos específicos de una facultad tienen prioridad en la solicitud de matrícula los estudiantes de dicha facultad y podrán matricularse estudiantes de otras facultades en caso de quedar plazas vacantes.
- La Secretaria es la persona encargada de realizar las matrículas de los estudiantes que lo soliciten y que sean aceptadas. Puede ser del departamento central de Práctica Profesional o de una facultad específica de acuerdo a la clasificación del curso.
- El Estudiante es la persona que realiza la solicitud de matrícula en los cursos que hayan sido abiertos con este fin.

2.5 Diagrama de casos de uso del negocio.

El diagrama de Casos de Uso del negocio muestra gráficamente la interacción de los actores con los procesos del negocio.

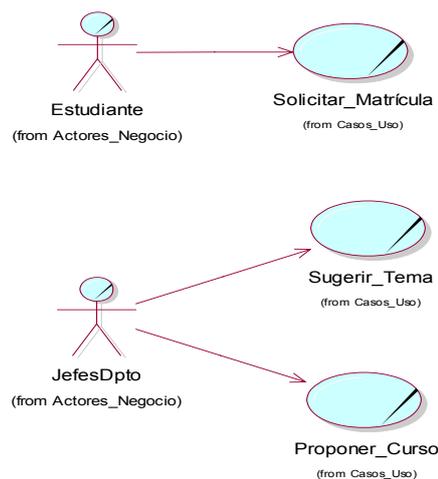


Fig. 2.1 Diagrama de Casos de Uso del negocio.

2.6 Casos de uso del negocio.

2.6.1 CU1 Proponer curso.

Tabla 2.3 CU1 Proponer curso.

CU1	Proponer programa analítico de un curso.
ACTORES	JefesDpto (Inicia)
Resumen: El caso de uso se inicia cuando uno de los Jefes de Departamento de un departamento de una facultad le envía al departamento central de Práctica Profesional un programa analítico de un curso sobre determinado tema de los que se encuentran publicados esperando propuestas, y finaliza cuando el documento es analizado y se determina si se ofertará como curso optativo o de segundo perfil dicho programa analítico.	

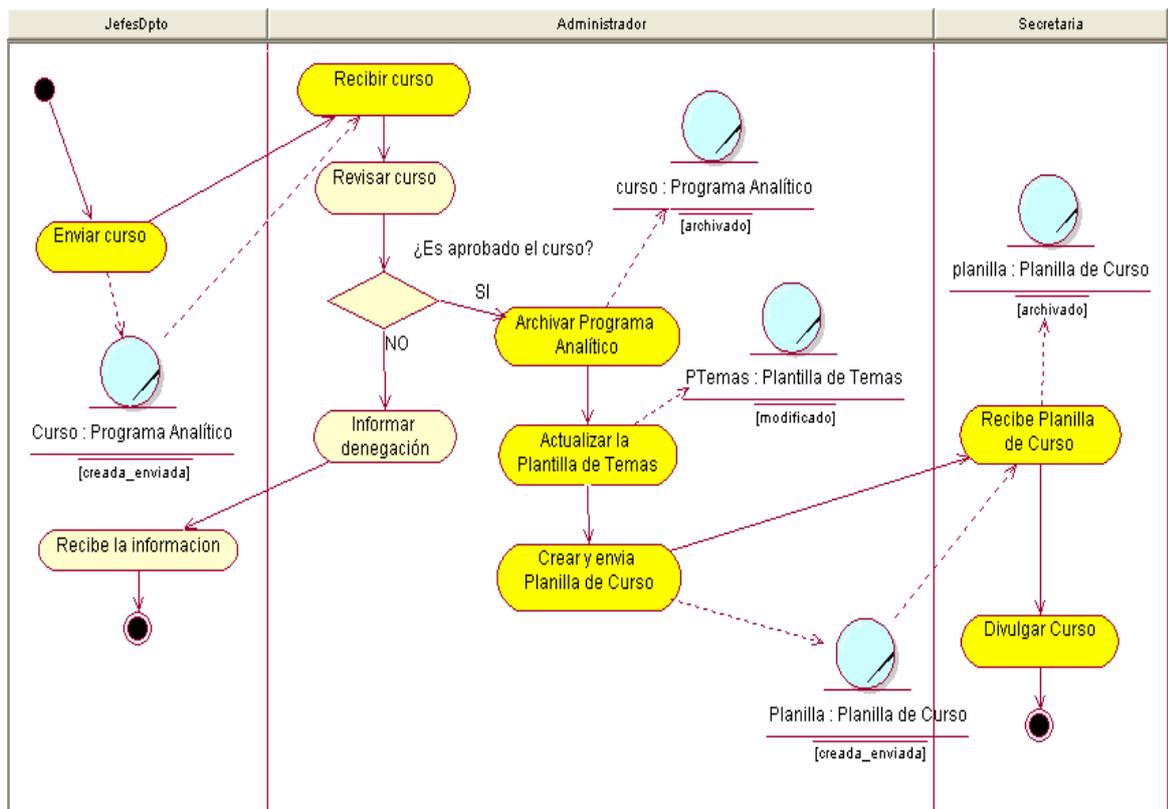


Fig. 2.2 Diagrama de Actividades del Caso de Uso Proponer curso.

2.6.2 CU2 Sugerir tema.

Tabla 2.4 CU2 Sugerir tema.

CU2	Sugerir tema para un curso.
ACTORES	JefesDpto (Inicia)
Resumen: El caso de uso se inicia cuando uno de los Jefes de Departamento de un departamento de una facultad le envía al departamento central de Práctica Profesional una tema para un curso sobre el cual necesitan propuestas de programas analítico y finaliza cuando el tema es analizado y se determina si se publicará como posible tema para curso optativo o de segundo perfil o se descarta dicho tema.	

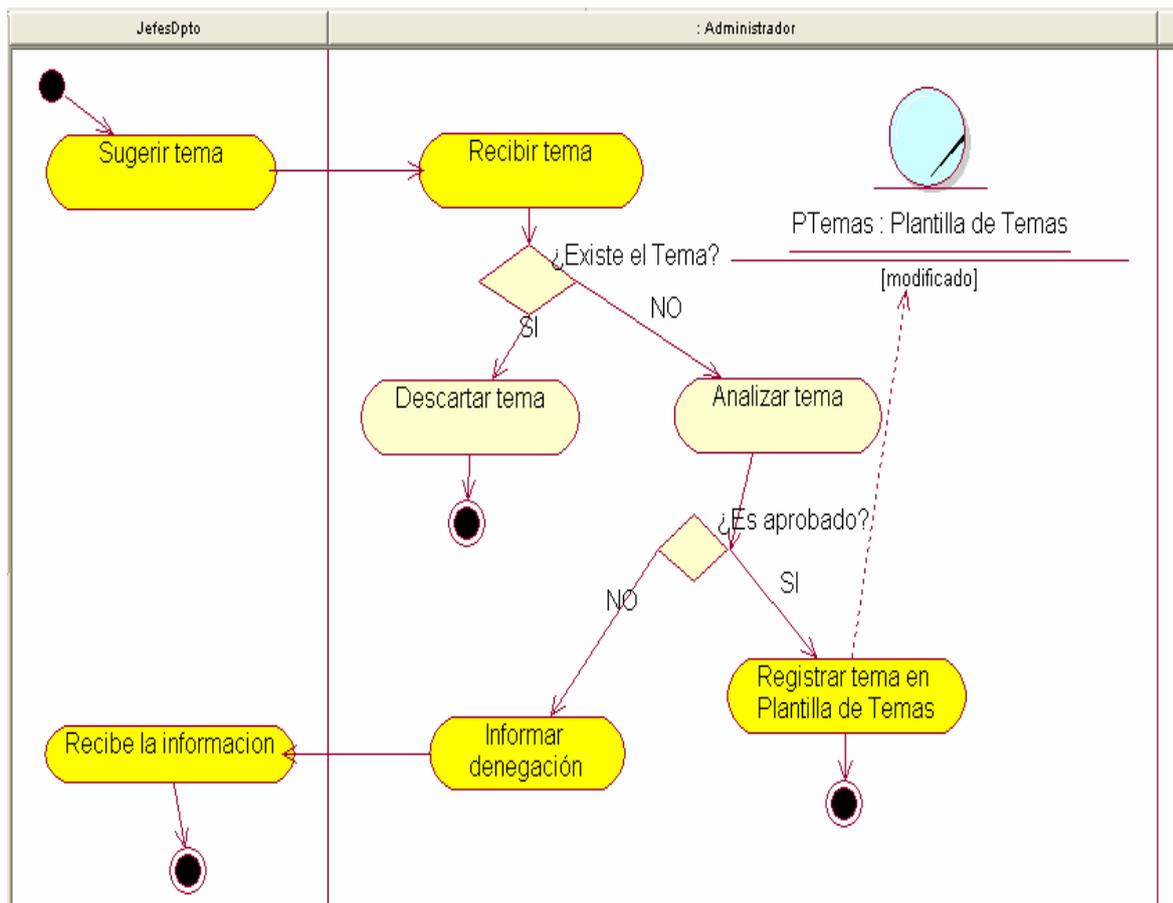


Fig. 2.3 Diagrama de Actividades del CU2 Sugerir tema.

2.6.3 CU3 Solicitar matrícula.

Tabla 2.5 CU2 Sugerir tema.

CU3	Solicitar matrícula en un curso.
ACTORES	Estudiante (Inicia)
Resumen: El caso de uso se inicia cuando un estudiante solicita la matrícula en un curso de los que se encuentran en oferta, se analiza la solicitud, si es viable y cumple con todas las disposiciones se realiza la matrícula; de lo contrario se deniega la solicitud de matrícula. El caso de uso finaliza cuando el estudiante es informado de su matrícula en el curso o de la denegación de la misma.	

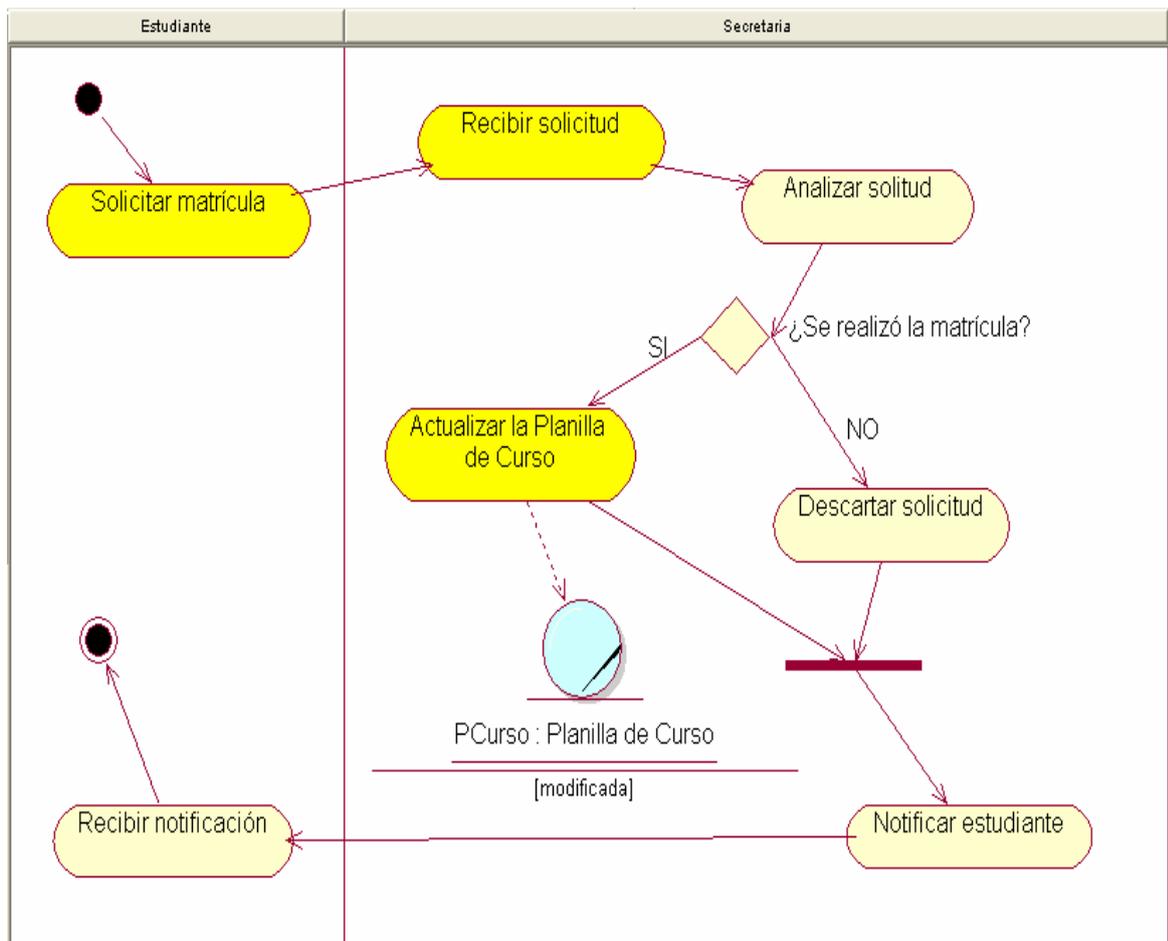


Fig. 2.4 Diagrama de Actividades del CU3 Solicitar matrícula.

2.7 Modelo de objeto.

El Modelo de Objeto muestra gráficamente la interacción entre los trabajadores y las entidades del negocio. Para nuestro negocio tenemos el esquema siguiente.

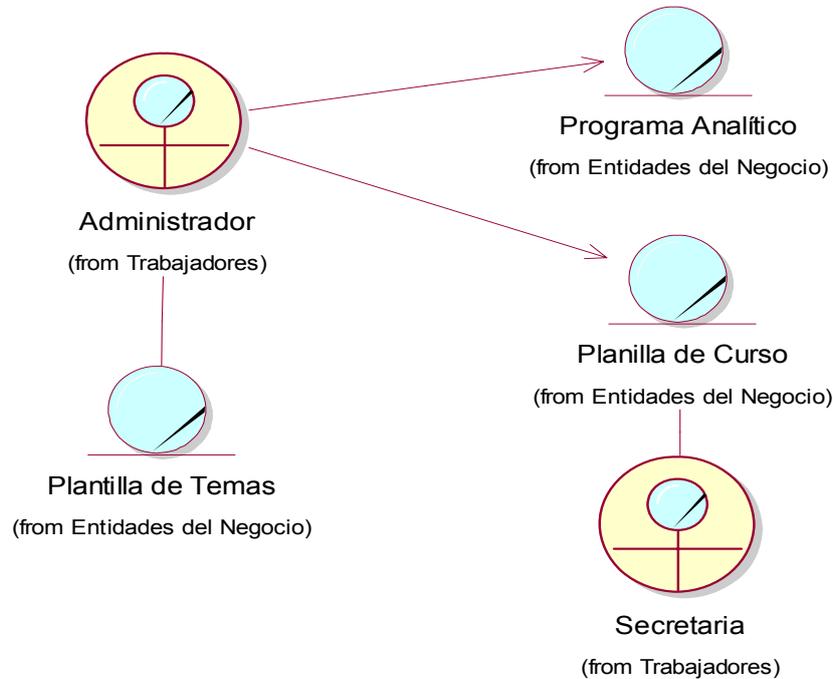


Fig. 2.5 Diagrama del Modelo de Objeto.

2.8 Conclusiones.

En este capítulo se realizó un análisis detallado del negocio, quedaron bien definidos los actores y trabajadores, así como los casos de usos, los cuales se modelaron a través del diagrama de casos de uso del negocio; también se hizo una descripción ampliada de los mismos, donde se esquematizan todas las acciones que se realizan en cada uno de ellos, finalizando con el modelo de objeto. Todo esto nos ayuda a comprender mejor los procesos involucrados en el campo de acción, con el objetivo de determinar a partir de aquí todos los requerimientos que de tener nuestro sistema y comenzar así con la captura de requisitos.

3.1 Introducción.

En este capítulo haremos una descripción completa del sistema detallando los requisitos funcionales y no funcionales, así como los actores y trabajadores y los casos de usos, llegando a obtener el prototipo total del sistema que vamos a diseñar posteriormente.

3.2 Especificación de requerimientos.

3.2.1 Requisitos funcionales.

R1 Autenticar usuario.

- 1.1 Comparar usuario y contraseña con los usuarios del sistema.
- 1.2 Asignar privilegios.

R2 Buscar datos de los cursos.

- 2.1 Mostrar listado de cursos disponibles.
- 2.2 Mostrar los datos de un curso.

R3 Gestionar cursos propuestos.

- 3.1 Listar los cursos propuestos.
- 3.2 Publicar cursos propuestos.
- 3.3 Eliminar cursos propuestos.

R4 Gestionar cursos publicados.

- 4.1 Listar los cursos publicados.
- 4.2 Modificar cursos publicados.
- 4.3 Eliminar cursos publicados.

R5 Gestionar temás.

- 5.1 Enviar temás.
- 5.2 Eliminar temás.
- 5.2 Publicar temás.
- 5.3 Listar temás sugeridos.
- 5.4 Listar temás publicados.

R6 Insertar programa analítico de un curso.

R7 Realizar matrícula en un curso.

- 7.1 Listar solicitudes de matrículas.
- 7.2 Matricular estudiante.
- 7.3 Enviar notificación a los estudiantes matriculados.

R8 Solicitar matrícula para un curso.

R9 Gestionar usuarios.

3.2.2 Requisitos no funcionales.

Software

1. El sistema se desarrollara con tecnología **PHP**, por lo que no se necesita un servidor en específico, puede ser Apache o Internet Information Server.
2. Se podrá utilizar un servidor con plataforma de funcionamiento Windows 2003 Server o cualquier versión de Unix o Linux.
3. En las terminales clientes solo será necesario garantizar un navegador Web.
4. La comunicación de las terminales clientes con el servidor será a través de conexiones a una velocidad constante de 100 Mbps.

5. El sistema utilizará una Base de Datos implementada en **MySQL**.

Hardware

1. El servidor debe tener los siguientes requerimientos técnicos: procesador Pentium III 1 GHz o superior, 256 MB de memoria RAM (incluye la utilizada por el SO) y 1 Gb de capacidad en disco duro.
2. En el cliente se requiere un procesador Pentium con 64 MB de memoria RAM, resolución de pantalla a 800x600 o superior.

Apariencia o interfaz externa.

1. La interfaz será amigable y sencilla.
2. Cualquier usuario debe tener posibilidad de interactuar con la información sin necesidad de ayuda ni tutoriales.
3. No debe usarse frames.
4. Cada página no debe exceder los 300Kb en las imágenes.

Portabilidad

El sistema puede correr sobre Windows, pero dado que el servidor es Apache (multiplataforma) permite que el servidor pueda ser cambiado a Linux, efectuándose esta reestructuración sin necesidad de efectuar cambios significativos.

Requerimiento de soporte

Se necesita de cualquier servidor con Windows 2003 Server o cualquier versión de Linux y Apache instalado, además del servidor de Base de Datos **MySQL**. No se requiere que corran el servidor Web y el de base de datos en el mismo servidor, recomendamos esta opción.

Usabilidad

El software podrá ser usado por cualquier usuario con mínimos conocimientos de utilización de ordenadores y del ambiente Web en sentido general.

Rendimiento

Se debe garantizar la rapidez de respuesta del sistema ante las solicitudes de los usuarios. Como la aplicación está concebida para un ambiente cliente/servidor debe garantizarse que los tiempos de respuestas sean generalmente rápidos al igual que la velocidad de procesamiento de la información. Se sigue una lógica de negocios en la comunicación entre el cliente y el servidor Web, que cada cual procesa lo que le corresponda.

Seguridad y privacidad

El sistema contendrá la información específica para los diferentes tipos de usuarios, de acuerdo al nivel de privilegio de cada uno. Por eso es de vital importancia la total integridad de la información que se gestione, para ello se autenticará cada usuario que acceda al sitio, haciendo las verificaciones pertinentes con la base de datos UCI y con la propia del sistema, y partir de los resultados que arrojen estos controles se otorgaran los privilegios correspondientes. Para asegurar todo esto además el sistema debe comunicarse con un protocolo seguro y los datos no pueden viajar de forma transparente por la red.

3.3 Descripción del sistema propuesto.

Para darle solución a los objetivos propuestos al inicio de este trabajo, y teniendo en cuenta todos los requerimientos planteados, el sistema que se propone debe tener las características siguientes:

Para utilizar todas las funcionalidades del sistema, se determina la existencia de 4 roles: un usuario general que se autentica en el sistema y obtiene los privilegios correspondientes, un estudiante que puede ver toda la información

sobre los cursos y solicitar la matrícula en alguno de ellos, un jefe de departamento que accede a la relación de temas y cursos publicados y puede enviar una sugerencia de tema para un curso y también puede enviar una propuesta de programa analítico para un tema de curso que se encuentre publicado, una secretaria que accede a las solicitudes de matrículas y realiza las mismas en los casos aprobados y viables, y un administrador que accede a toda la información y es el que publica los temas para cursos, aprueba los programas analíticos de los cursos, abre las ofertas de matrículas y gestiona los usuarios. En el SGBD que se va a utilizar se controlarán los datos y el rol de cada persona que quiera acceder al sistema.

En resumen, con este sistema los Jefes de Departamento de las distintas facultades podrán sugerir temas para cursos optativos y de segundo perfil, así como enviar los programas analíticos para los cursos, los administradores del departamento central de Práctica Profesional podrán publicar los temas para cursos y los curso que se aprueben y se oferten para matrícula, y los estudiantes podrán solicitar la matrícula en los cursos que se encuentren abiertos.

3.4 Modelo de Casos de Uso del sistema.

En el modelo de Casos de Uso del sistema se muestran todos los elementos por los que esta compuesto nuestro futuro proyecto de software, entre los que se encuentran los actores y los casos de uso con su descripción detallada, todo esto encaminado al cumplimiento de los requisitos funcionales expuestos anteriormente.

3.4.1 Actores del sistema.

Los actores del sistema representan los roles que desempeñan una o varias personas, o un sistema automatizado y que interactúan con nuestra aplicación siendo protagonistas de los casos de uso.

Tabla 3.1 Actores del sistema.

Actor	Descripción
Usuario	Se autentica en el sistema y luego con los permisos específicos que se le otorgan accede a las funcionalidades que brinda la aplicación.
Estudiante	Tiene acceso a toda la información sobre los cursos que se encuentran publicados y tiene la posibilidad de solicitar matrícula en uno o varios de ellos.
JefesDpto	Tiene acceso a toda la información de los cursos y los temas publicados y puede proponer programas analíticos para dichos temas de cursos y sugerir nuevos temas para cursos.
Secretaria	Tiene acceso a toda la información sobre los cursos publicados y a las solicitudes de matrículas echas por los estudiantes, siendo la encargada de aprobar o denegar dichas solicitudes.
Administrador	Administra y controla toda la información del sitio, tiene todos los privilegios.

3.4.2 Diagrama de Casos de Uso.

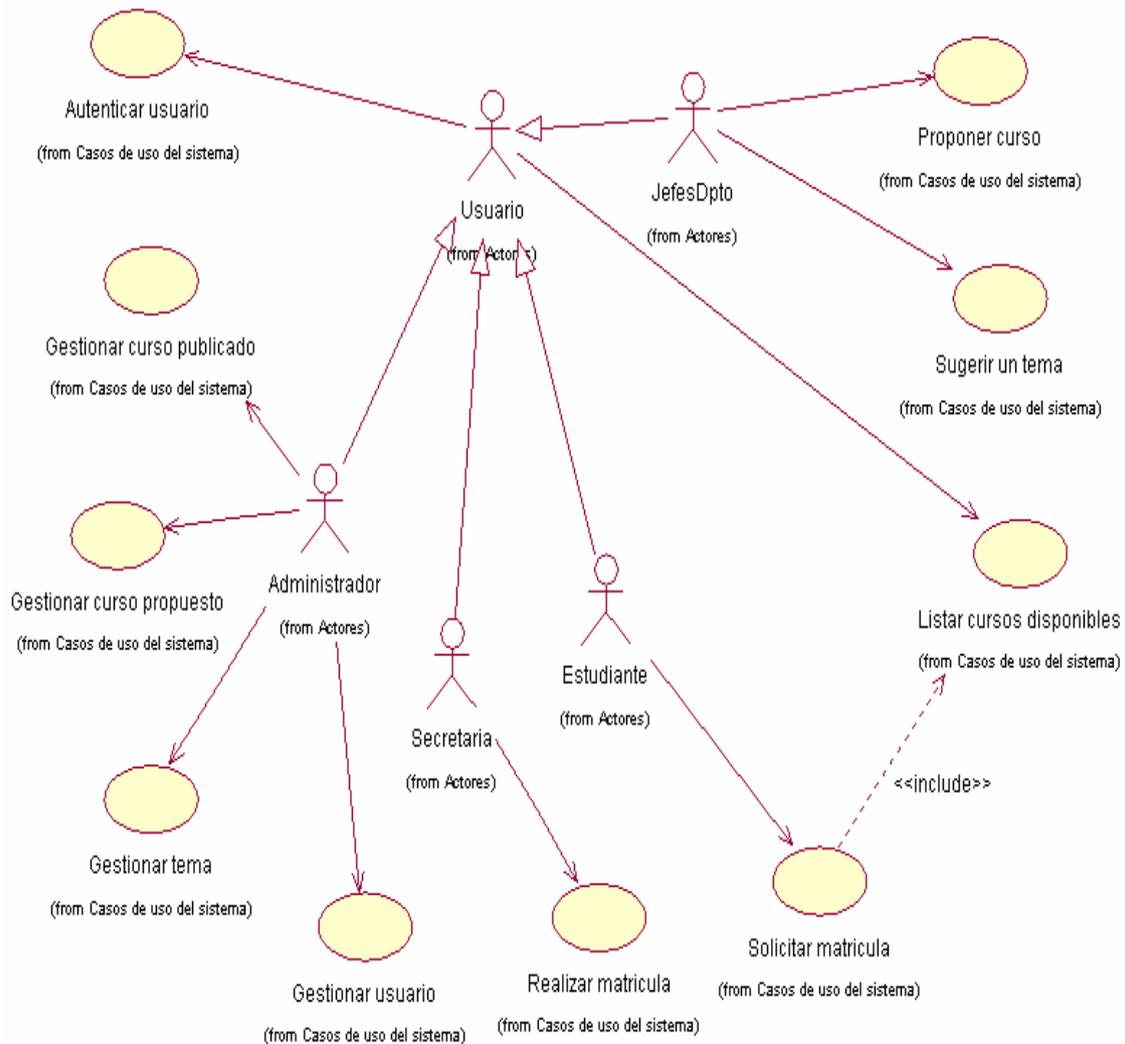


Fig. 3.1 Diagrama de Casos de Uso del sistema.

3.5 Descripción detallada de los Casos de Uso del sistema.

Los casos de usos determinados para nuestro sistema son los siguientes:

- 🚩 CU1 – Autenticar usuario.
- 🚩 CU2 – Navegar cursos.
- 🚩 CU3 – Gestionar cursos propuestos.
- 🚩 CU4 – Gestionar cursos publicados.
- 🚩 CU5 – Proponer curso.

- 🚩 CU6 – Sugerir tema.
- 🚩 CU7 – Realizar matrícula.
- 🚩 CU8 – Gestionar tema.
- 🚩 CU9 – Listar cursos disponibles.
- 🚩 CU10 – Gestionar usuario.

3.5.1 CU1 Autenticar usuario.

Tabla 3.2 CU1 Autenticar usuario.

Caso de Uso	Autenticar usuario.
Actores	Usuario
Propósito	Permitir autenticarse en el sistema.
Resumen	El caso de uso se inicia cuando el Usuario introduce su loguin y contraseña del dominio, luego el sistema determina el tipo de usuario a que pertenece, le asigna los privilegios de acceso y finaliza habilitándole las funcionalidades que le corresponden de acuerdo al rol que desempeña.
Referencia	<i>R1</i>
Precondiciones	
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1. El Usuario introduce su loguin y contraseña.	<p>1.1 El sistema consulta con la base de datos UCI la existencia del loguin y la contraseña, determinando la existencia del usuario y el tipo de usuario.</p> <p>1.2 Si existe el usuario y es del tipo profesor, el sistema consulta su base de datos, determinando si es de los profesores privilegiados (JefesDpto) o si es el Administrador, asignándole los privilegios correspondientes.</p>

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	1.3. En caso de no existir el usuario en la base de datos de la UCI el sistema muestra un mensaje de error.
Prioridad	Crítico
Poscondiciones	El Usuario entra en el sistema con sus privilegios establecidos y se le habilitan las funcionalidades según los privilegios.

3.5.2 CU2 Solicitar matrículas.

Tabla 3.3 CU2 Solicitar matrícula.

Caso de Uso	Solicitar matrículas.	
Actores	Estudiante	
Propósito	Permitir ver los cursos disponibles y realizar la solicitud de la matrícula en cursos abiertos.	
Resumen	El caso de uso se inicia cuando el Estudiante solicita ver los cursos que están a publicados en el sitio, y finaliza cuando la aplicación le muestra el listado de los cursos, y realiza o no el pedido de matrícula.	
Referencia	<i>R2, R6, CU9 Listar cursos disponibles</i>	
Precondiciones	Haberse autenticado en el sistema como estudiante.	
Flujo Normal de Eventos		
Sección “”		
Acción del Actor	Respuesta del Sistema	
1. El Estudiante selecciona el vínculo que le mostrara los cursos disponibles.	1.1 El sistema muestra el listado de los cursos disponibles.	
2. El Estudiante selecciona el		

curso para ver su información.	2.1. El sistema muestra toda la información relacionada con el curso seleccionado.
3. El Estudiante realiza la solicitud de matrícula en el curso seleccionado.	3.1 Si el curso seleccionado esta abierto para matrícula el sistema realiza el envío de la solicitud de matrícula al banco de solicitudes.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3.1 Si el curso seleccionado está cerrado se muestra un mensaje de error.
Prioridad	Crítica
Poscondiciones	Se actualiza la información de las solicitudes de matrículas en la base de datos.

3.5.3 CU3 Gestionar cursos propuestos.

Tabla 3.4 CU3 Gestionar cursos propuestos.

Caso de Uso	Gestionar cursos propuestos.
Actores	Administrador
Propósito	Permitir publicar y eliminar cursos propuestos.
Resumen	El caso de uso se inicia cuando el administrador accede a la aplicación con el objetivo de publicar los cursos propuestos que sean aprobados y eliminar los cursos propuestos que sean desechados.
Referencia	R3
Precondiciones	Haberse autenticado en el sistema como

Administrador.

Flujo Normal de Eventos

Sección “”

Acción del Actor	Respuesta del Sistema
1. El Administrador accede a la aplicación para publicar un curso y eliminar los cursos desechados.	1.1 La aplicación muestra un listado de todos los temas publicados. El sistema ejecuta alguna de las acciones siguientes. a) Si decide publicar un curso ir a la sección "Publicar curso". b) Si decide eliminar un curso ir a la sección "Eliminar curso".

Sección “Publicar curso”

Acción del Actor	Respuesta del Sistema
2. El Administrador selecciona el tema del que desea ver los cursos propuestos.	2.1 El sistema muestra la lista de los cursos propuesto sobre el tema seleccionado.
3. El Administrador selecciona el curso que desea aprobar y publicar.	3.1 El sistema muestra los datos del curso seleccionado y pide la información adicional que se debe entrar.
4. El Administrador introduce los datos solicitados.	4.1 El sistema publica el curso, elimina el resto de los cursos del tema y elimina el

	tema de los temas publicados.
	4.2 El sistema muestra un mensaje de confirmación.
Sección “Eliminar cursos”	
Acción del Actor	Respuesta del Sistema
5. El Administrador selecciona el tema del que desea ver los cursos propuestos.	5.1 El sistema muestra la lista de los cursos propuesto sobre el tema seleccionado.
6. El Administrador selecciona el curso que desea eliminar.	6.1 El sistema elimina el curso seleccionado. 6.2 El sistema actualiza los datos y muestra un mensaje de confirmación.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Prioridad	Crítica
Poscondiciones	Se actualiza la información de los cursos propuestos en la base de datos y de los temas y cursos publicados.

3.5.4 CU4 Gestionar cursos publicados.

Tabla 3.5 CU4 Gestionar cursos publicados.

Caso de Uso	Gestionar cursos publicados.
Actores	Administrador
Propósito	Permitir modificar y quitar cursos publicados.

Resumen	El caso de uso se inicia cuando el Administrador accede a la aplicación con el objetivo de modificar cursos publicados y quitar cursos publicados que hayan finalizado.
Referencia	R4
Precondiciones	Haberse autenticado en el sistema como Administrador
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1. El Administrador accede a la aplicación para modificar un curso publicado y eliminar los cursos publicados que hayan finalizado.	1.1 La aplicación muestra un listado de todos los cursos publicados. El sistema ejecuta alguna de las acciones siguientes: <ul style="list-style-type: none"> a) Si decide modificar un curso ir a la sección "Modificar curso". b) Si decide quitar un curso ir a la sección "Quitar curso".
Sección “<i>Modificar curso</i>”	
Acción del Actor	Respuesta del Sistema
2. El Administrador selecciona el curso que desea modificar.	2.1 El sistema muestra los datos completos del curso.
3. El Administrador entra los nuevos datos para modificar los anteriores.	3.1 El sistema guarda los datos y actualiza la información.

	3.2 El sistema muestra un mensaje de confirmación.
Sección “Quitar curso”	
Acción del Actor	Respuesta del Sistema
4. El Administrador elimina los cursos finalizados.	4.1 El sistema actualiza la información. 4.2 El sistema muestra un mensaje de confirmación.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Prioridad	Crítica
Poscondiciones	Se actualizan la información de los cursos publicados en la base de datos.

3.5.5 CU5 Proponer curso.

Tabla 3.6 CU5 Proponer curso.

Caso de Uso	Proponer curso
Actores	JefesDpto
Propósito	Permitir enviar propuestas de programas analíticos de cursos al departamento central de Práctica Profesional.
Resumen	El caso de uso se inicia cuando el JefesDpto de una facultad accede a la aplicación con el objetivo de proponer un programa analítico para un curso y finaliza cuando se haya enviado dicho programa analítico al banco de propuestas.
Referencia	R6
Precondiciones	Haberse autenticado en el sistema como JefesDpto.
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema

1. El JefesDpto accede a la aplicación para proponer un programa analítico de para un curso.	1.1 El sistema muestra el listado de temas publicados.
2. El JefesDpto selecciona el tema sobre el cual desean enviar su programa analítico.	2.1 El sistema muestra detalles del tema seleccionado.
3. El JefesDpto envía el programa analítico sobre el tema.	3.1 El sistema guarda el archivo enviado en el banco de cursos propuestos.
	4. El sistema muestra un mensaje de confirmación.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Prioridad	Crítica
Poscondiciones	Se actualiza la información de los cursos propuestos en la base de datos.

3.5.6 CU6 Sugerir tema.

Tabla 3.7 CU6 Sugerir tema.

Caso de Uso	Sugerir tema
Actores	JefesDpto
Propósito	Permitir enviar una sugerencia tema para un curso.
Resumen	El caso de uso se inicia cuando el JefesDpto de una facultad accede a la aplicación con el objetivo de sugerir un

	tema para curso y finaliza cuando se haya enviado dicha sugerencia de tema al banco de temas.
Referencia	R5
Precondiciones	Haberse autenticado en el sistema como JefesDpto.
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1. El JefesDpto accede a la aplicación para sugerir un tema para un curso.	<p>1.1 El sistema permite que el JefesDpto sugiera un tema para un curso.</p> <p>1.2 El sistema envía el tema sugerido al banco de temas.</p> <p>1.3 El sistema muestra un mensaje de confirmación.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Prioridad	Crítica
Poscondiciones	Se actualiza la información de los temas sugeridos en la base de datos.

3.5.7 CU7 Realizar matrícula.

Tabla 3.8 CU7 Realizar matrícula.

Caso de Uso	Realizar matrícula.
Actores	Secretaria.
Propósito	Permitir matricular a los estudiantes en los cursos.
Resumen	El caso de uso se inicia cuando la Secretaria accede a la aplicación con el objetivo de matricular a los estudiantes que y finaliza cuando se realizan o no las matrículas.
Referencia	R7

Precondiciones	Haberse autenticado en el sistema como Secretaria.	
Flujo Normal de Eventos		
Sección “”		
	Acción del Actor	Respuesta del Sistema
	1. La Secretaria accede a la aplicación con el objetivo de chequear las solicitudes de matrícula.	1.1 El sistema muestra el listado de las solicitudes de matrículas con todos sus datos.
	2. La Secretaria marca los estudiantes que serán matriculados y selecciona matricularlos.	2.1 El sistema matrícula los estudiantes. 2.2 El sistema muestra un mensaje de confirmación. 2.3 El sistema envía un mensaje de correo electrónico a los estudiantes que solicitaron las matrículas confirmando o denegando la solicitud.
		.
Flujos Alternos		
	Acción del Actor	Respuesta del Sistema
Prioridad	Crítica	
Poscondiciones	Se actualiza la información de las matrículas en la base de datos.	

3.5.8 CU8 Gestionar temas.

Tabla 3.9 CU8 Gestionar temas.

Caso de Uso	Gestionar temas.
Actores	Administrador

Propósito	Permitir publicar y eliminar temas.
Resumen	El caso de uso se inicia cuando el Administrador accede a la aplicación con el objetivo de publicar y eliminar temas.
Referencia	R5
Precondiciones	Haberse autenticado en el sistema como Administrador.
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1. El Administrador accede a la aplicación con el objetivo de publicar y eliminar temas.	<p>1.1 El sistema ejecuta una de las acciones siguientes:</p> <p style="margin-left: 40px;">a) Si decide publicar un tema a la sección "Publicar tema".</p> <p style="margin-left: 40px;">b) Si decide eliminar un tema a la sección "Eliminar tema".</p>
Sección “Publicar tema”	
Acción del Actor	Respuesta del Sistema
2. El Administrador desea publicar un tema.	2.1 El sistema muestra la lista de temas sugeridos y la opción de publicar.
3. El Administrador selecciona el tema que desea publicar de la lista o introduce uno nuevo.	<p>3.1 El sistema publica el tema seleccionado o entrado.</p> <p>3.2 El sistema muestra un mensaje de confirmación.</p>
Sección “Eliminar tema”	

Acción del Actor	Respuesta del Sistema
4. El Administrador desea eliminar un tema.	4.1 El sistema muestra la lista de temas sugeridos.
5. El Administrador selecciona el tema que desea eliminar.	5.1 El sistema elimina el tema. 5.2 El sistema muestra un mensaje de confirmación.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Prioridad	Crítico
Poscondiciones	Se actualiza la información de los temas sugeridos y publicados en la base de datos.

3.5.9 CU9 Listar cursos disponibles.

Tabla 3.10 CU9 Listar cursos publicados.

Caso de Uso	Listar cursos disponibles.
Actores	Usuario
Propósito	Permitir ver los cursos que están publicados en el sistema.
Resumen	El caso de uso se inicia cuando el Usuario su accede a la aplicación con el objetivo de ver el listado de los cursos que están publicados y finaliza cuando se muestra el listado.
Referencia	<i>R4</i>
Precondiciones	Haberse autenticado en el sistema como Usuario.
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1. El Usuario	1.1 El sistema le muestra el listado de los cursos que se

accede a la aplicación con el objetivo de ver la lista de los cursos que se encuentran publicados.	encuentran publicados.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	1.3. En caso de no existir ningún curso publicado el sistema muestra un mensaje de error.
Prioridad	Crítico
Poscondiciones	

3.5.10 CU10 Gestionar usuarios.

Tabla 3.11 CU10 Gestionar usuarios.

Caso de Uso	Gestionar usuarios.
Actores	Administrador
Propósito	Permitir insertar y eliminar usuarios con roles.
Resumen	El caso de uso se inicia cuando el Administrador accede a la aplicación con el objetivo de insertar y eliminar usuarios con roles y finaliza cuando se realiza la operación solicitada.
Referencia	<i>R10</i>
Precondiciones	Haberse autenticado en el sistema como Administrador.
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1. El Administrador accede a la aplicación con el objetivo de insertar y eliminar usuarios con	1.1 El sistema ejecuta una de las acciones siguientes: a) Si decide insertar un usuario con rol ir a la

roles.	sección "Insertar usuario". b) Si decide eliminar un usuario con rol ir a la sección "Eliminar usuario".
Sección "Insertar usuario"	
Acción del Actor	Respuesta del Sistema
2. El Administrador desea insertar un usuario con rol.	2.1 El sistema pide los datos del usuario.
3. El Administrador entra los datos.	3.1 El sistema inserta el usuario con rol. 3.2 El sistema muestra un mensaje de confirmación.
Sección "Eliminar usuario"	
Acción del Actor	Respuesta del Sistema
4. El Administrador desea eliminar un usuario con rol.	4.1 El sistema muestra la lista de usuarios con roles existentes.
5. El Administrador selecciona el usuario con rol que desea eliminar.	5.1 El sistema elimina el usuario con rol. 5.2 El sistema muestra un mensaje de confirmación.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Prioridad	Crítico

Poscondiciones

Se actualiza la información de los usuarios con roles en la base de datos.

3.6 Conclusiones.

En este capítulo quedó bien determinado el sistema que vamos a implementar, quedando completamente definidos los actores y trabajadores del mismo. También analizamos el diagrama de casos de uso del sistema y se realizó la descripción detallada de cada uno de los casos de uso, con sus correspondientes detalles. Ya basado en todo esto podemos dar paso al siguiente capítulo donde realizaremos el diseño de la aplicación.

4.1 Introducción.

El análisis de un sistema se centra en la investigación del problema y no en la manera de definir una solución, mientras que el diseño pone de relieve una solución lógica: cómo el sistema satisface los requerimientos funcionales, requerimientos de calidad y las restricciones, es decir, en esta parte del proceso de desarrollo del software se decide como se va a llevar a cabo el mismo.

En este capítulo se realiza el análisis y diseño de la propuesta de solución, se presenta el diagrama de clases del análisis y del diseño de los diferentes casos de usos, se muestra el modelo lógico y físico de datos, se especifican los principios de diseño así como una concepción general del tratamiento de los errores que pueden aparecer. Finalmente se realiza el modelo de despliegue y el de componentes para una mejor descripción de la solución propuesta.

4.2 Diagramas de interacción.

Los diagramas de interacción están constituidos por dos tipos: los diagramas de colaboración y los diagramas de secuencia. Ambos expresan información similar, pero en una forma diferente.

Diagrama de colaboración

Muestra la interacción entre varios objetos y los enlaces que existen entre ellos. Representa las interacciones entre objetos organizadas alrededor de los objetos y sus vinculaciones.

Diagramas de secuencia

Muestran las interacciones entre un conjunto de objetos, ordenadas según el tiempo en que tienen lugar. Representa una forma de indicar el período durante el que un objeto está desarrollando una acción directamente o a través de un procedimiento [11].

4.2.1 CU1 Autenticar usuario.

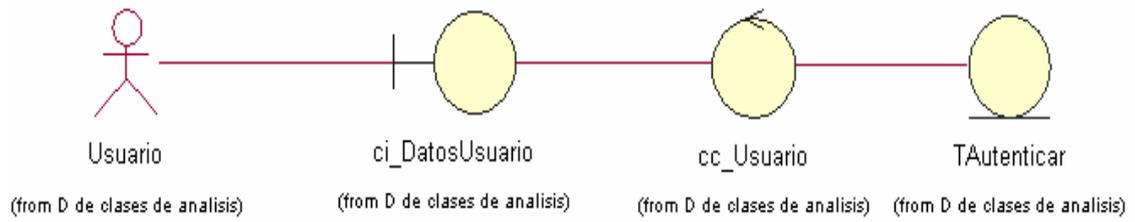


Fig. 4.1 Diagrama de Clases de Análisis CU1 Autenticar usuario.
Diagramas de Interacción (Ver Anexo A.a).

4.2.2 CU2 Solicitar matrícula.

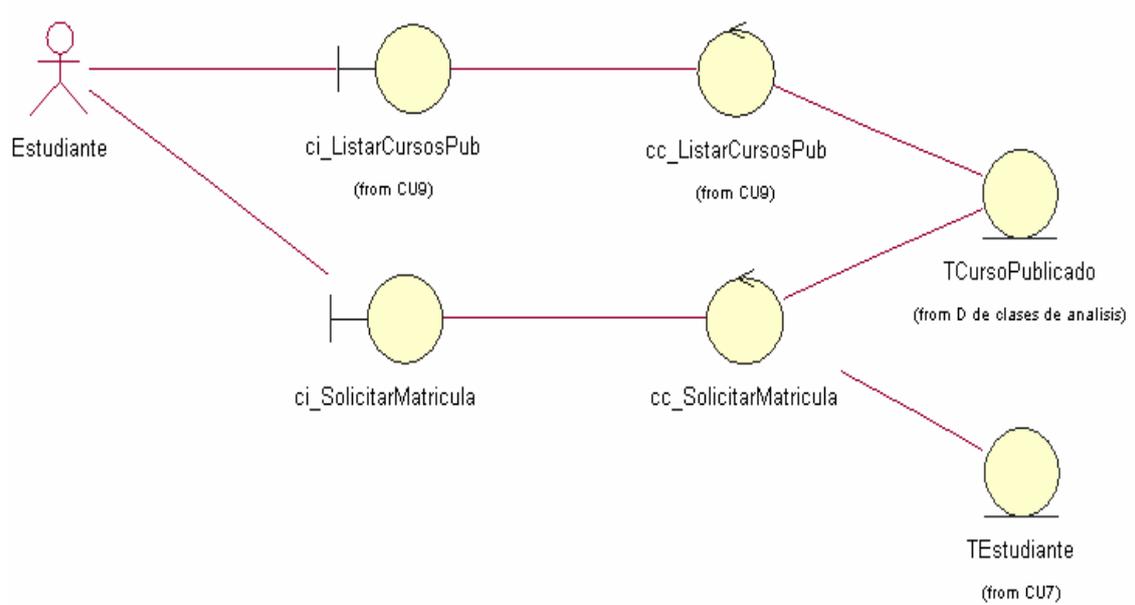


Fig. 4.2 Diagrama de Clases de Análisis CU2 Solicitar matrícula.
Diagramas de Interacción (Ver Anexo A.b).

4.2.3 CU3 Gestionar cursos propuestos.

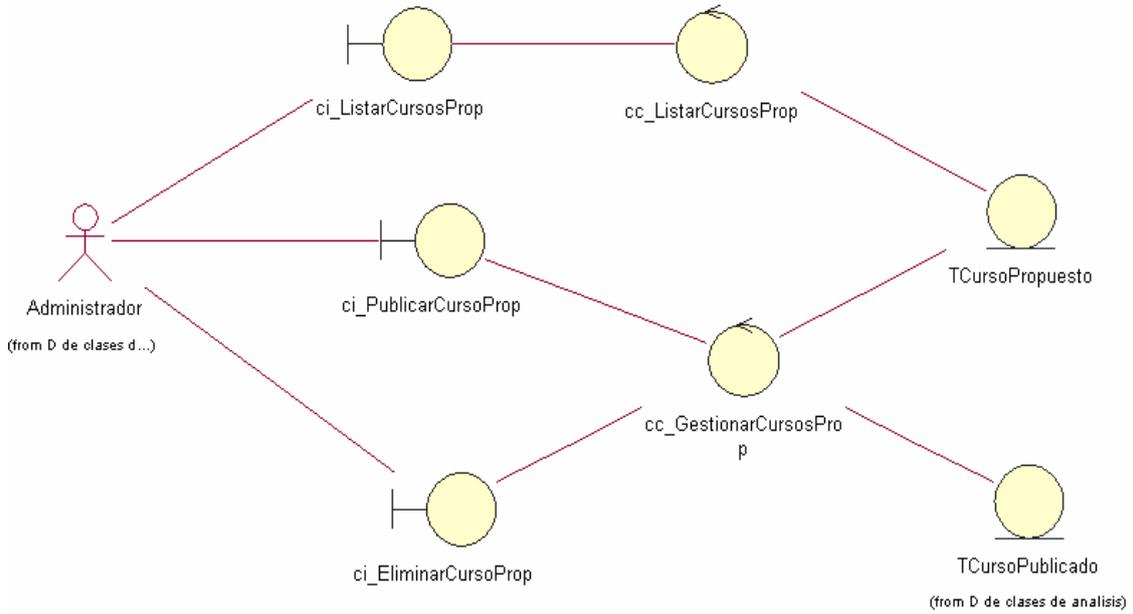


Fig. 4.3 Diagrama de Clases de Análisis CU3 Gestionar cursos propuestos.
Diagramas de Interacción (Ver Anexo A.c).

4.2.4 CU4 Gestionar cursos publicados.

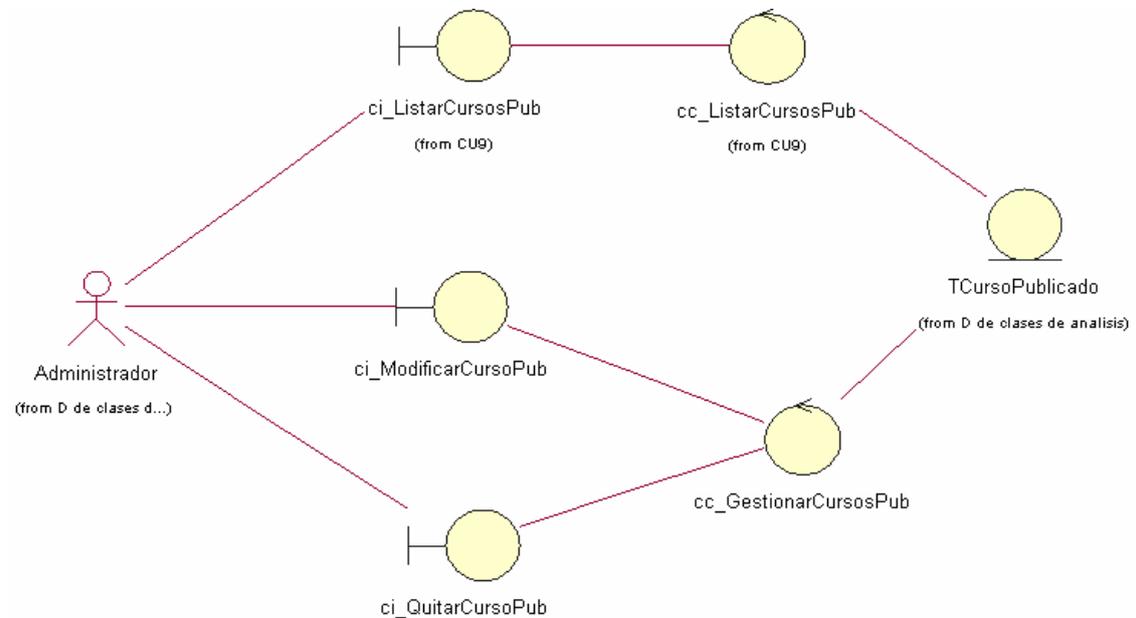


Fig. 4.4 Diagrama de Clases de Análisis CU4 Gestionar cursos publicados.
Diagramas de Interacción (Ver Anexo A.d).

4.2.5 CU5 Proponer curso.

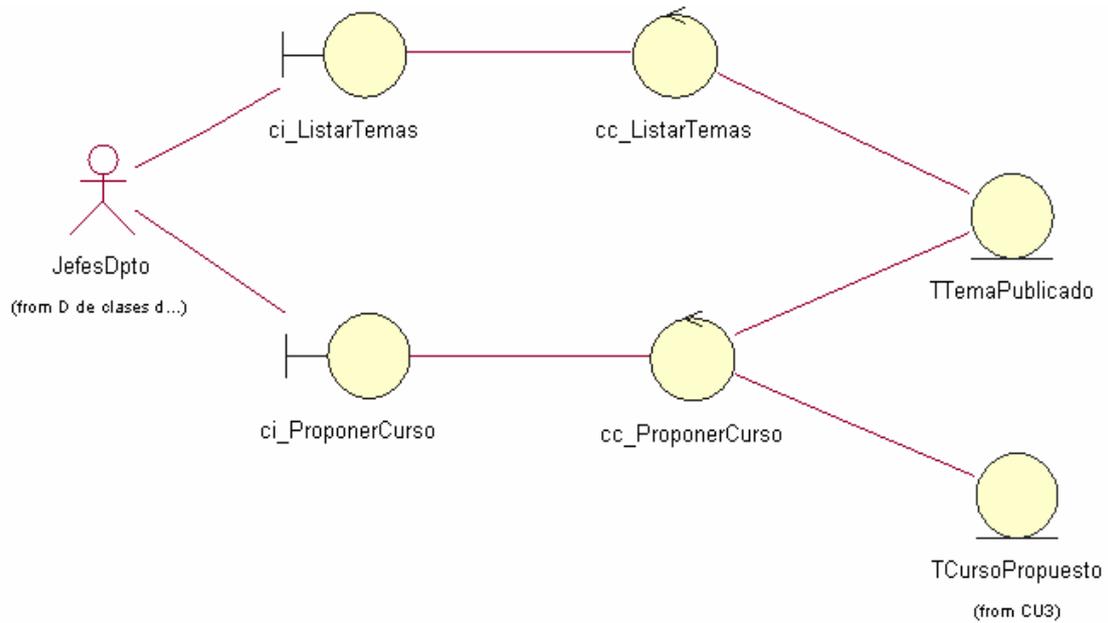


Fig. 4.5 Diagrama de Clases de Análisis CU5 Proponer curso.
Diagramas de Interacción (Ver Anexo A.e).

4.2.6 CU6 Sugerir tema.



Fig. 4.6 Diagrama de Clases de Análisis CU6 Sugerir tema.
Diagramas de Interacción (Ver Anexo A.f).

4.2.7 CU7 Realizar matrícula.

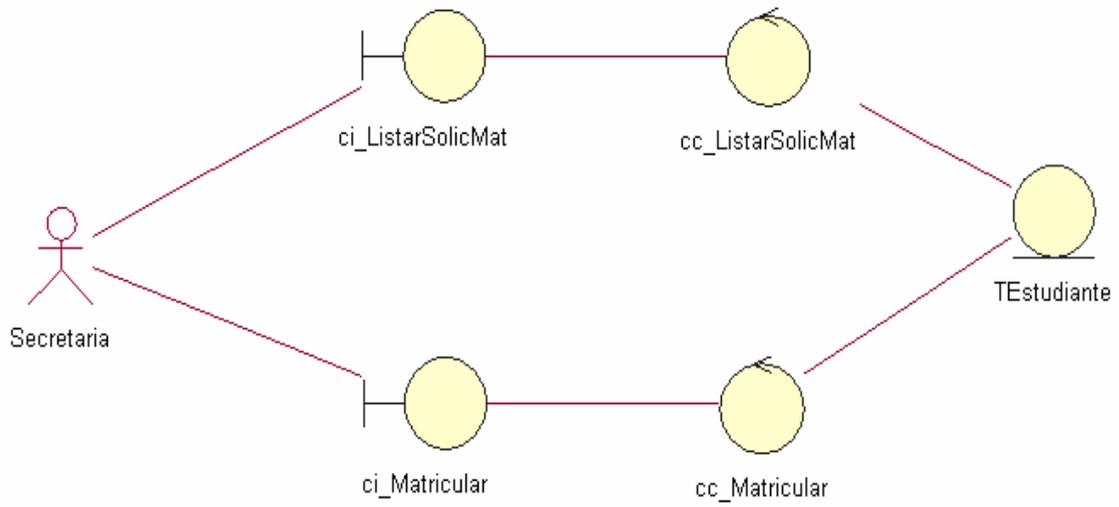


Fig. 4.7 Diagrama de Clases de Análisis CU7 Realizar matrícula.
Diagramas de Interacción (Ver Anexo A.g).

4.2.8 CU8 Gestionar tema.

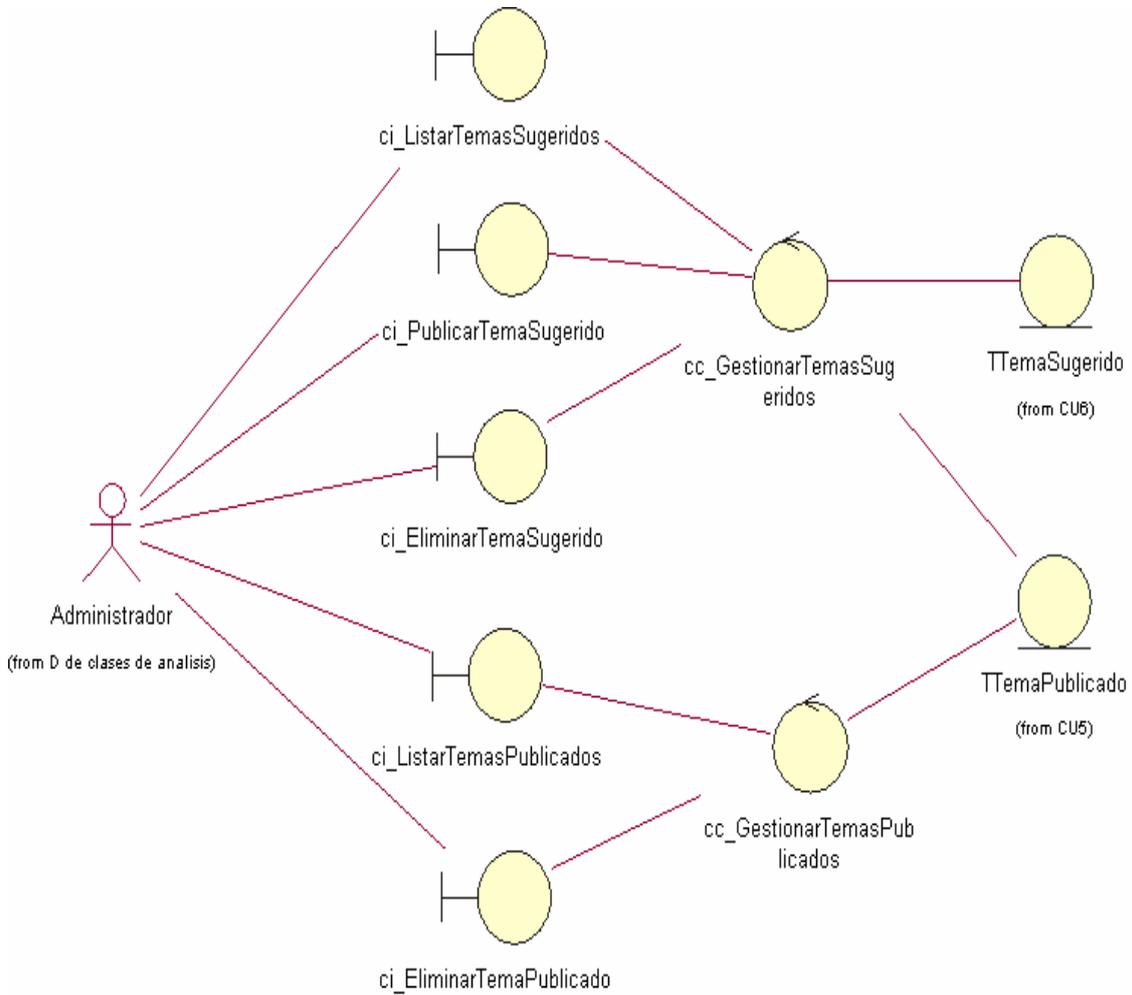


Fig. 4.8 Diagrama de Clases de Análisis CU8 Gestionar temas. Diagramas de Interacción (Ver Anexo A.h).

4.2.9 CU9 Listar cursos disponibles.

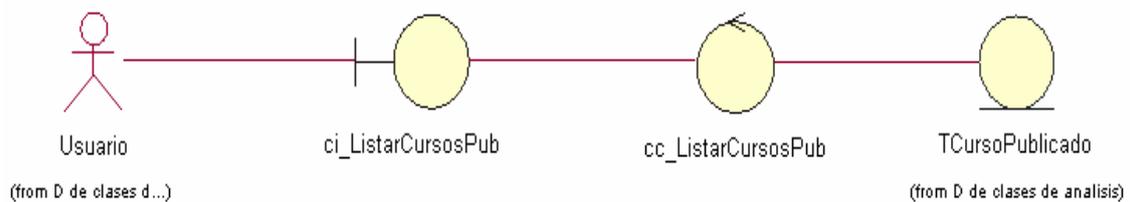


Fig. 4.9 Diagrama de Clases de Análisis CU9 Listar cursos disponibles. Diagramas de Interacción (Ver Anexo A.i).

4.2.10 CU10 Gestionar usuarios.

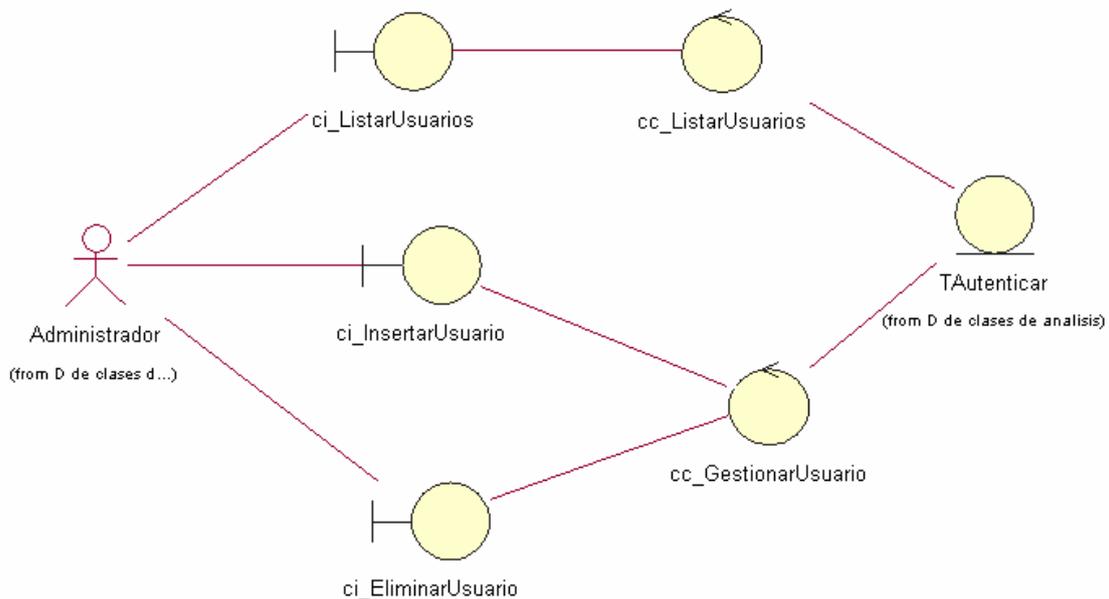


Fig. 4.10 Diagrama de Clases de Análisis CU10 Gestionar usuarios.
Diagramas de Interacción (Ver Anexo A.j).

4.3 Diagrama de clases de diseño.

Los diagramas de clase de diseño muestran gráficamente como los objetos se comunican entre ellos a fin de cumplir con los requerimientos y funcionalidades. Estos diagramas resumen la definición de las clases que se pueden implementar en el software. A continuación se relacionan los diagramas de clases de diseño de cada caso de uso del sistema propuesto (Ver Anexo B).

4.4 Principios de diseño.

El diseño en una aplicación Web es parte importante del éxito de la misma. No nos referimos con esto solamente a la apariencia estética, a la combinación de colores, o al uso de logos e imágenes de forma acertada; sino al buen diseño gráfico, una buena navegabilidad, usabilidad y distribución de los contenidos en las páginas. De todo esto depende que la información sea útil, que los servicios

se puedan usar, es decir, el diseño convierte a una aplicación Web en algo atractivo para el usuario, por su estética y su utilidad.

Con el objetivo de lograr una interfaz de usuario exitosa, en el diseño de nuestro sistema hemos tenido en cuenta los principios siguientes:

- ✓ Evitar las caídas inesperadas del sistema y los vínculos rotos.
- ✓ Garantizar la legibilidad de la aplicación, el color de los textos debe contrastar con el del fondo, y el tamaño de fuente debe ser suficientemente grande.
- ✓ Requerir un mínimo proceso de aprendizaje y permitir su utilización desde el primer momento, por cualquier persona que tenga un básico dominio del uso de ordenadores.
- ✓ Mostrar al usuario solamente aquellas opciones a las que tiene derecho a acceder.
- ✓ Requerir de los usuarios un mínimo esfuerzo para alcanzar sus objetivos.
- ✓ Limitar el número de acciones que puede realizar el usuario sobre la aplicación, mostrando sugerencias (opciones) para cada posible acción, evitando así al máximo los errores de usuario.
- ✓ Mostrar al usuario, siempre que vaya a realizar una acción relevante sobre el sistema, un mensaje de confirmación o alerta que le permita asegurarse de que es correcta o no la opción seleccionada.
- ✓ Mostrar la mayor cantidad de información acerca de las opciones brindadas de forma tal que el usuario siempre sepa cuáles son las operaciones a las que puede acceder y en qué consiste cada una de ellas.

4.5 Tratamiento de errores.

El tratamiento correcto de los errores posibilita que nuestra aplicación funcione íntegra y correctamente en todo momento. Esto lo logramos haciendo un análisis de los posibles tipos de errores con los que nos podemos encontrar, llegando a la conclusión de que puede haber errores por parte del usuario que interactúa con el sistema y también errores lógicos.

Para prevenir los errores de los usuarios a esto se les brinda en solo las funcionalidades mínimas para cada operación. Cuando se determina el tipo de usuario se le muestran solo las opciones a las que tiene acceso dado el rol que juega.

Para la validación del lado del cliente se utiliza el lenguaje JavaScript, con esto se garantiza la total integridad y consistencia de los datos suministrado por los usuarios. De esta manera se verifican los campos obligatorios y los tipos de datos, usándose menús o listas de selección así como opciones de marcado, para que los usuarios tecleen lo menos posible, evitando así muchos errores en la entrada de información al sistema. En todos los casos se les muestra mensajes de alerta y error bastante claros a los usuarios.

Otros tipos de errores que se nos pueden presentar son del lado del servidor, éstos tienen que ver con el funcionamiento interno de la aplicación. En estos casos también están previstas páginas con mensaje de error bastante específicos y entendibles para los usuarios del sistema.

4.6 Diseño de la base de datos.

4.6.1 Modelo Lógico de Datos.

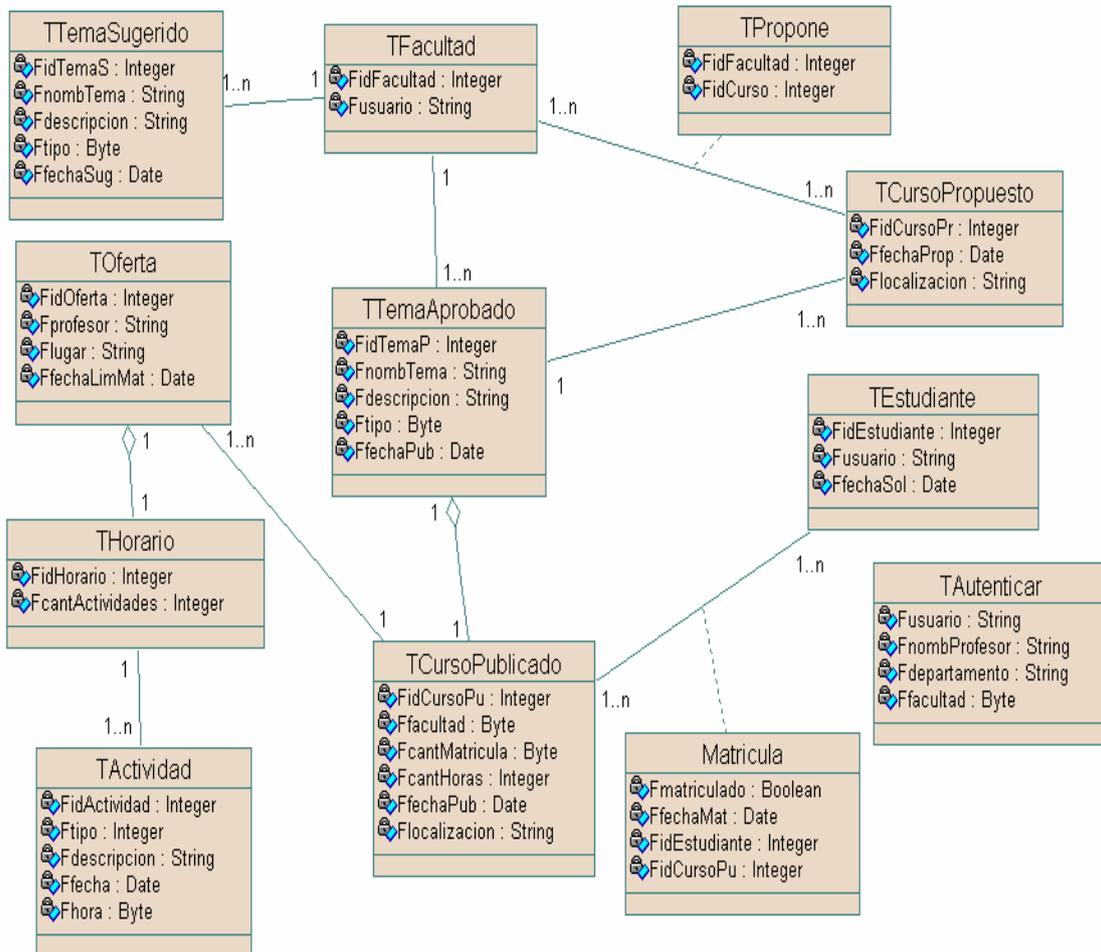


Fig. 4.11 Diagrama de Clases Persistentes.

4.6.2 Modelo Físico de Datos.

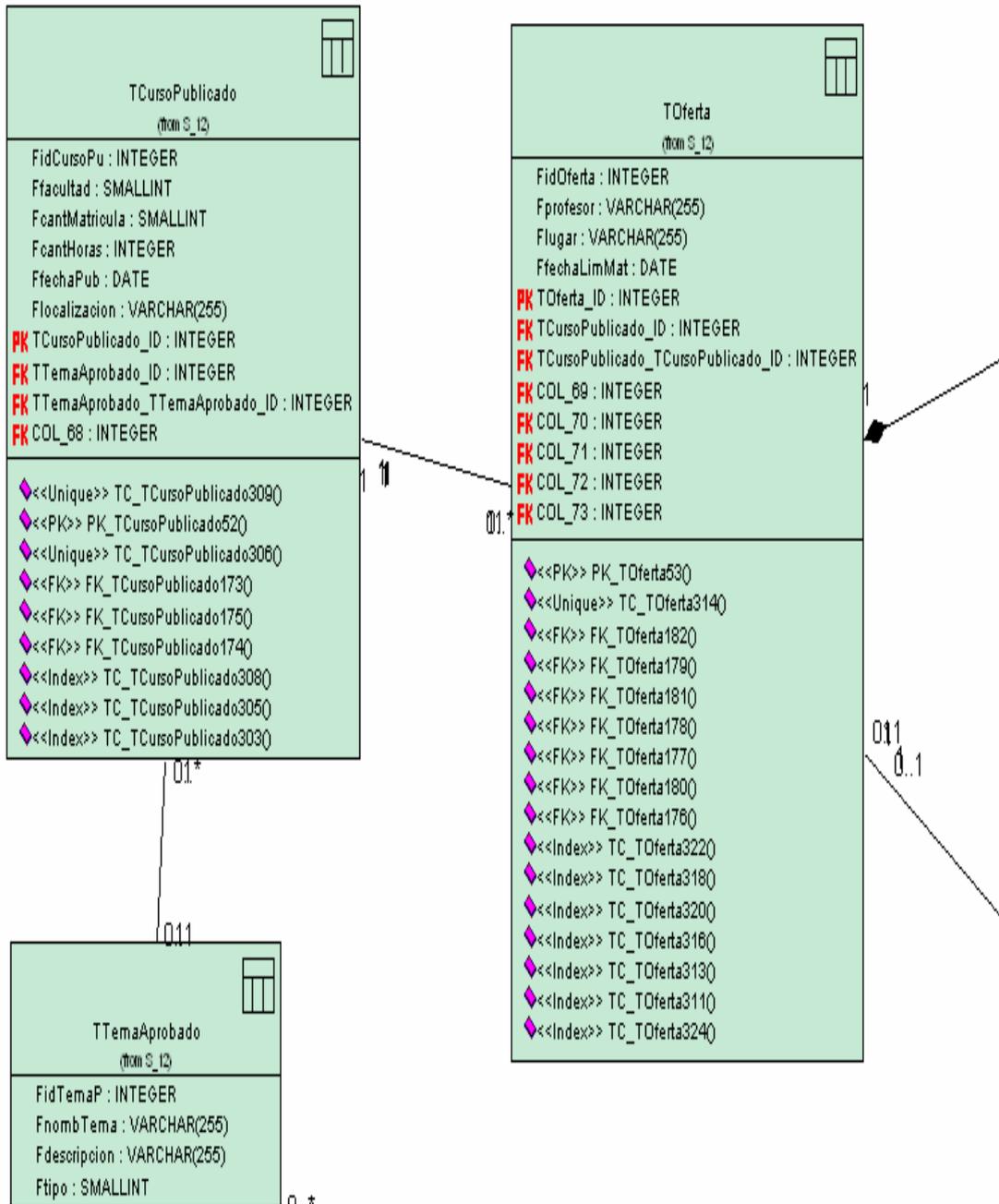


Fig. 4.12 Diagrama del Modelo Físico de Datos [parte I].

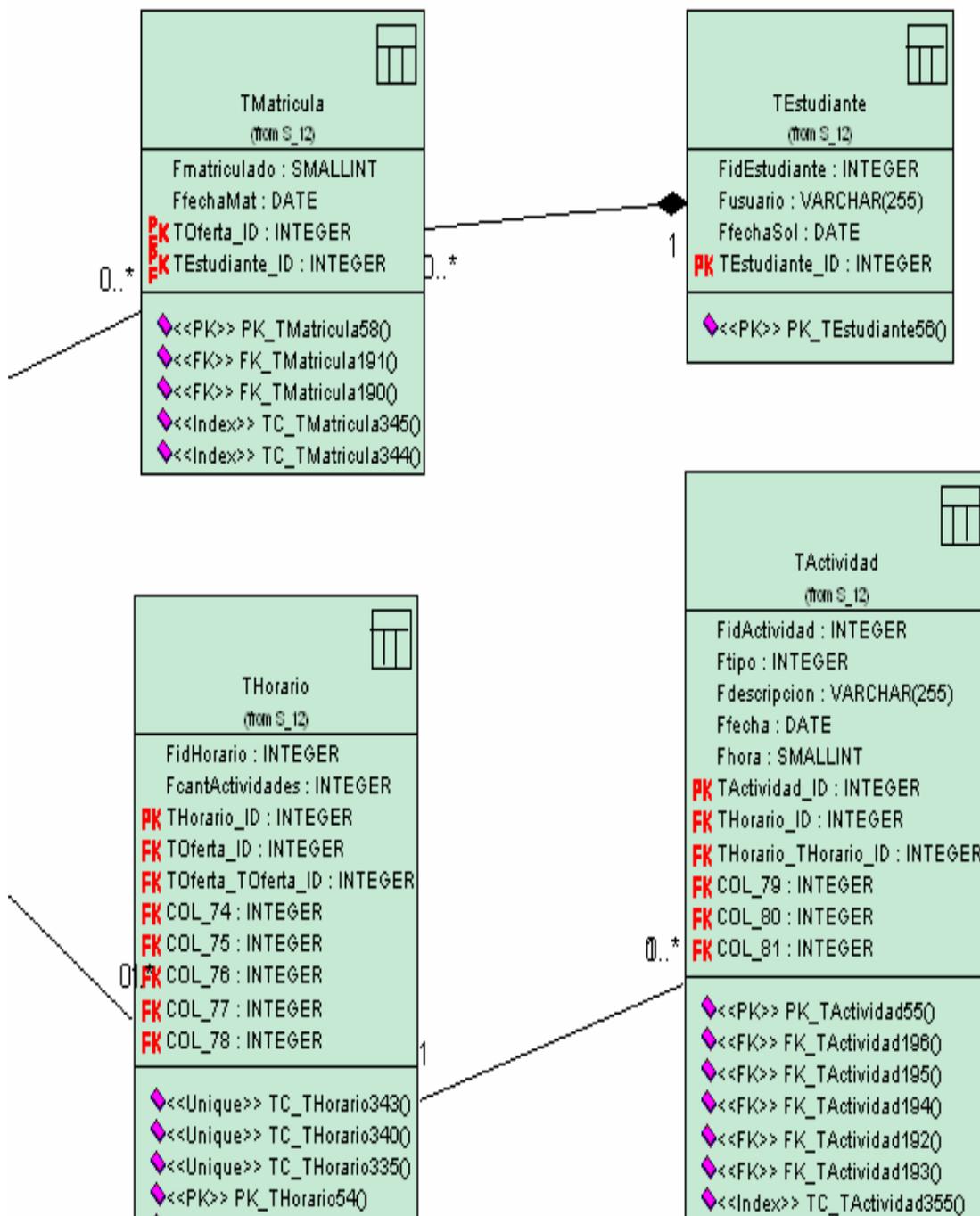


Fig. 4.13 Diagrama del Modelo Físico de Datos [parte II] (A continuación a la derecha de parte I).

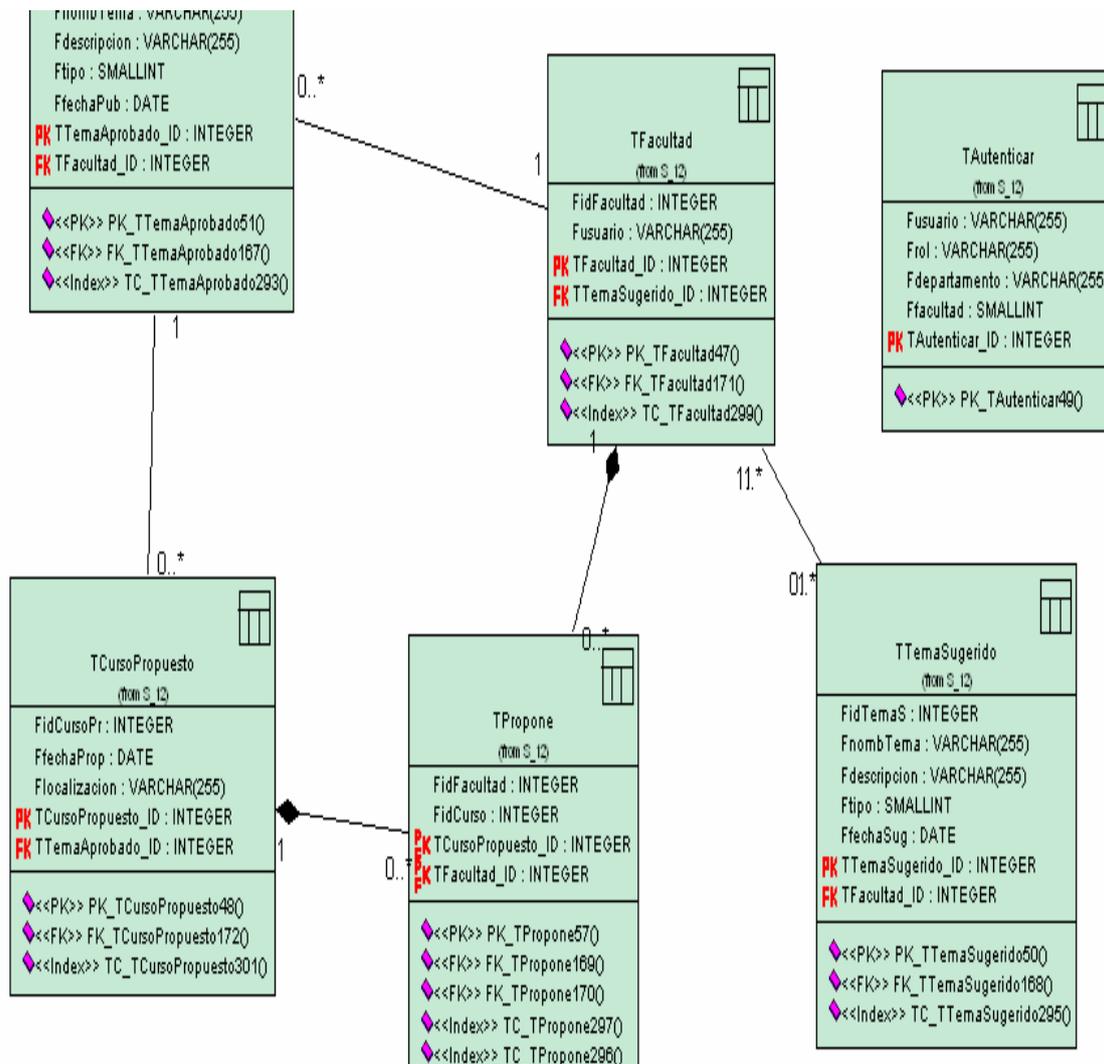


Fig. 4.14 Diagrama del Modelo Físico de Datos [parte III] (A continuación debajo de parte I y II).

4.7 Modelos de implementación.

4.7.1 Diagrama de Despliegue.

En el diagrama de despliegue se muestra de forma grafica la distribución física real del sistema de computo diseñado, a través de él se ve como están distribuidos los componentes de software entre los diferentes nodos de aplicación. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware.

Nuestra aplicación se basa en una arquitectura cliente/servidor, nuestro diagrama de despliegue esta representado por 4 nodos de cómputo. Uno de los nodos es la Estación de Trabajo con navegador, que representa a las PC clientes de los usuarios que van acceder a la aplicación publicada en el Servidor Web Apache usando el protocolo **http**. Éste se comunica mediante el protocolo **TCP/IP** con los otros dos nodos representados, el Servidor de BD MySQL en el cual corre la base de datos propia de nuestra aplicación y el Servidor de BD UCI, necesario para obtener informaciones necesarias para nuestro sistema.

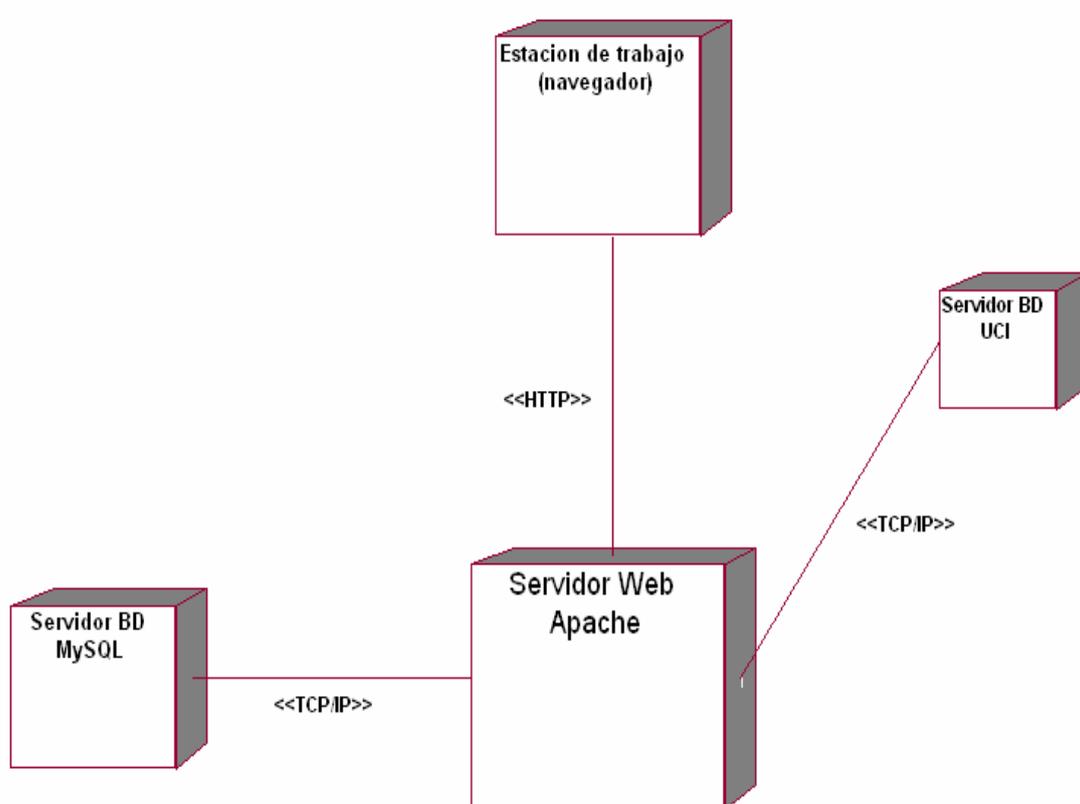


Fig. 4.15 Diagrama de Despliegue.

4.7.2 Diagrama de Componentes.

En el diagrama de componentes se ilustran un conjunto de componentes y sus relaciones. Gráficamente representan una colección de nodos o componentes y arcos, los que representan componentes e interfaces, y relaciones de dependencia respectivamente.

Se presenta el diagrama de componentes general y uno por cada módulo especificado. En el módulo de *Autenticar*, se especifica lo referentes al acceso de los usuarios, el módulo de *Funcionalidades usuarios* se refiere a todas las operaciones que realizan los usuarios y el módulo de *Administración* abarca todos los componentes relacionados con las operaciones de control del administrador.

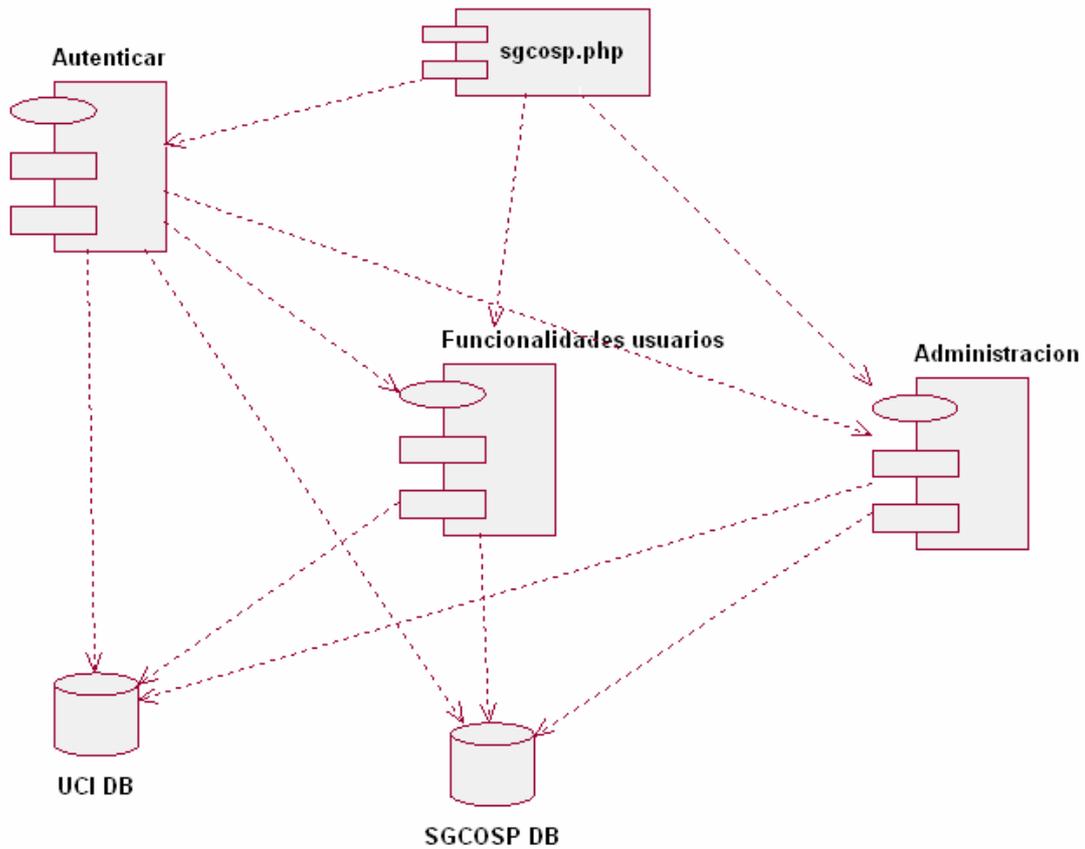


Fig. 4.16 Diagrama de Componentes principal.

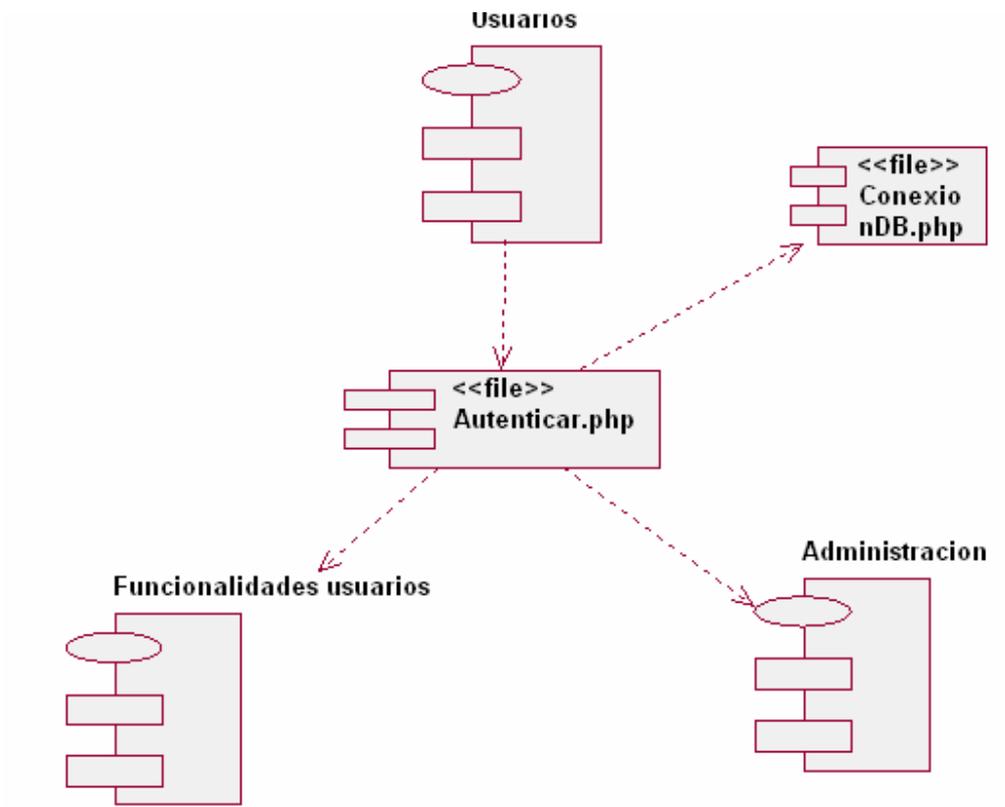


Fig. 4.17 Diagrama de Componentes del modulo Autenticar.

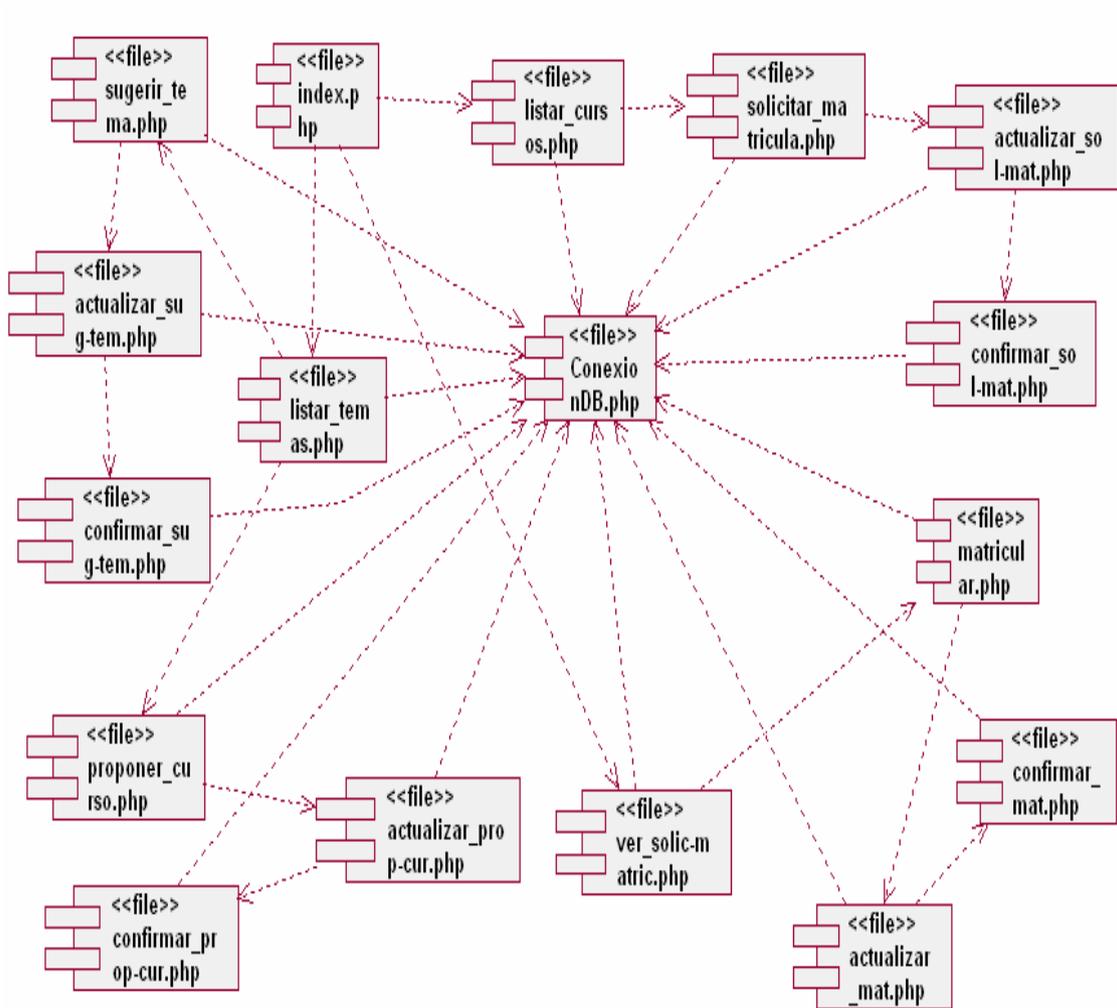


Fig. 4.18 Diagrama de Componentes del modulo Funcionalidades usuarios.

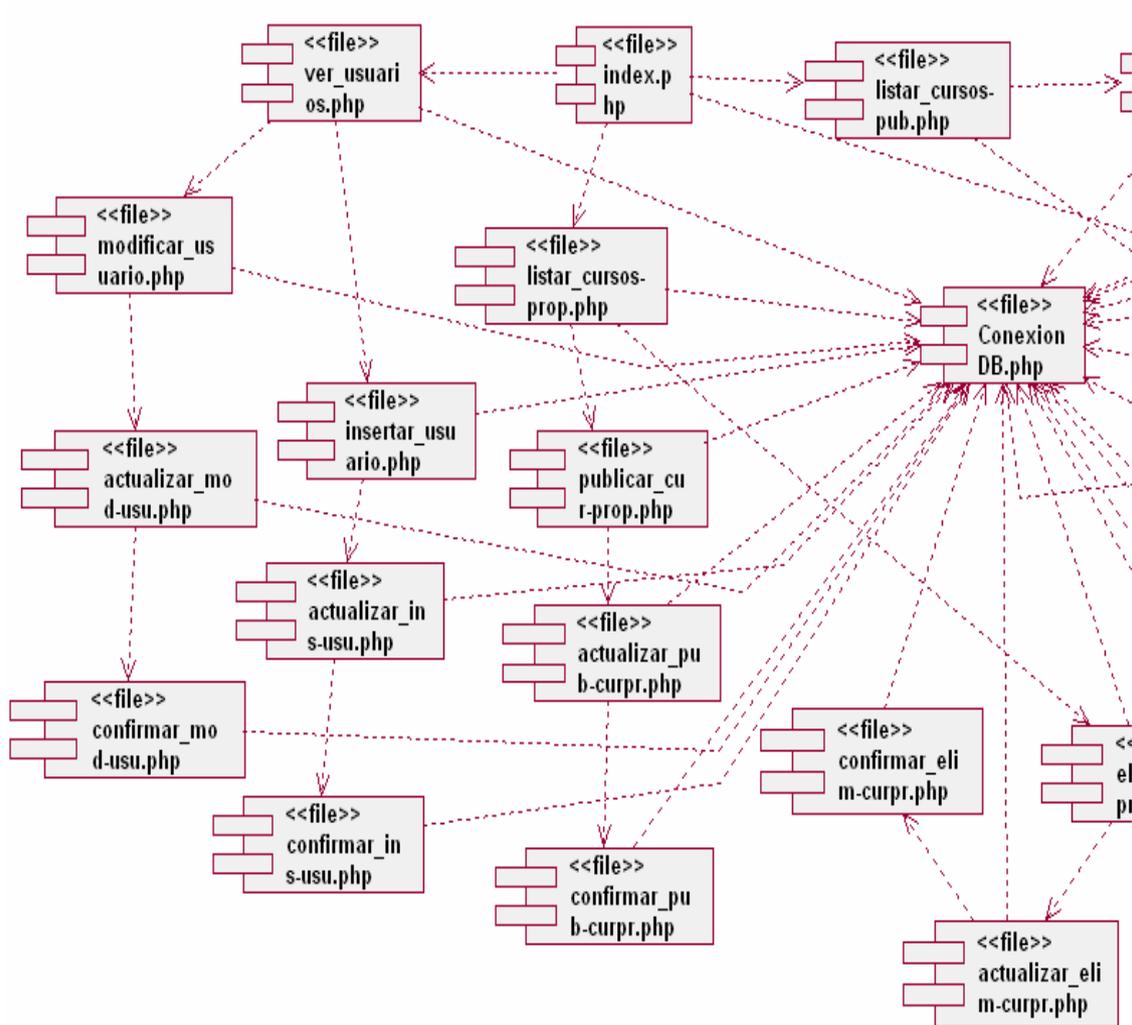


Fig. 4.19 Diagrama de Componentes del modulo Administración [parte I].

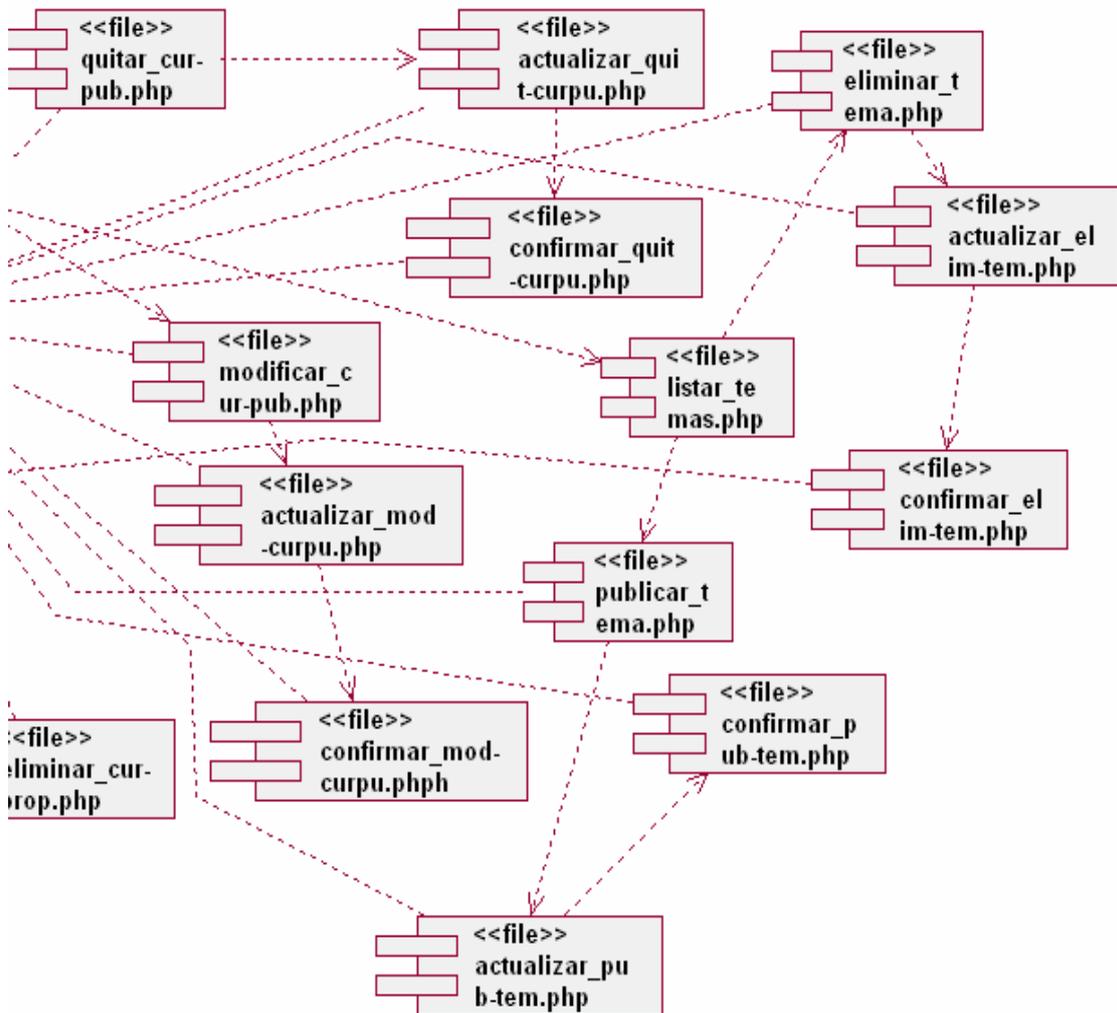


Fig. 4.20 Diagrama de Componentes del modulo Administración [parte II] (A continuación a la derecha de parte I).

4.8 Conclusiones.

En este capítulo se le ha dado cumplimiento a todo lo planteado relacionado con el análisis y diseño de la solución propuesta. Se obtuvo el diagrama de clases del análisis y del diseño de cada caso de uso del sistema, se construyó el modelo de datos necesario para gestionar toda la información necesaria en la aplicación, se expusieron los principios seguidos para el diseño de la interfaz de usuario, y finalmente se explicó cómo está estructurado físicamente el sistema mediante el diagrama de despliegue.

5.1 Introducción.

El análisis de la factibilidad de la realización o no de un proyecto es un proceso de vital importancia y del cual no se puede prescindir a la hora de realizar cualquier proyecto de software por pequeño que este sea. De aquí sale el cálculo de costos y beneficios, entre otros datos, que son muy necesarios para los planificadores del proyecto.

En este capítulo se realiza el estudio de factibilidad de nuestro sistema por planificación de puntos de función utilizando COCOMO (*Constructive Cost Model*) donde se obtienen los resultados del análisis de costos y beneficios, a partir de cálculos realizados con factores como: entradas y salidas externas, ficheros internos, interfaces externas y el lenguaje utilizado. Al final se listan los beneficios tangibles e intangibles que se obtienen.

5.2 Planificación por punto de función.

La planificación tiene como propósito establecer planes razonables para desarrollar la Ingeniería de Software y manejar los cambios de un proyecto de software. Es la actividad de estimación de los resultados del proyecto y los valores de costo, tiempo y recursos requeridos. Establece los pactos necesarios y define el plan de desarrollo.

Para la estimación del tamaño de un sistema a partir de sus requerimientos, una de las técnicas más difundidas es el Análisis de Puntos de Función. Ésta técnica permite cuantificar el tamaño de un proyecto de software en unidades independientes del lenguaje de programación, las metodologías, plataformas y/o tecnologías utilizadas.

Existen varios modelos para estimar los costos. Entre ellos uno de los más aceptados en la actualidad es COCOMO v2. Este modelo expresa el esfuerzo de desarrollo en términos de personas/mes.

En el caso del sistema que se ha diseñado (**SiGesCOSP**) no se hace necesario calcular el costo monetario ya que es un sistema a desarrollar en la Universidad de Ciencias Informáticas (UCI) y se cuenta con el capital humano necesario para su desarrollo (compuesto por alumnos y profesores de la Universidad), además de la totalidad de los recursos de softwares y hardware que se requieren.

5.2.1 Planificación.

Tabla 5.1 Entradas externas.

Nombre de la entrada externa	Cantidad de Ficheros	Cantidad de Elementos de datos	Clasificación (Bajo, Medio y Alto)
Proponer curso	3	4	Bajo
Publicar curso propuesto	5	10	Alto
Eliminar curso propuesto	3	5	Medio
Modificar cursos publicados	4	8	Alto
Eliminar cursos publicados	6	18	Alto
Enviar tema	2	4	Bajo
Eliminar tema	2	4	Bajo
Publicar tema	3	6	Medio
Solicitar matrícula	4	5	Alto
Matricular estudiante	4	4	Medio
Insertar usuario	1	4	Bajo
Eliminar usuario	1	4	Bajo

Tabla 5.2 Salidas externas.

Nombre de la salida externa	Cantidad de Ficheros	Cantidad de Elementos de datos	Clasificación (Bajo, Medio y Alto)
Listar cursos propuestos	2	3	Bajo
Listar cursos publicados	4	13	Alto
Listar temas sugeridos	2	4	Bajo
Listar temas publicados	1	7	Bajo
Listar solicitudes de matrícula	4	6	Alto
Listar usuarios	1	4	Bajo

Tabla 5.3 Peticiones.

Nombre de la petición	Cantidad de Ficheros	Cantidad de Elementos de datos	Clasificación (Bajo, Medio y Alto)
Buscar usuario, departamento y facultad	1	3	Bajo
Buscar cursos propuestos	2	5	Bajo
Buscar cursos publicados	1	5	Bajo
Buscar temas sugeridos	1	5	Bajo
Buscar temas publicados	1	7	Bajo
Buscar solicitudes de matrícula	4	6	Alto

Tabla 5.4 Ficheros internos.

Nombre del fichero interno	Cantidad de record	Cantidad de Elementos de datos	Clasificación (Bajo, Medio y Alto)
Autenticar	1	4	Bajo
Facultad	1	2	Bajo
Curso_Propuesto	1	4	Bajo
Propone	1	2	Bajo
Tema_Sugerido	1	6	Bajo
Tema_Publicado	1	8	Bajo
Curso_Publicado	1	7	Bajo
Oferta	1	6	Bajo
Horario	1	3	Bajo
Actividad	1	5	Bajo
Estudiante	1	3	Bajo
Matrícula	1	4	Bajo

Tabla 5.5 Interfaces externas.

Elementos	Bajos		Medios		Altos		Subtotal de puntos de función
	No	X Peso	No	X Peso	No	X Peso	
Ficheros internos	12	7	0	10	0	15	84
Entradas externas	5	3	3	4	4	6	51
Salidas externas	4	4	0	5	2	7	30
Peticiones	5	3	0	4	1	6	21
Total							186

5.2.2 Costos.

Tabla 5.6 Instrucciones fuentes estimadas.

Características	Valor
Puntos de función desajustados	186
Lenguaje	PHP (80 %) MySQL (20%)
Instrucciones fuentes por puntos de función	15 39
Instrucciones fuentes por lenguaje (miles de instrucciones)	(2,232) (1,451)
Instrucciones fuentes (miles de instrucciones)	3,683

Tabla 5.7 Factores de escala.

Factores	Valor	Justificación
PREC	2.48	El equipo de desarrollo esta familiarizado con el desarrollo de este tipo de aplicaciones. Se han desarrollado algunos proyectos similares con anterioridad.
FLEX	2.03	La aplicación debe sustentar todos los requerimientos de una forma integral y adecuarse a los criterios y especificaciones de la interfaz externa.
TEAM	4,38	Existen problemas cooperativos entre los integrantes del equipo de desarrollador del software.
RESL	2.83	La herramienta de desarrollo que se va a utilizar posee gran tolerancia ante fallos y capacidad de resolución de riesgos que pueden ocurrir.
PMAT	6.24	El proceso tiene un bajo nivel 1 de madurez,

		considerado bajo.
--	--	-------------------

Tabla 5.8 Multiplicadores de esfuerzo.

Nombre	Valor	Justificación
RUSE	1.00	El desarrollo del proyecto debe permitir la reutilización del código para los demás módulos del Sistema de Gestión de Cursos Optativos y de Segundo Perfil. Existe concordancia entre la documentación y las necesidades del ciclo de vida.
PDIF	1.00	Las restricciones de almacenamiento y tiempo en el desarrollo presentan una dificultad mediana. La plataforma escogida es de estable volatilidad.
PREX	1.33	El conocimiento de los especialistas de las herramientas para el desarrollo de la aplicación es baja, ya que ha sido muy poca la experiencia adquirida previamente.
FCIL	0.87	Se realiza un uso notable de herramientas case para documentar e implementar el sistema.
SCED	1.00	Planificación normal.
RCPX	1.00	La complejidad del software es normal, la documentación con un buen nivel de confiabilidad. La base de datos es de tamaño mediano. Alta confianza requerida.
PERS	1.00	Los desarrolladores tienen una alta capacidad para el análisis y la implementación del sistema.

La ecuación que nos brinda COCOMO para determinar el esfuerzo de desarrollo es la siguiente:

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

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

$$E = 0,91 + 0,01 * 17.96 = 1.0896$$

$$PM = 2,94 * 3,683^{1.0896} * 1.1571 = 14.0815 \approx 14 \text{ hombres/mes.}$$

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

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

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

$$F = 0,28 + 0,2 * (1.0896 - 0,91) = 0,31592$$

$$TDEV = 3,67 * 14^{0,31592} = 8,44 \approx 8 \text{ meses}$$

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

$$CH = PM / TDEV$$

$$CH = 14/8 = 1,75 \approx 2 \text{ personas}$$

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

$$TDEV = 14 \text{ meses.}$$

$$CH = 1$$

5.3 Beneficios tangibles e intangibles.

El **SiGesCOSP** es un software que no se diseñó con el objetivo de comercializarse, surge como parte de una necesidad existente en la Universidad de la Ciencias Informáticas, por lo que los beneficios obtenidos son en su totalidad intangibles, ya que permite automatizar una serie de procesos que se realizan de forma manual y con nuestra solución se le da una posibilidad muy amplia a todos los usuarios de la UCI para acceder eficazmente y de forma flexible a la información relacionada con el tema.

A pesar de lo antes mencionado se considera que nuestro sistema puede ser utilizado en otros ámbitos externos, ya que en su diseño se tuvo en cuenta la posible realización de nuevas versiones y la adaptación a cambios sin la necesidad de realizar grandes transformaciones, aunque su fin es responder a necesidades actuales de la UCI. Los principales beneficios que aporta son:

- ✓ Al estar la información en una base de datos se puede controlar el acceso a la misma lo posibilita la confidencialidad, organización, centralización en el procesamiento de la misma.
- ✓ La información es accesible desde cualquier punto de la red de la UCI, garantizando con esto un alto grado de actualización en cuanto al estado y contenido.
- ✓ Se mejora la calidad de los procesos ya que se logra una gran flexibilidad tanto para estudiantes, como para docentes y directivos.

5.4 Análisis de costo y beneficios.

Para llevar a cabo el desarrollo de un producto de software se debe tener en cuenta la relación entre los beneficios que va a reportar y el costo que se incurre en su producción, para determinar si el proyecto es viable o no. Los beneficios aportados pueden ser de tipo económicos, cuando se trata de ingresos en dinero propiamente dicho; o de orden social, cuando la aplicación va dirigida a resolver una necesidad existente. Estos beneficios son tan importantes como los comerciales.

El **SiGesCOSP** que se propone está dirigido a resolver un problema existente en la universidad. Su utilización propiciará una gran flexibilidad y agilidad a docentes y estudiantes que tendrán respuesta a la mayoría de sus necesidades relacionadas con el tema de la gestión de los cursos optativos y de segundo perfil. No requiere inversión en software, ya que las herramientas que se utilizan en su diseño, y se emplearán en su desarrollo pertenecen al grupo de los softwares libres y el grupo de desarrollo está compuesto por estudiantes de la Universidad de la Ciencias Informáticas (UCI) y del Instituto Superior Politécnico “José Antonio Echeverría” (ISPJAE).

El **SiGesCOSP** es factible puesto que el costo que tendrá será mucho menor que el costo que implicaría adquirir un sistema en el mercado de software. Proporciona beneficios importantes y significativos (de orden social) como ahorro del material de oficina y esfuerzo humano en el desempeño de los procesos que se automatizaron. Además garantiza la confidencialidad de la información y el acceso centralizado a la misma.

5.5 Conclusiones.

En este capítulo se realizó el análisis de factibilidad de nuestro sistema, incluyendo los beneficios que aporta y se concluyó que es factible llevar a cabo el desarrollo de la aplicación ya que repercutirá de una forma muy positiva en la organización de los procesos en la universidad. También demuestra las posibilidades que brinda la utilización de herramientas informáticas en la gestión de procesos.

CONCLUSIONES

El desarrollo y futura implantación del proyecto que se ha diseñado, **SiGesCOSP**, permitirá darle solución a gran parte de los problemas relacionados con la gestión de la información de los cursos optativos y de segundo perfil en la universidad, quedando algunos que se podrán pulir en futuras versiones del producto. Se constará con un sistema flexible que brindará a los usuarios gran comodidad a la hora de realizar las tareas y éstas se realizarán además con gran rapidez y confiabilidad.

Entre los principales resultados obtenidos tenemos los siguientes:

- Se cumplen los objetivos propuestos al inicio de este trabajo.
- La solución que se propuso utilizando el Modelo del Negocio ha sido acertada, los requerimientos soportan al sistema y los casos de uso satisfacen las necesidades funcionales.
- Se siguen los principios básicos de diseño descritos para el desarrollo del sistema.
- Se logra una seguridad y protección de la información manipulada por el sistema consecuente con el nivel de seguridad y confiabilidad requerido.

Los estudios realizados sobre las tecnologías existentes nos hicieron inclinarnos a proponer la combinación **PHP-MySQL** en la implementación del sistema ya que nos brinda grandes ventajas como que puede correr sobre cualquier plataforma y su uso es libre sin necesidad de licencias. También nos inclinamos por la metodología **RUP**, la cual fue decisiva a la hora de diseñar la aplicación.

Por último después de realizar los estudios de factibilidad correspondiente se obtuvo que es factible la implementación y puesta en ejecución de la aplicación dados los requerimientos de la organización y los beneficios intangibles que se obtienen para la Universidad de las Ciencias Informáticas.

RECOMENDACIONES

Entre las recomendaciones más importantes que se han considerado están las siguientes:

- Llevar a cabo la implementación de la totalidad de las funcionalidades diseñadas que dan cumplimiento a los objetivos trazados.
- Incorporarle a las siguientes versiones el módulo de profesores con todas las funcionalidades requeridas.
- Poner en funcionamiento el sistema como parte de conjunto de servicios de la intranet de la UCI.

REFERENCIAS BIBLIOGRAFICAS

- [1] Anónimo, *¿Qué es el Software Libre?*
<http://www.vialibre.org.ar/index.php/article/articleview/2/1/12/> (10/03/2006).
- [2] Anónimo, *Cliente-Servidor* <http://es.wikipedia.org/wiki/Cliente-servidor>
(22/02/2006).
- [3] Anónimo, *¿Qué es ASP.NET?*
<http://es.gotdotnet.com/quickstart/aspplus/doc/whatisaspx.aspx>
(03/04/2006).
- [4] Anónimo, *.NET Framework*
<http://www.desarrolloweb.com/articulos/1328.php> (03/04/2006).
- [5] Anónimo, *PHP* <http://es.wikipedia.org/wiki/Php> (05/04/2006).
- [6] Christian Van Der Henst S., *¿Qué es el PHP?*
<http://www.maestrosdelweb.com/editorial/phpintro/> (05/04/2006).
- [7] Anónimo, *Curso de programación en PHP*
<http://www.adrformacion.com/cursos/php/leccion1/tutorial1.html>
(05/04/2006).
- [8] Anónimo, *Comparación entre sistemas gestores de base de datos (SGBD)*
<http://www.monografias.com/trabajos29/comparacion-sistemás/comparacion-sistemás.shtml> (15/05/2006).
- [9] Anónimo, *Comparación entre sistemas gestores de base de datos (SGBD)*
<http://www.monografias.com/trabajos29/comparacion-sistemás/comparacion-sistemás2.shtml> (15/05/2006).
- [10] Alberto Molpecere, *Procesos de desarrollo: RUP, XP y FDD*
<http://www.javahispano.org/articles.article.action?id=76> (13/02/2006).
- [11] Anónimo, *Ingeniería de Software UML*
<http://www.monografias.com/trabajos5/insof/insof.shtml> (07/06/2006).

BIBLIOGRAFIA

- ✚ <http://www.ciberaula.com>
- ✚ <http://www.uib.es/depart/gte/edutec-e/Revelec12/deBenito.html>
- ✚ <http://www.uoc.edu/mosaic/articulos/cms1204.html>
- ✚ <http://www.bitacoras.com/recursos/cms.php>
- ✚ <http://www.webaldia.net/productos/01b45a92a00f3d401.html>
- ✚ <http://www.maestrosdelweb.com/editorial/bitrix/>
- ✚ <http://tecnologias.gio.etsit.upm.es/elearning/lcms--learning-content-management-system--sistema-de-gestion-de-contenidos-de-aprendizaje-26.asp>
- ✚ http://www.webtaller.com/maletin/articulos/como_evaluar_un_sistema_de_gestion_de_contenidos.php
- ✚ <http://www.tierradenomadas.com/tw008.phtml>
- ✚ <http://www.cnc.una.py/desoft/sgrh.html>
- ✚ <http://moodle.org/mod/resource/view.php?id=2475>
- ✚ <http://www.directindustry.com.mx/>
- ✚ <http://www.cnc.una.py/desoft/sqs.html>
- ✚ <http://dis.um.es/~jmolina/EspecificacionPráctica2004.doc>
- ✚ http://elearning.fidm.edu/help/es/courseadmin/ch_courses_overview.html
- ✚ http://elearning.fidm.edu/help/es/courseadmin/ch_home_general_info.html
- ✚ http://elearning.fidm.edu/help/es/courseadmin/ch_home_my_courses.html
- ✚ <http://www.tecno-soluciones.com/aprendizaje.htm>
- ✚ <http://msdn.microsoft.com/asp.net/info/overview/default.aspx>
- ✚ <http://www.desarrolloweb.com/articulos/1328.php>
- ✚ <http://www.desarrolloweb.com/articulos/1329.php?manual=48>
- ✚ <http://www.monografias.com/trabajos28/agilidad-desarrollo-software-libre/agilidad-desarrollo-software-libre.shtml>
- ✚ <http://www.monografias.com/trabajos29/comparacion-sistemás/comparacion-sistemás.shtml>

- ✚ <http://www.monografias.com/trabajos24/seleccion-plataforma/seleccion-plataforma.shtml>
- ✚ <http://www.monografias.com/trabajos28/proyecto-uml/proyecto-uml.shtml>
- ✚ <http://www.microsoft.com/spanish/msdn/articulos/archivo/160201/voices/asp11122000.asp>
- ✚ http://www.cgrsoftware.com/colab_ver.htm?bmoral/1.htm
- ✚ <http://es.gotdotnet.com/quickstart/aspplus/doc/whatis.aspx.aspx>
- ✚ <http://es.wikipedia.org/wiki/Php>
- ✚ <http://www.maestrosdelweb.com/editorial/phpintro/>
- ✚ <http://www.adrformacion.com/cursos/php/leccion1/tutorial1.html>
- ✚ <http://www.monografias.com/trabajos29/comparacion-sistemás/comparacion-sistemás2.shtml>
- ✚ <http://www.dsic.upv.es/asignaturas/facultad/lisi/ejemplorup/>
- ✚ <http://es.wikipedia.org/wiki/RUP>
- ✚ <http://www.willydev.net/descargas/articulos/general/cualxpfddrup.PDF>
- ✚ <http://www.javahispano.org/articles.article.action?id=76>
- ✚ <http://asp.com-e.net/codigo/aspnet.asp>
- ✚ <http://www.bedelias.edu.uy/index.html>
- ✚ http://www.bureaudeprensa.com/es/view.php?bn=bureaudeprensa_inte&key=1077564309&pattern=Grupo
- ✚ http://www.techne.com.br/espanhol/Institucional/noticias/not_fort.htm
- ✚ http://czsa.enlaces.cl/ERED/docs_pie/5_pieclaret.doc
- ✚ http://www.edu.aytolacoruna.es/servicios/ayuda_y_manuales/como_puedo_crear_una_cuenta_privada_de_usuario_para_el_aula_como_se_usa_el_sistema_de_gestion_de_usuarios
- ✚ http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0034-75311997000200011&lng=es&nrm=iso&tlng=es
- ✚ <http://www.educaweb.com/EducaNews/interface/asp/web/NoticiasMostrar.asp?NoticialID=1034&SeccionID=1484>
- ✚ <http://www.elquintero.net/Manuales.aspx?jscript=true>
- ✚ http://es.wikipedia.org/wiki/Software_libre

-  <http://www.sindominio.net/biblioweb/telematica/softlibre/>
-  <http://laventana.casa.cult.cu/modules.php?name=News&file=article&sid=272>
-  <http://www.vialibre.org.ar/index.php/article/articleview/2/1/12/>
-  <http://www.unesco.org.ve/general/software8.htm>
-  <http://www.programacion.com/php/articulo/porquephp/>
-  <http://www.monografias.com/trabajos5/insof/insof.shtml>
-  <http://es.wikipedia.org/wiki/Cliente-servidor>
-  <http://www.monografias.com/trabajos5/insof/insof.shtml>
-  http://www.ecologia.edu.mx/posgrado/cursos_optativos.htm
-  http://licbq.fcien.edu.uy/menu/submenu_planes/optativas.html
-  <http://cursos.puc.cl/catalogo/sitio/aap.html>
-  http://mamcyp.unam.edu.ar/index.php?option=com_content&task=view&id=12&Itemid=15
-  http://www.quimica.izt.uam.mx/posgrado-quimica/cursos_optativos.htm

GLOSARIO DE TÉRMINOS

UCI: Universidad de las Ciencias Informáticas

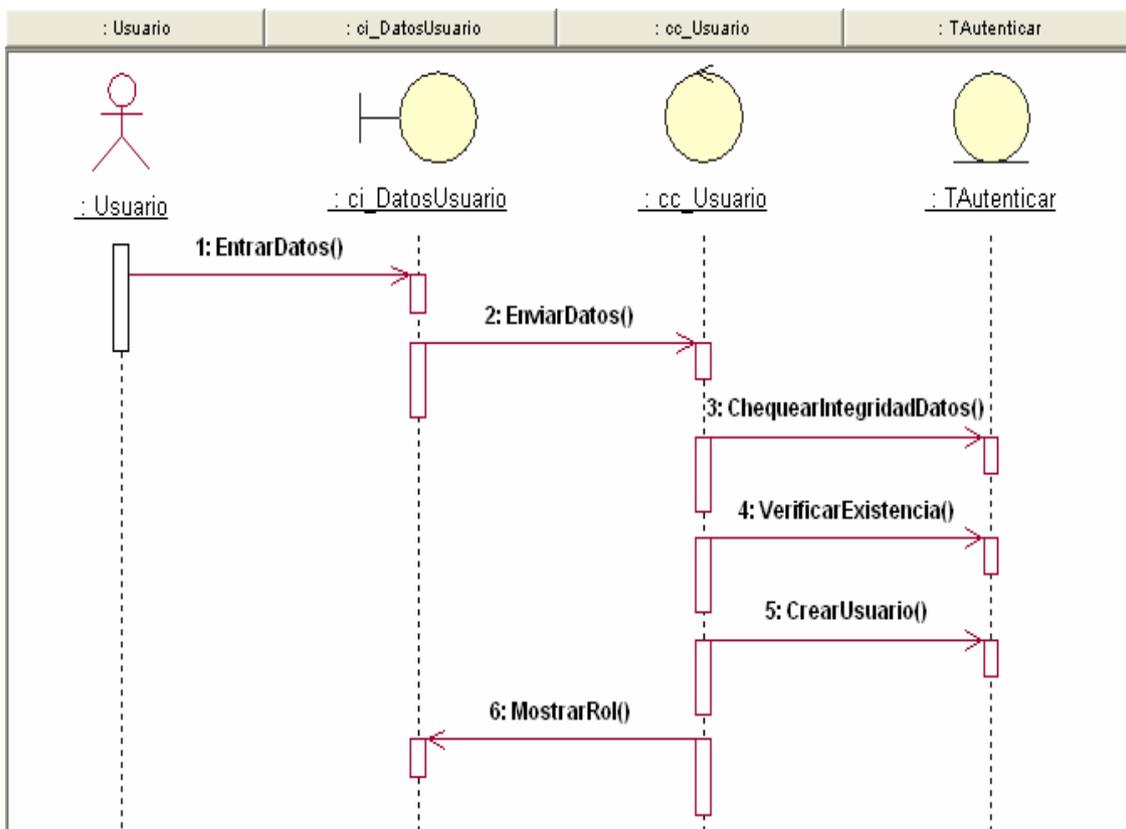
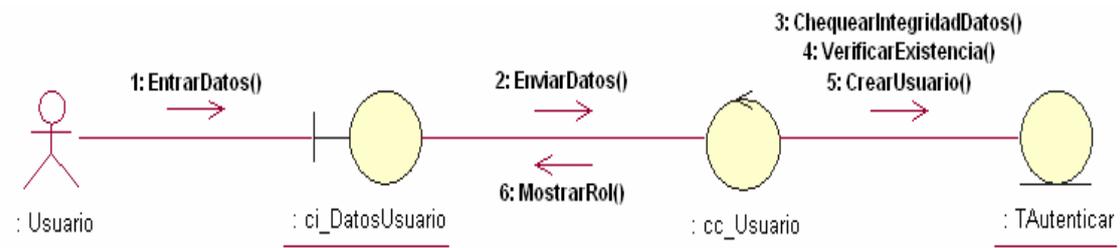
Programa analítico: Se constituyen en el documento que soporta un curso y que contiene informaciones tales como: orientaciones metodológicas, secuencia de actividades, sistema de evaluación, planificación del horario, etc.

HTTP: Protocolo de transferencia de hipertextos.

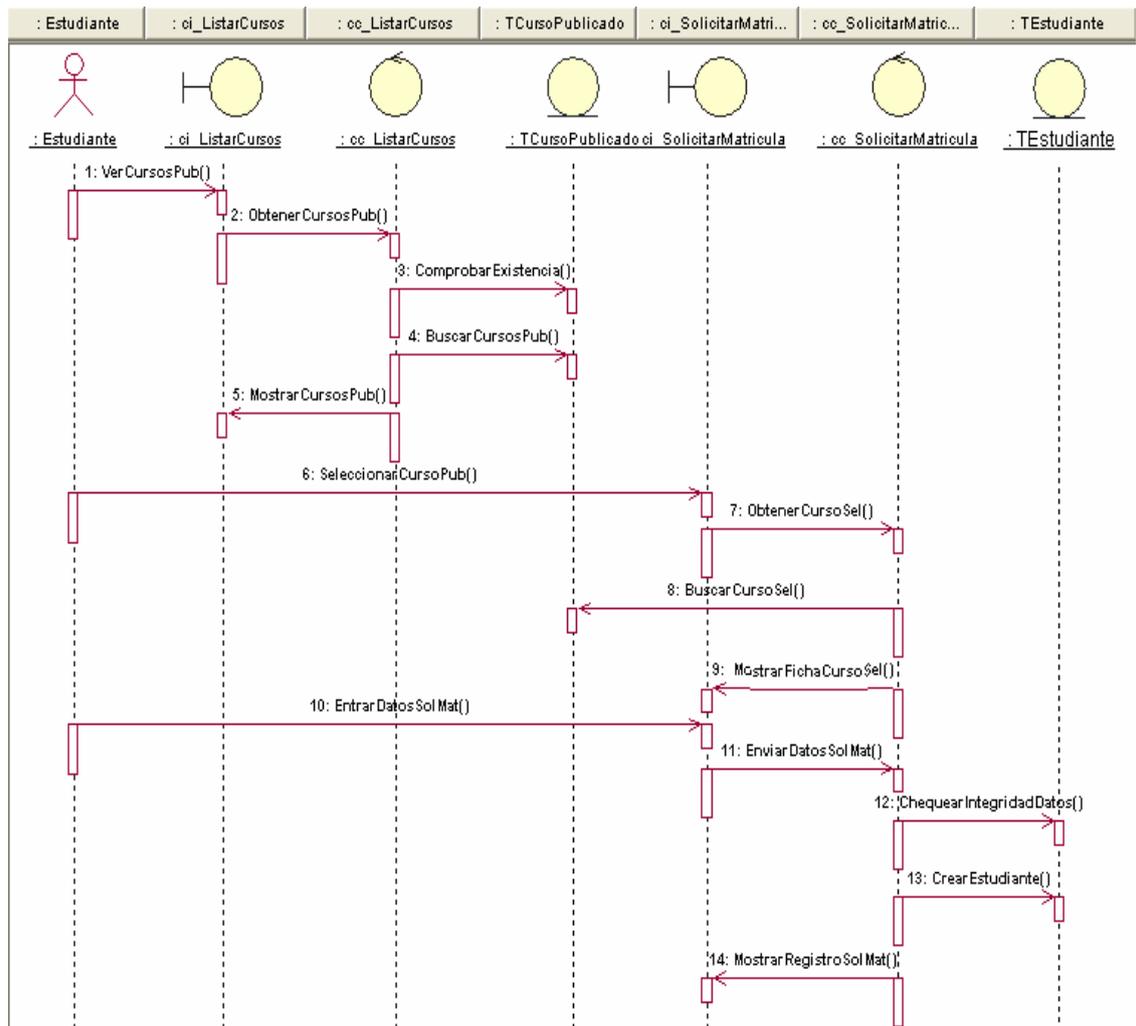
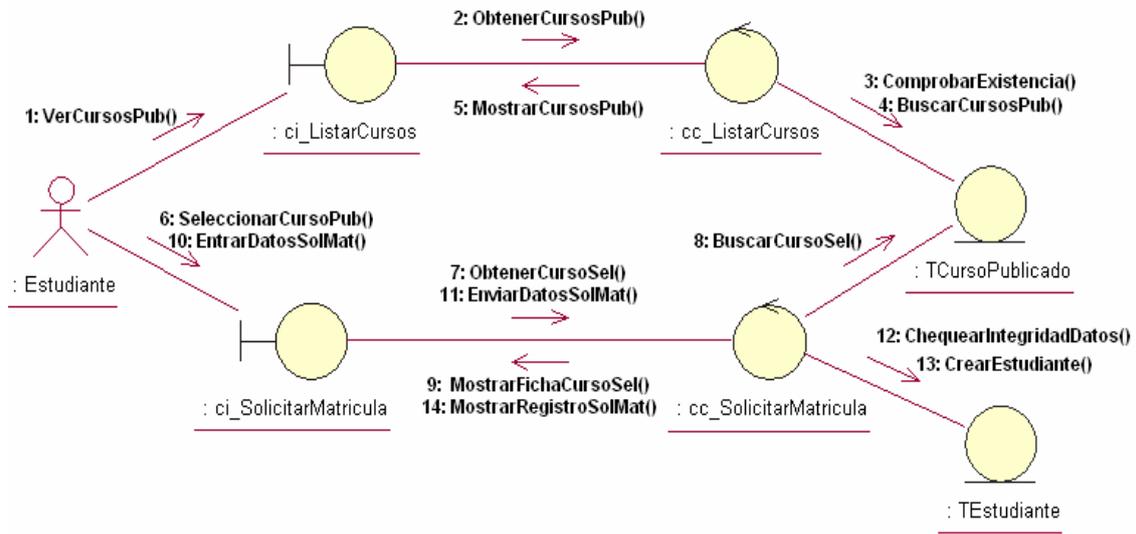
Anexo A

Diagramas de Colaboración y Secuencia del Análisis

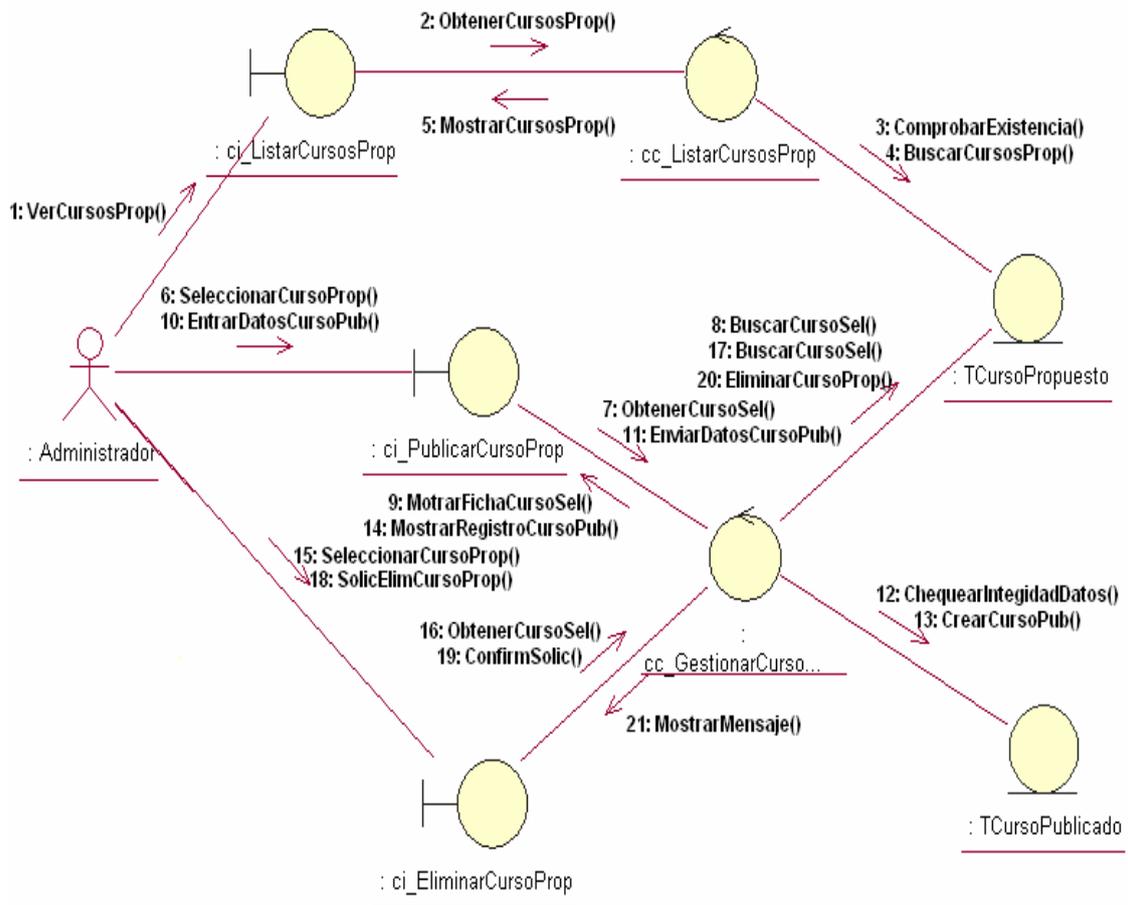
A.a) CU1: Autenticar usuario

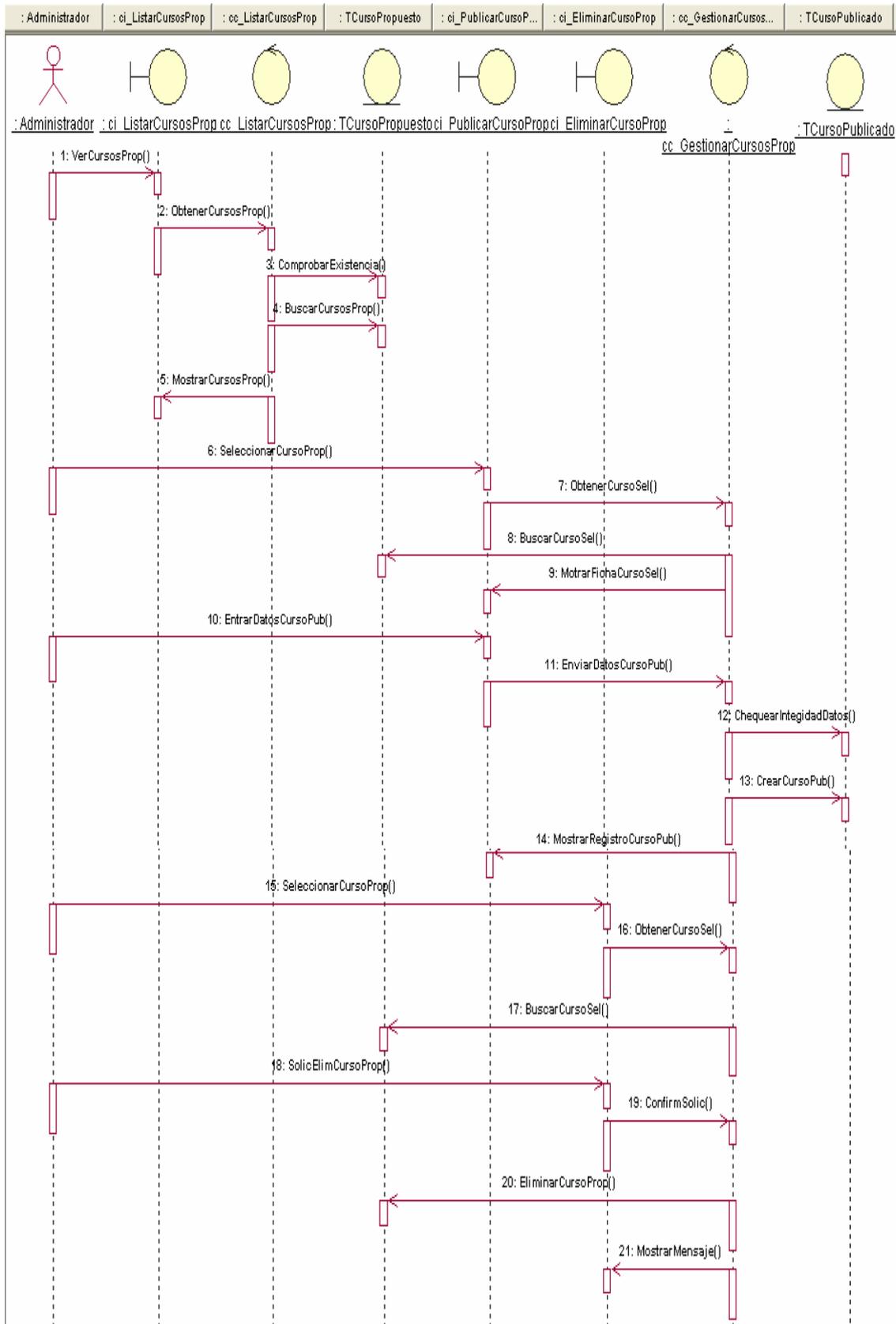


A.b) **CU2: Solicitar matrícula.**

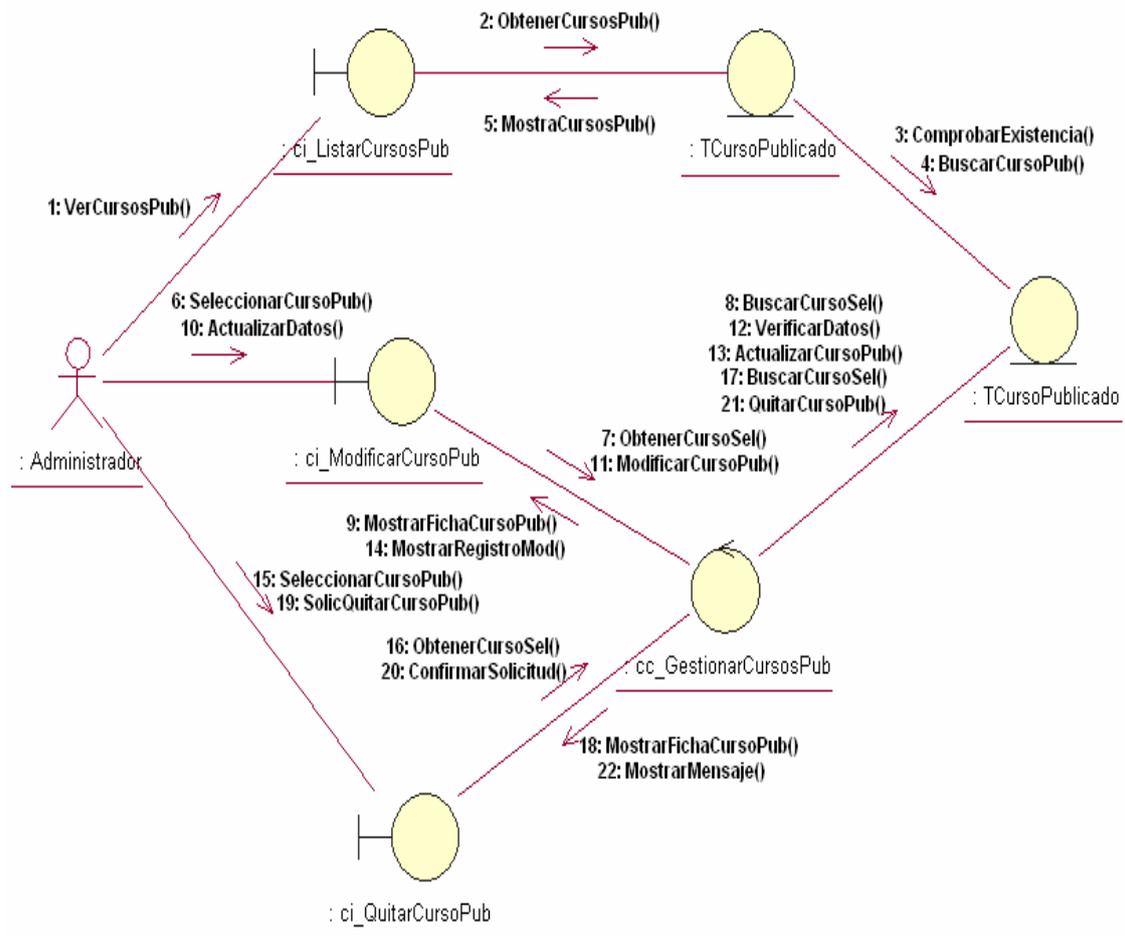


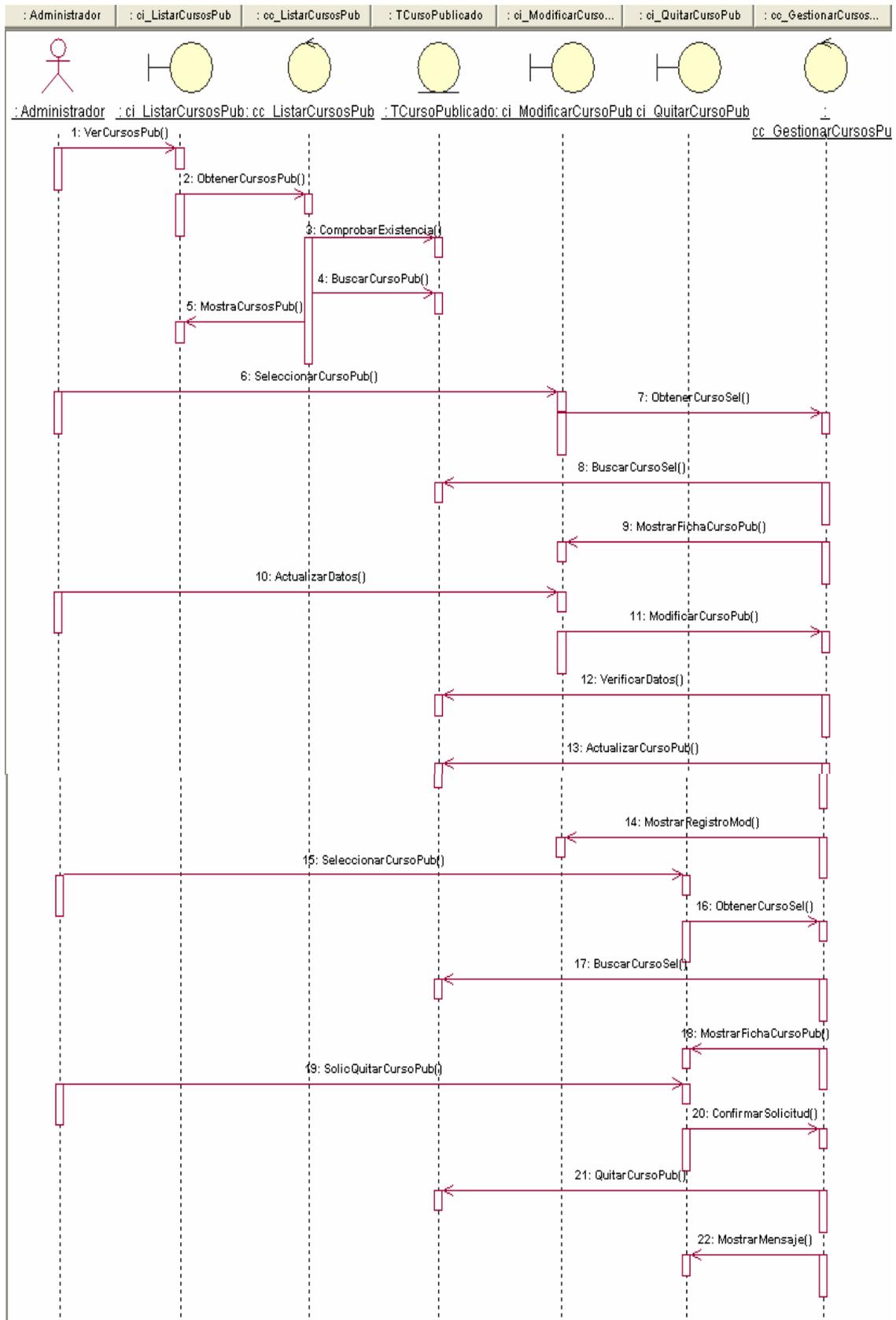
A.c) **CU3: Gestionar cursos propuestos.**



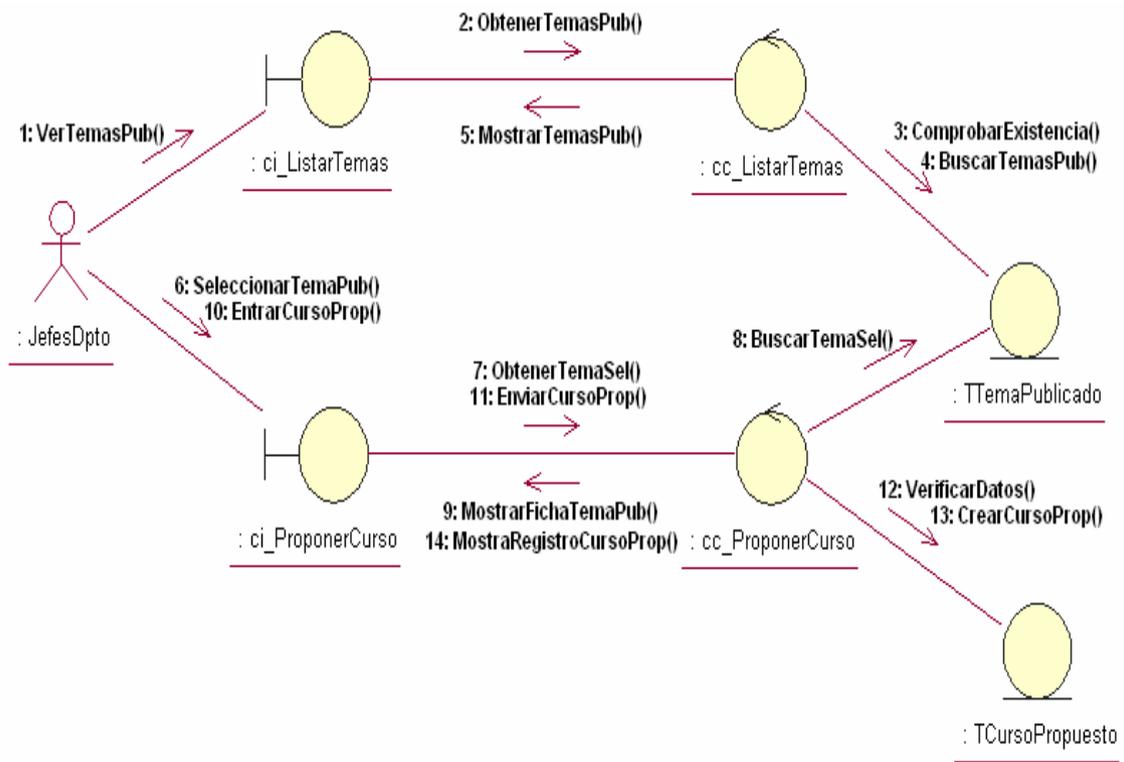


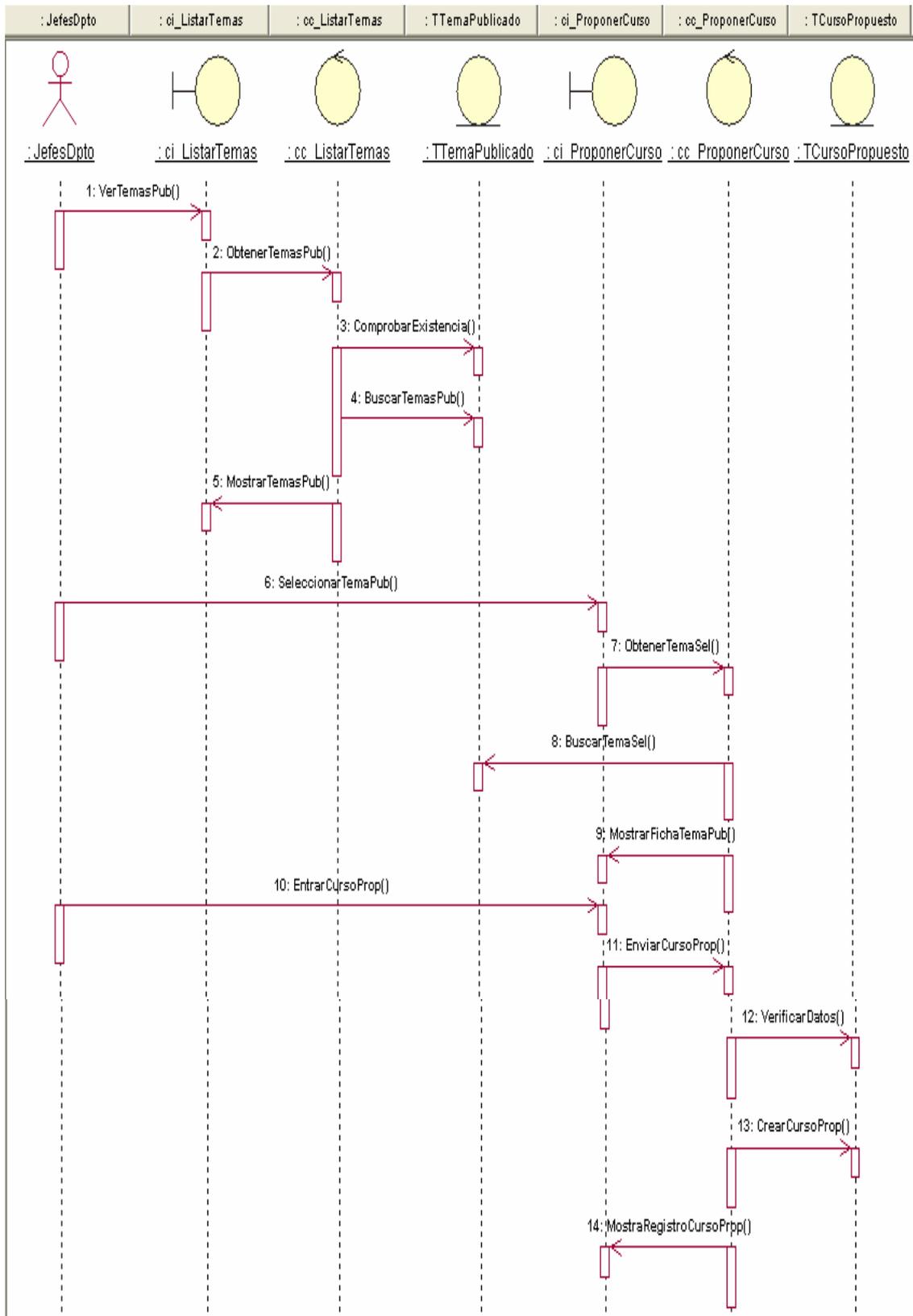
A.d) **CU4: Gestionar cursos publicados.**



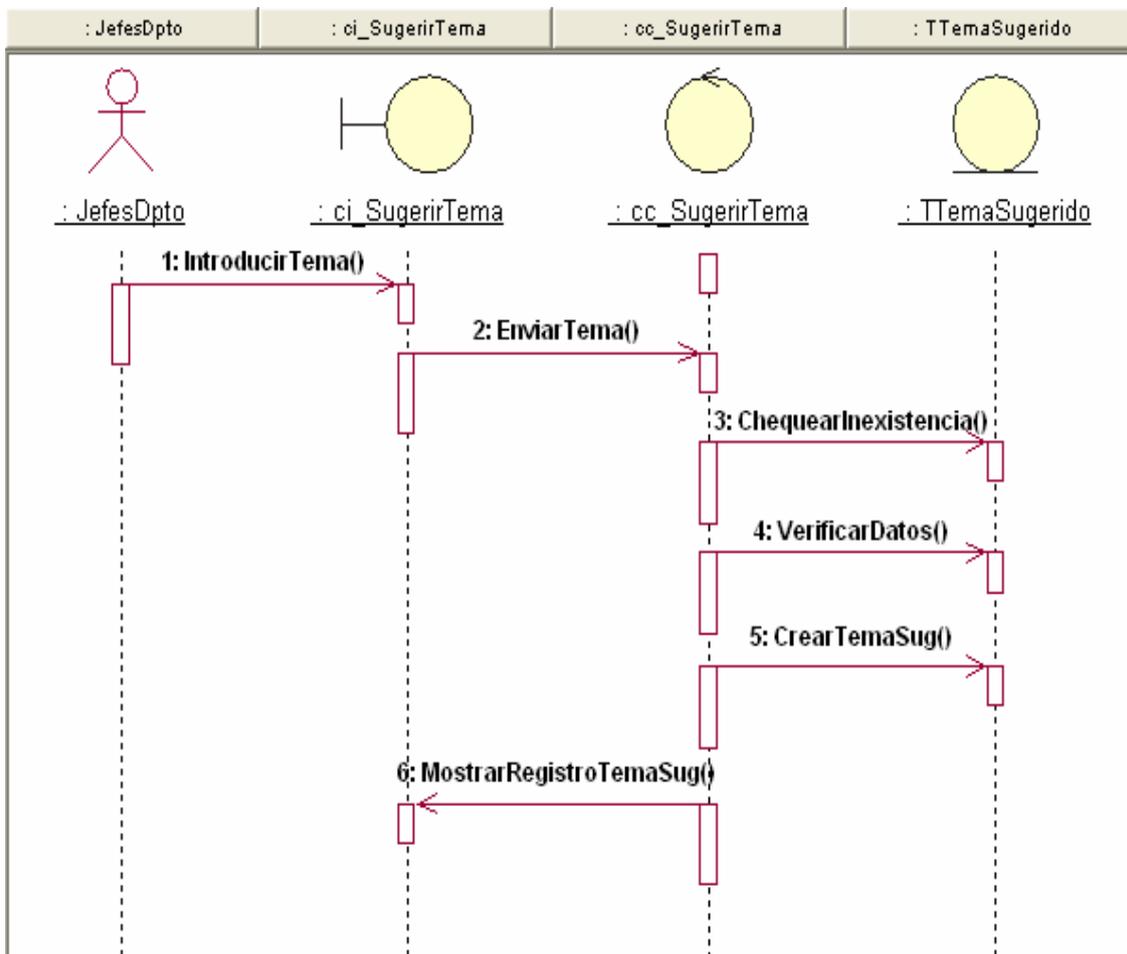
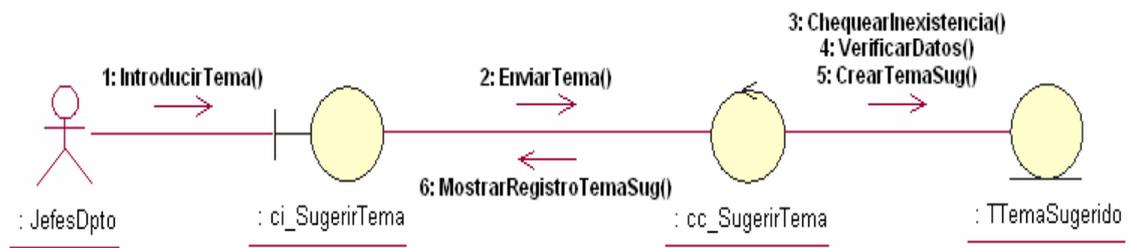


A.e) **CU5: Proponer curso.**

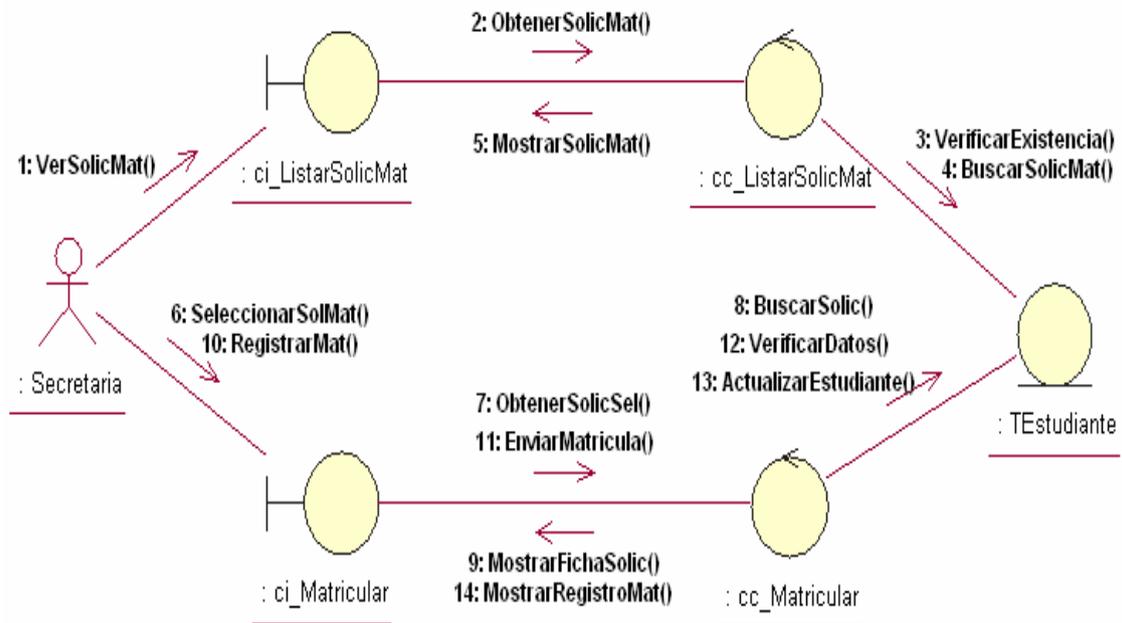


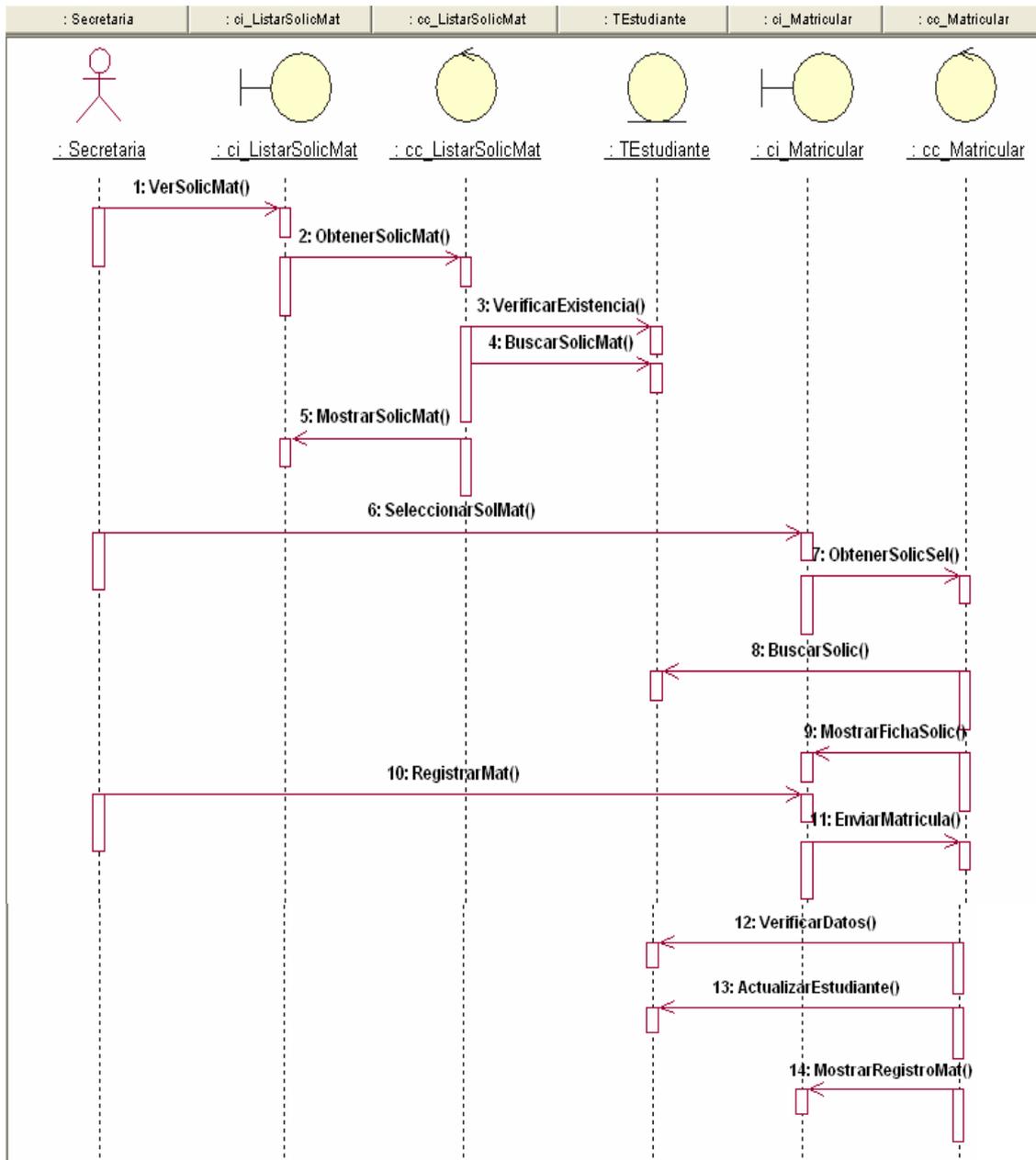


A.f) **CU6: Sugerir tema.**

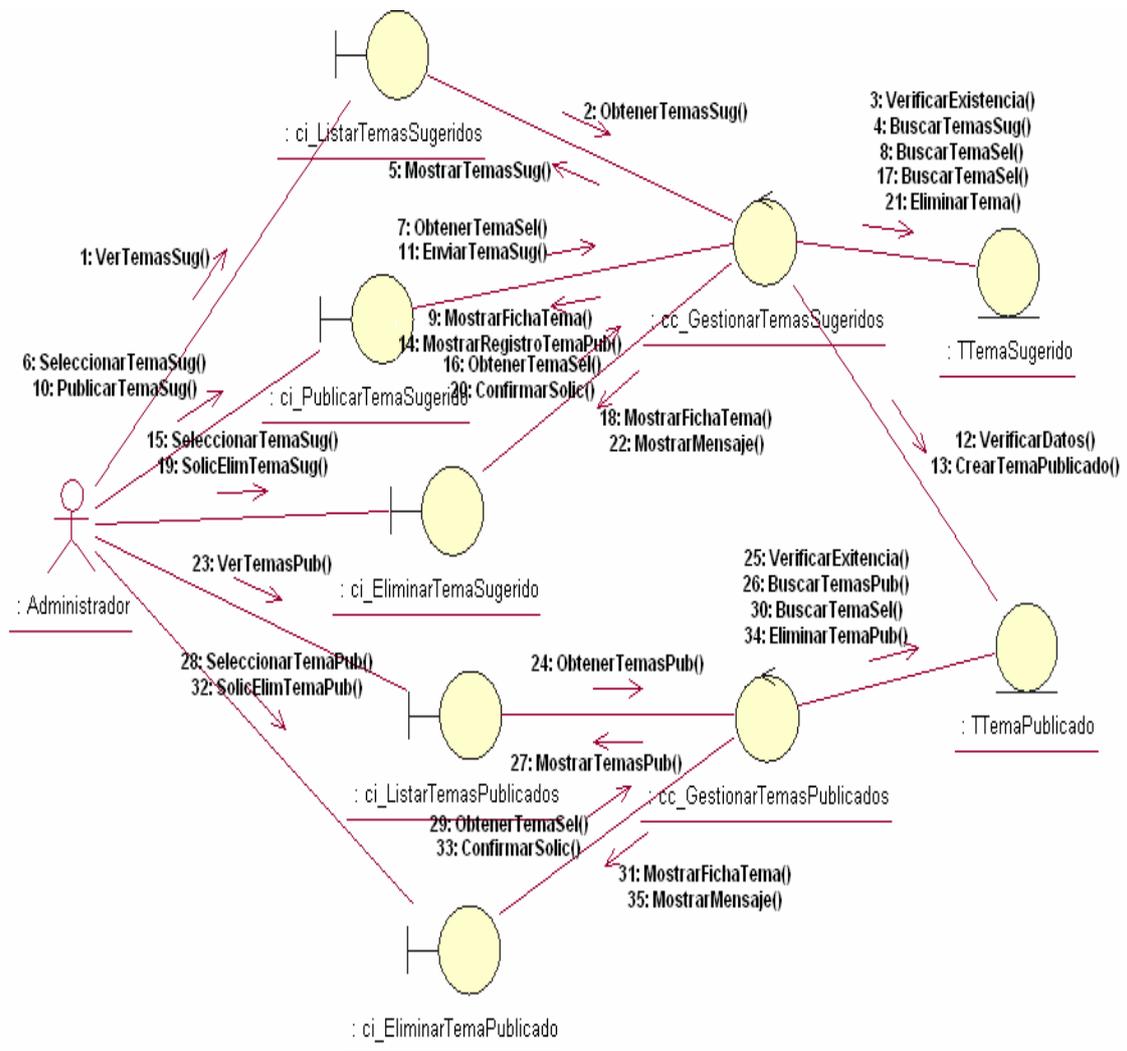


A.g) **CU7: Realizar matrícula.**

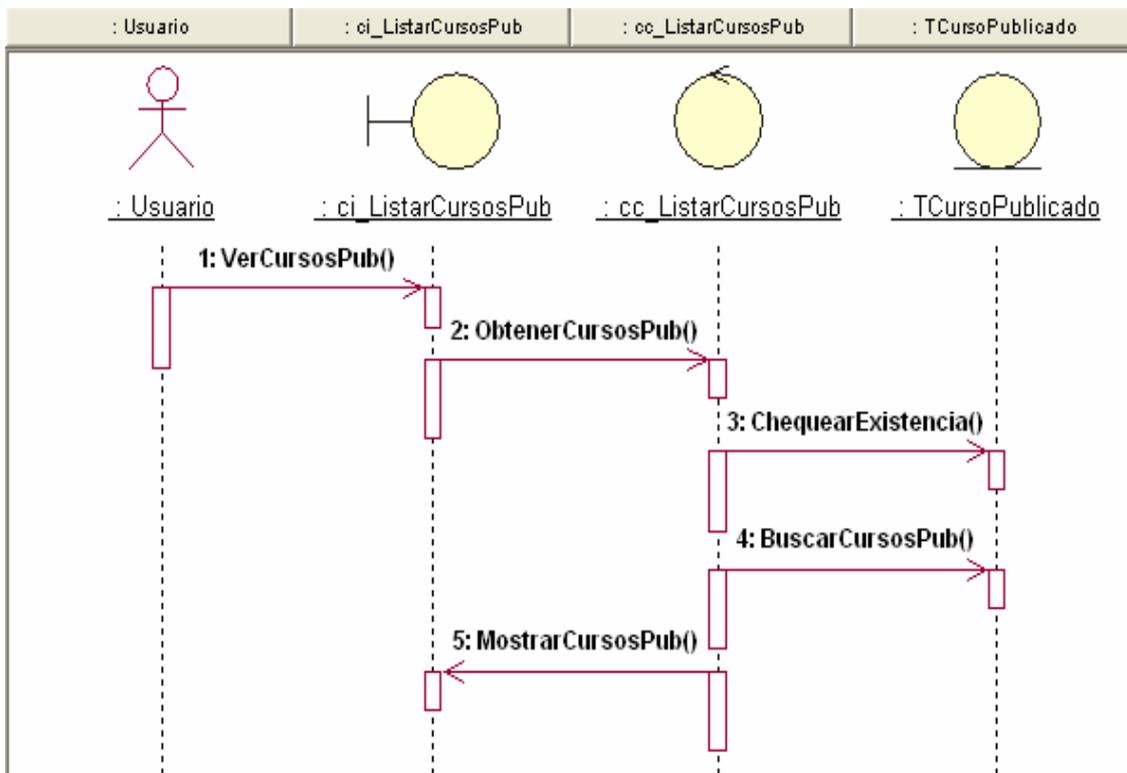
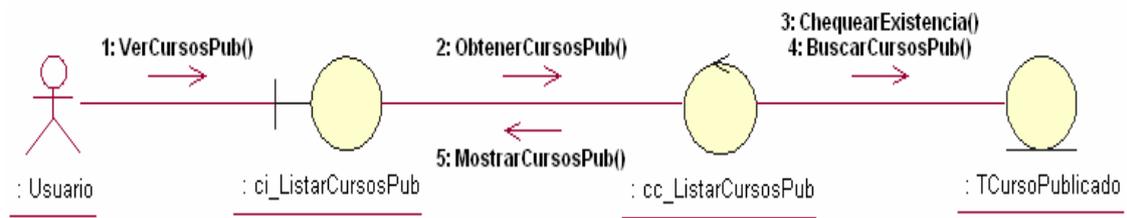




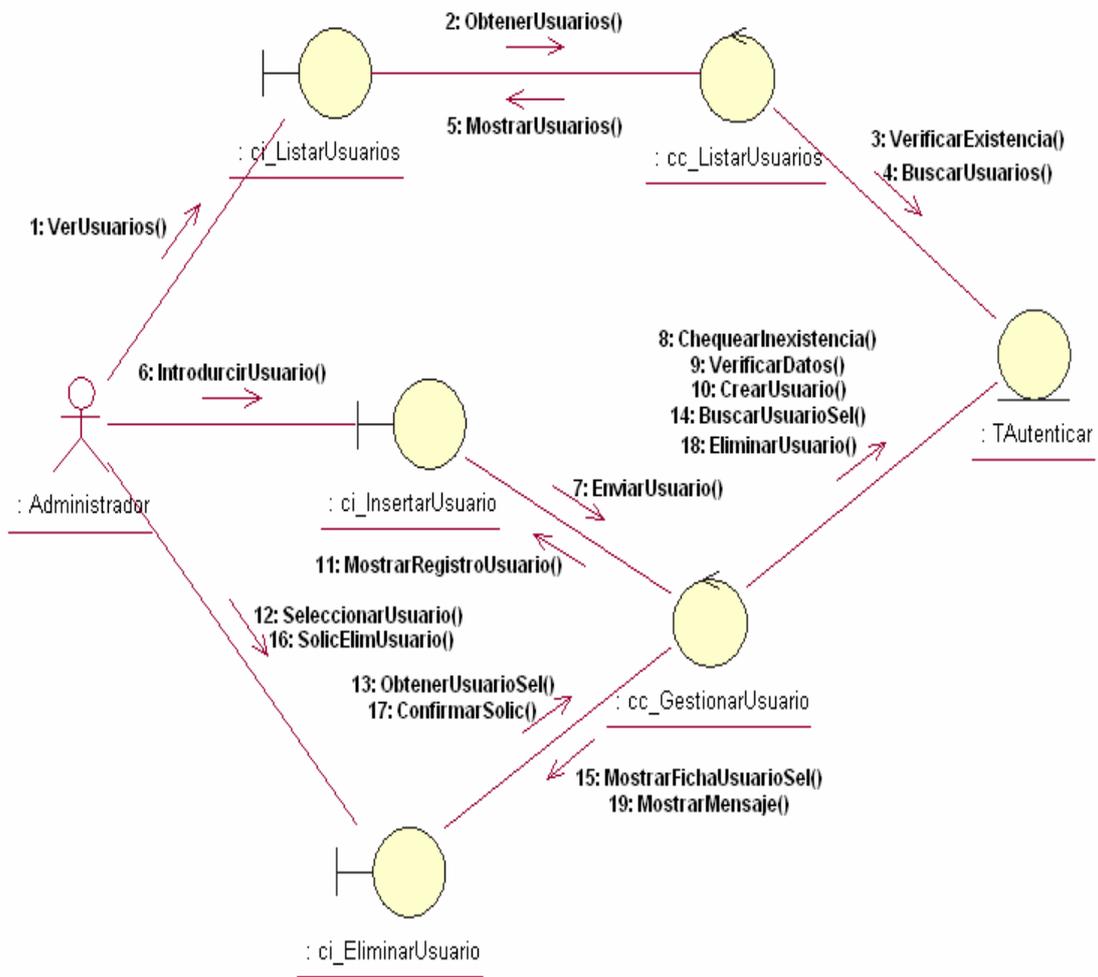
A.h) **CU8: Gestionar temas.**

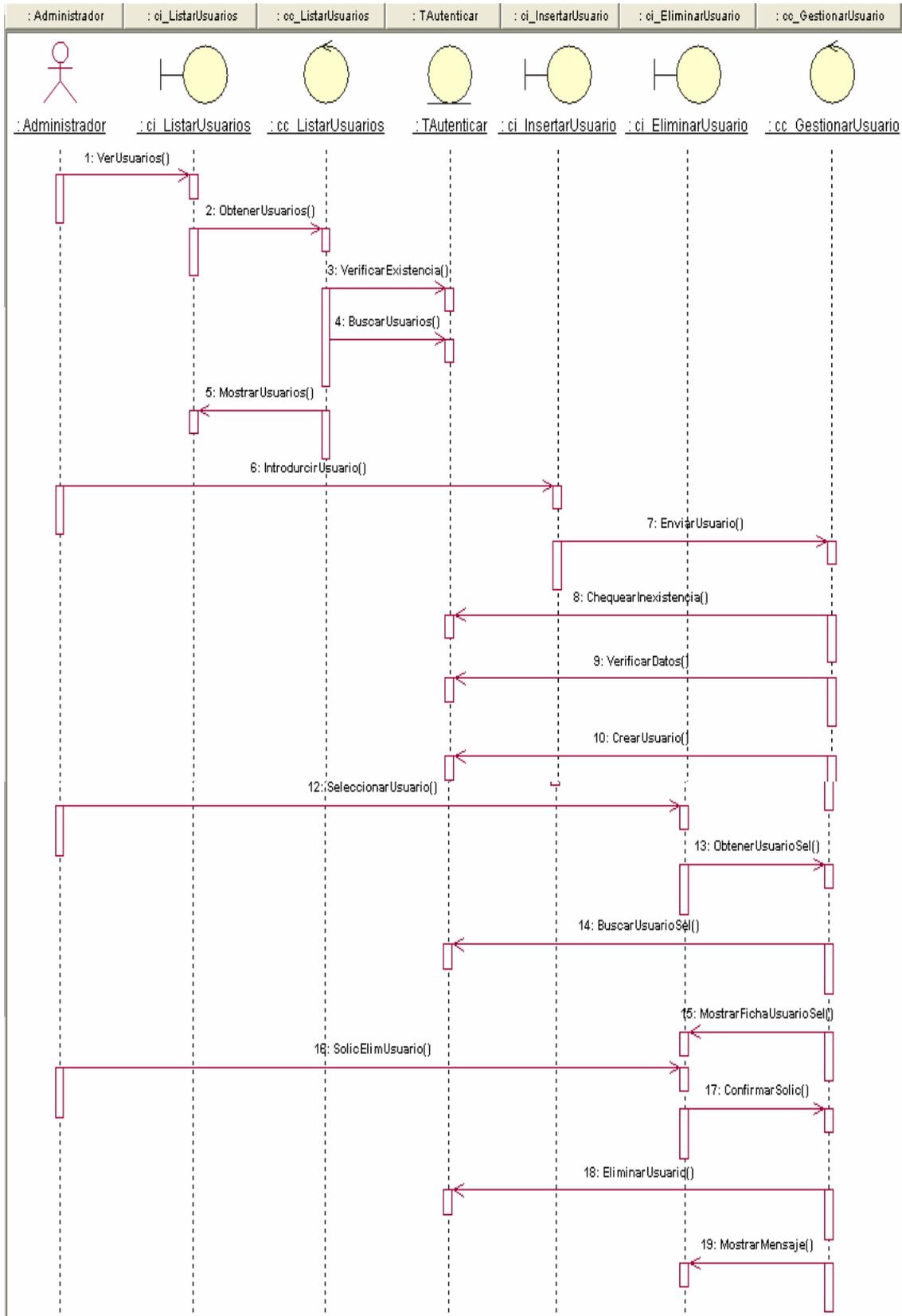


A.i) **CU9: Listar cursos disponibles.**



A.j) **CU10: Gestionar usuarios.**

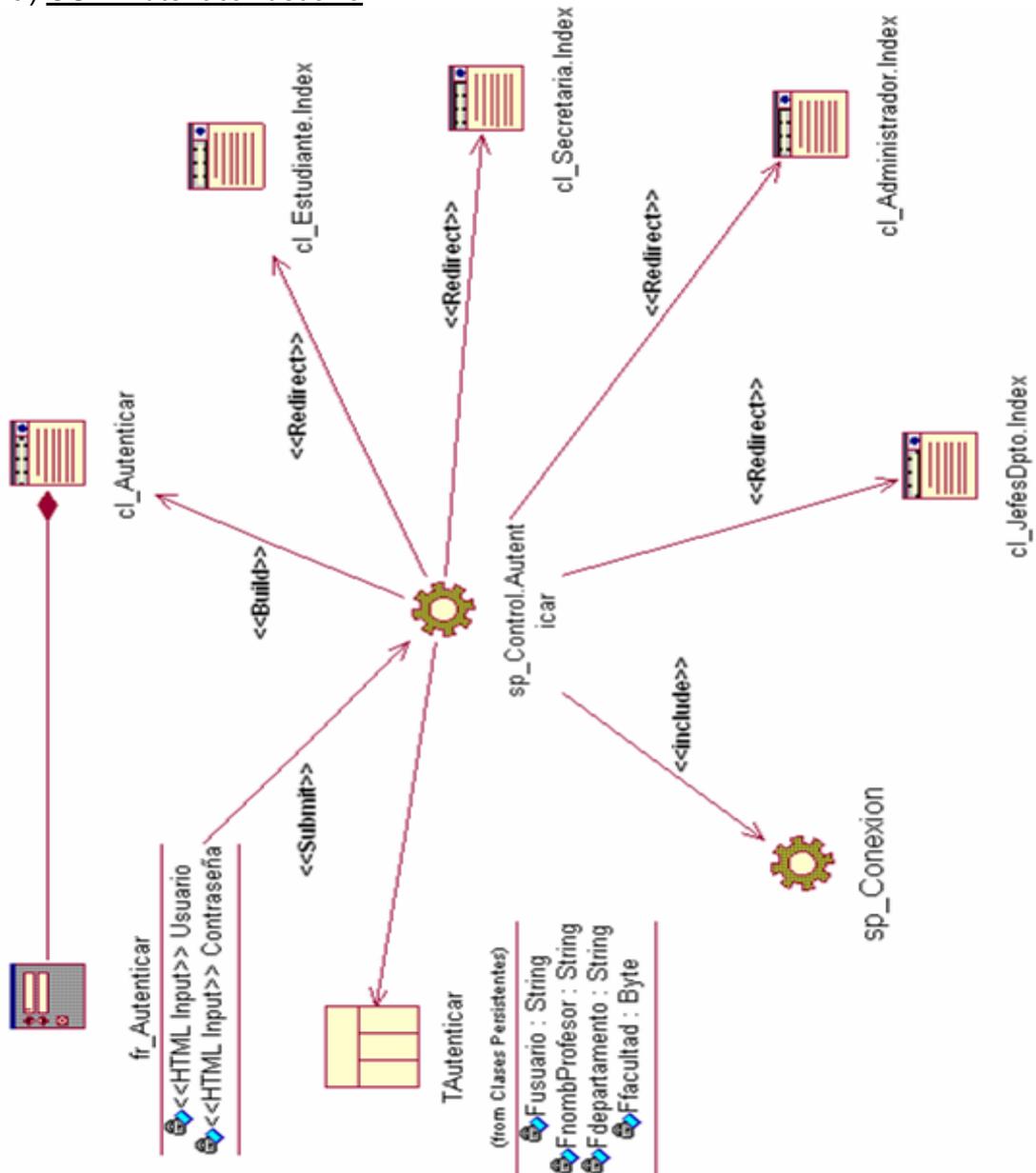




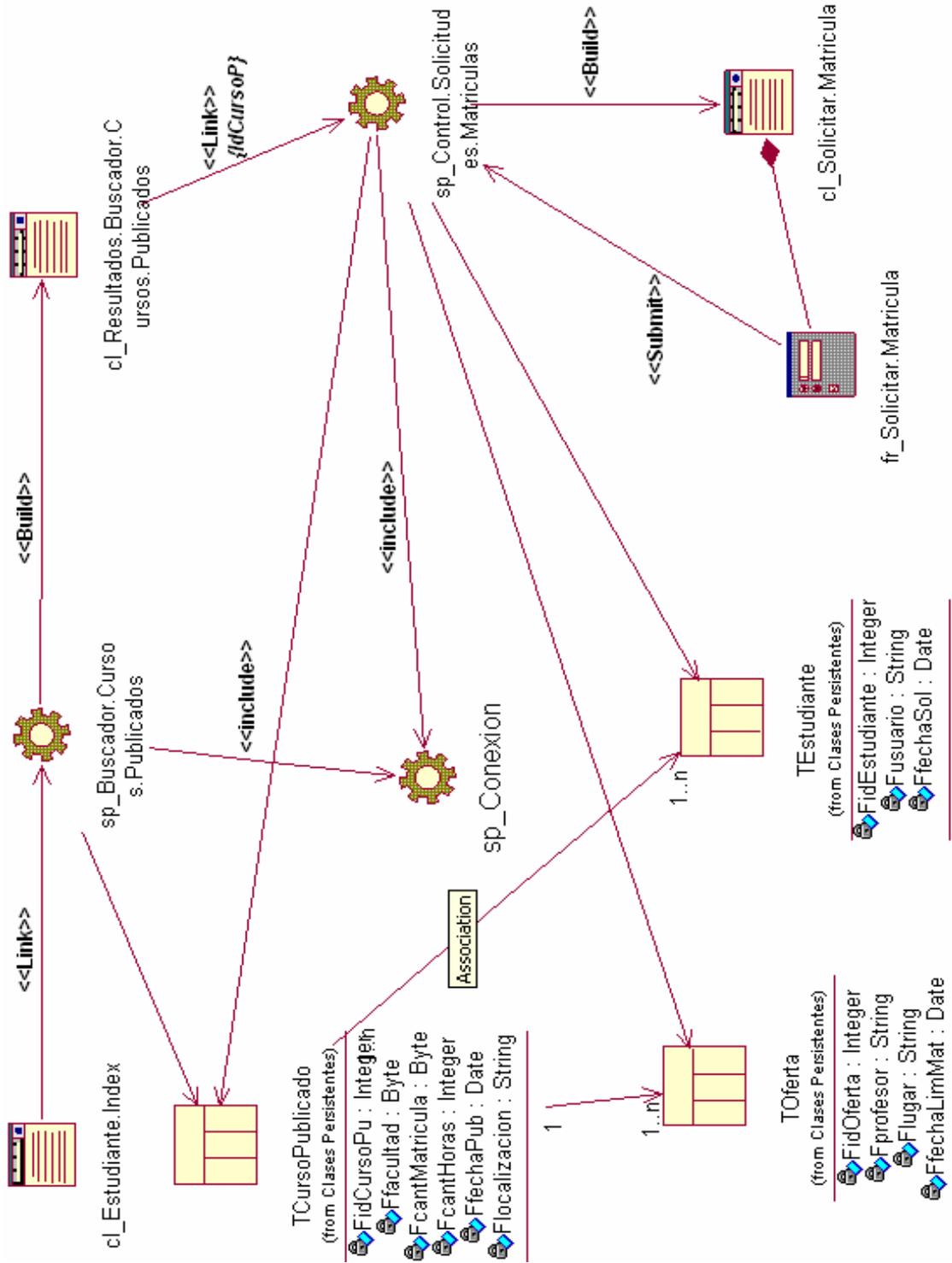
Anexo B

Diagramas de Clases de Diseño

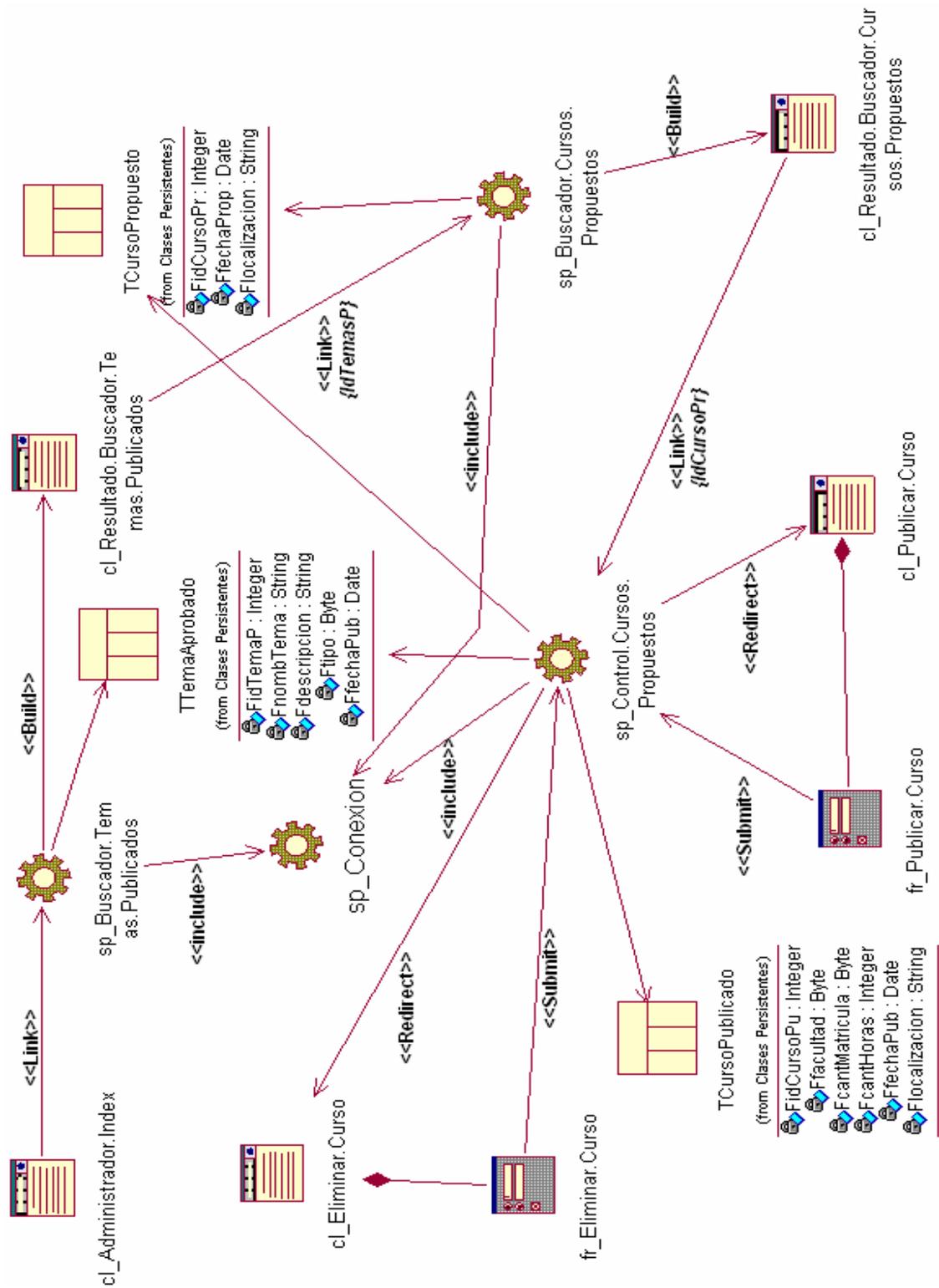
B.a) CU1: Autenticar usuario



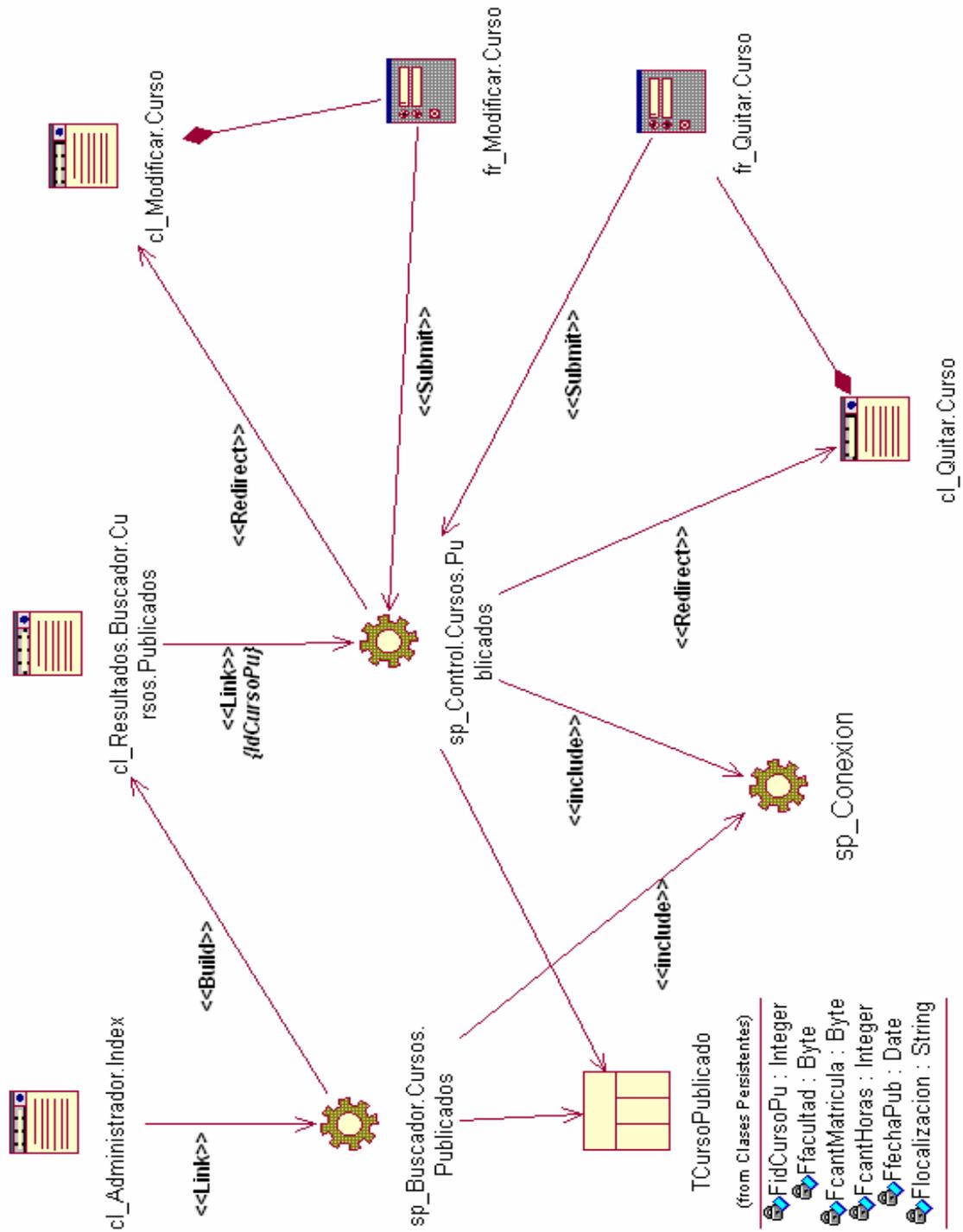
B.b) **CU2: Solicitar matrícula**



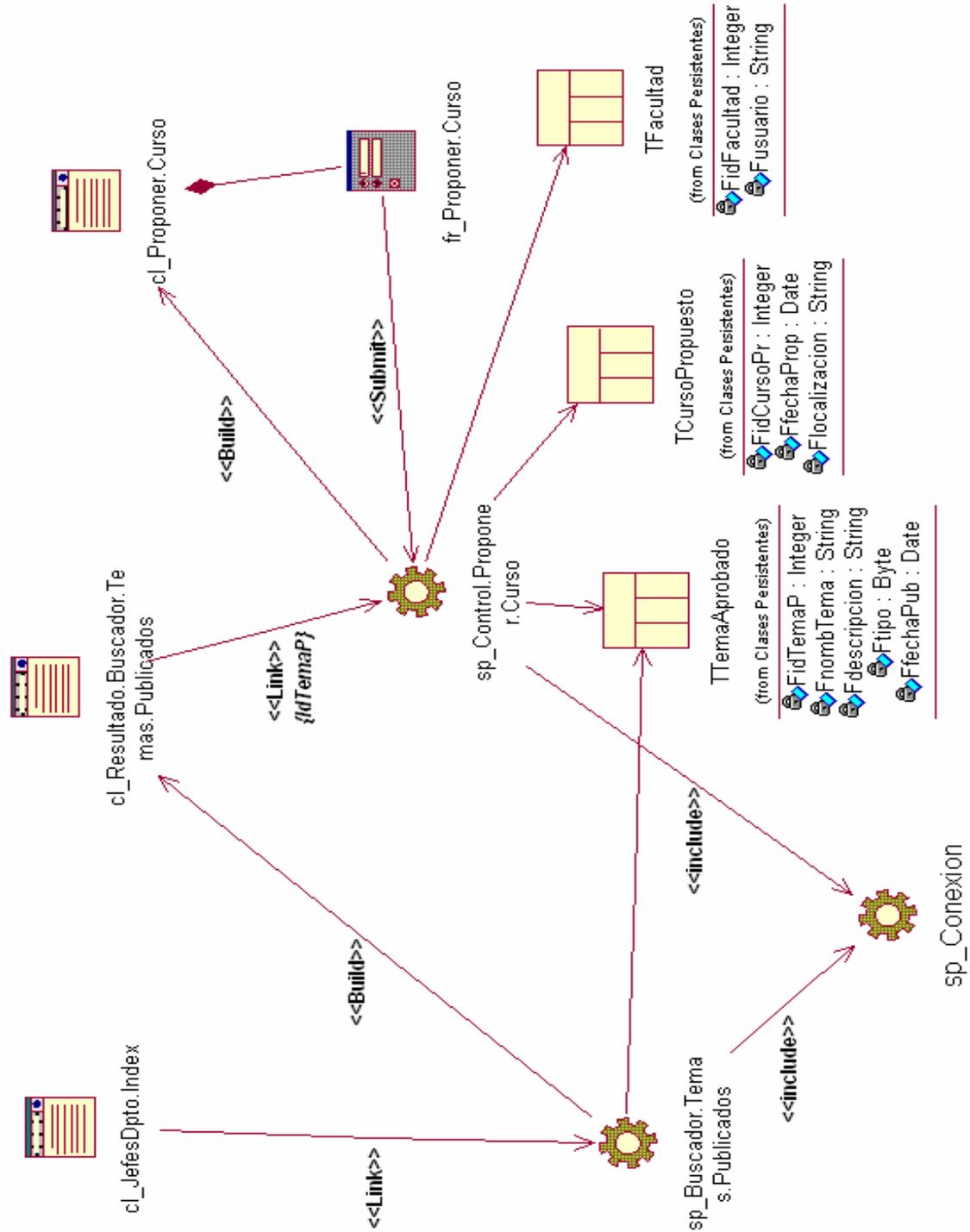
B.c) **CU3: Gestionar cursos propuestos**



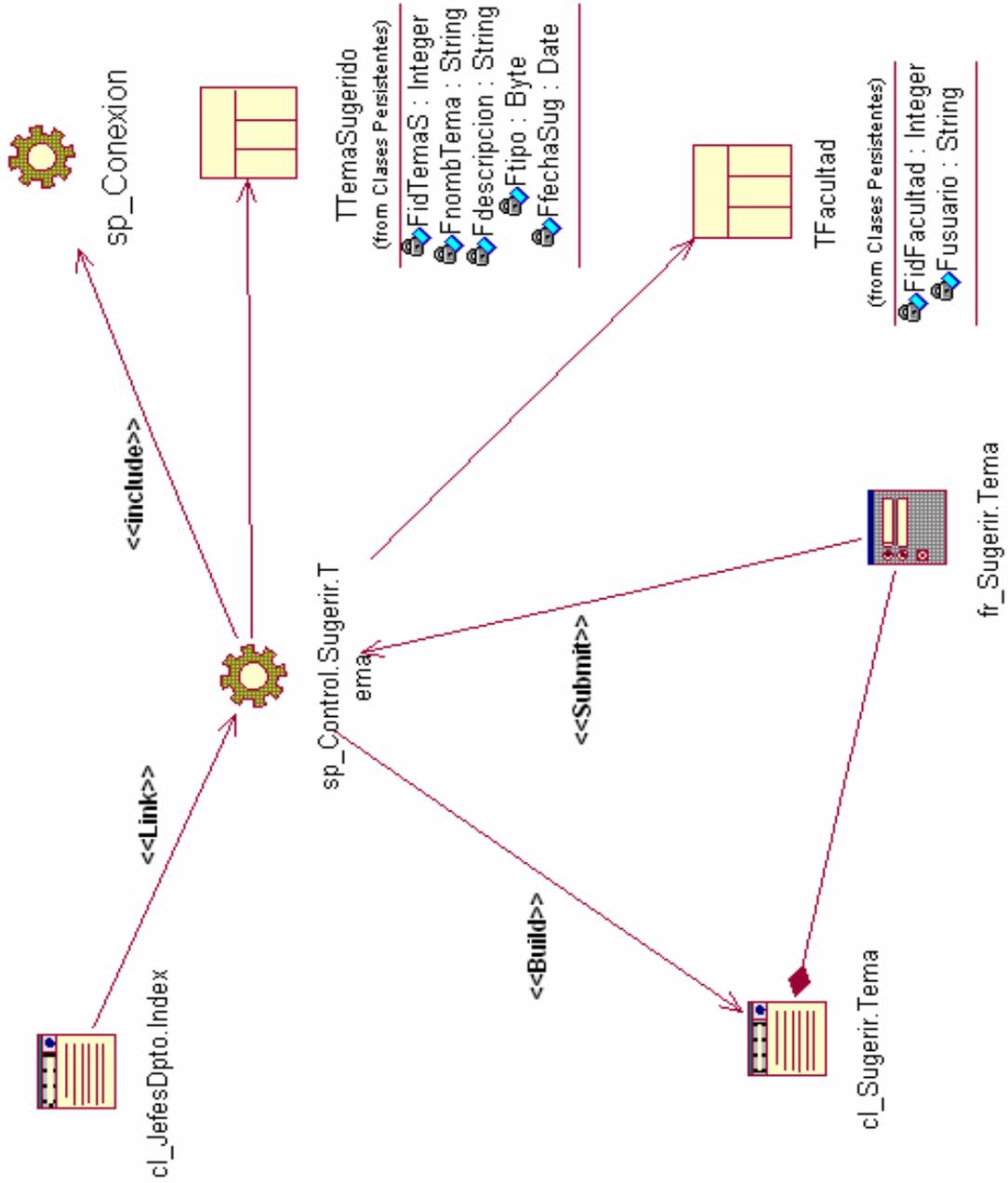
B.d) **CU4: Gestionar cursos publicados**



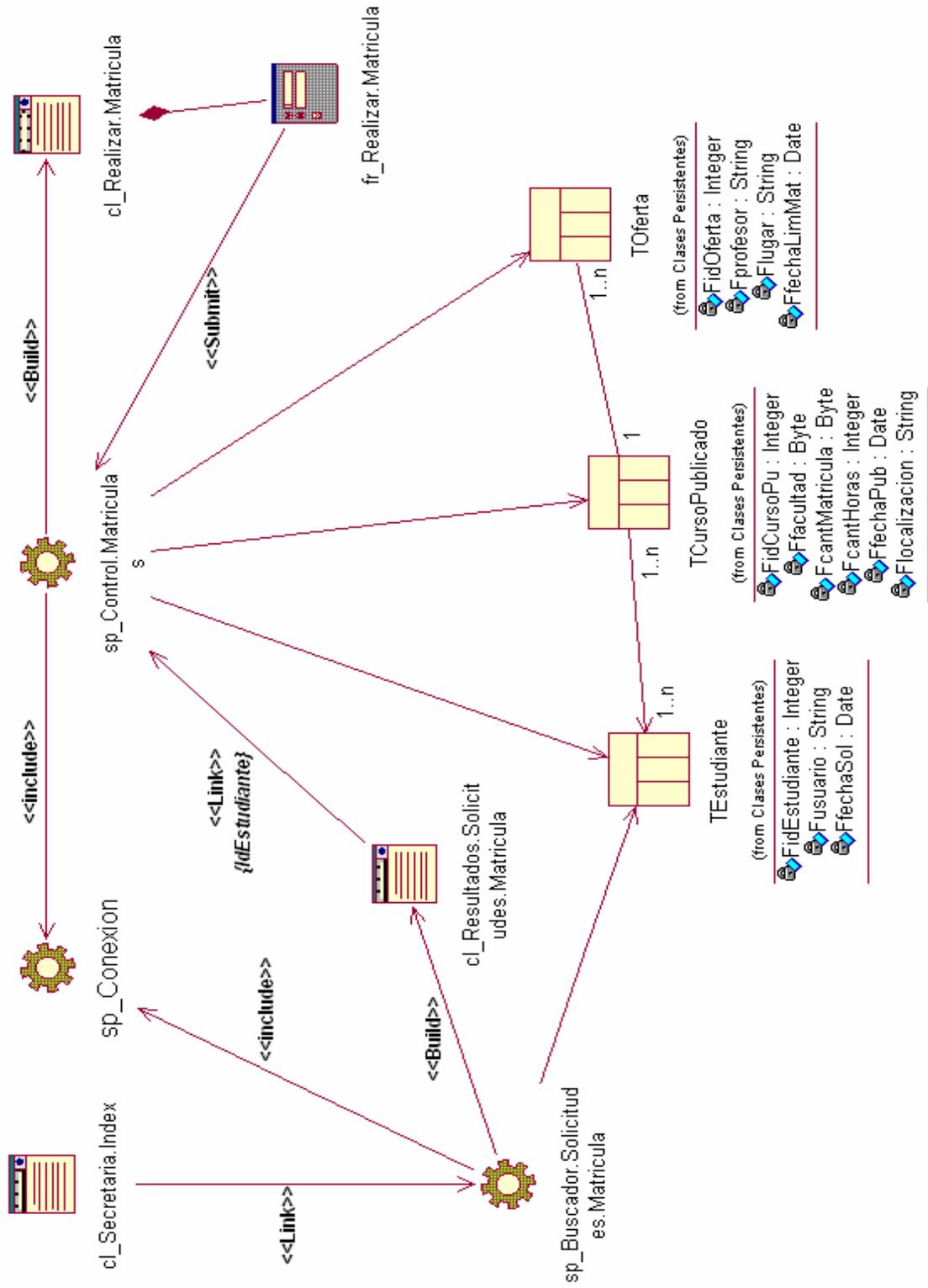
B.e) **CU5: Proponer curso**



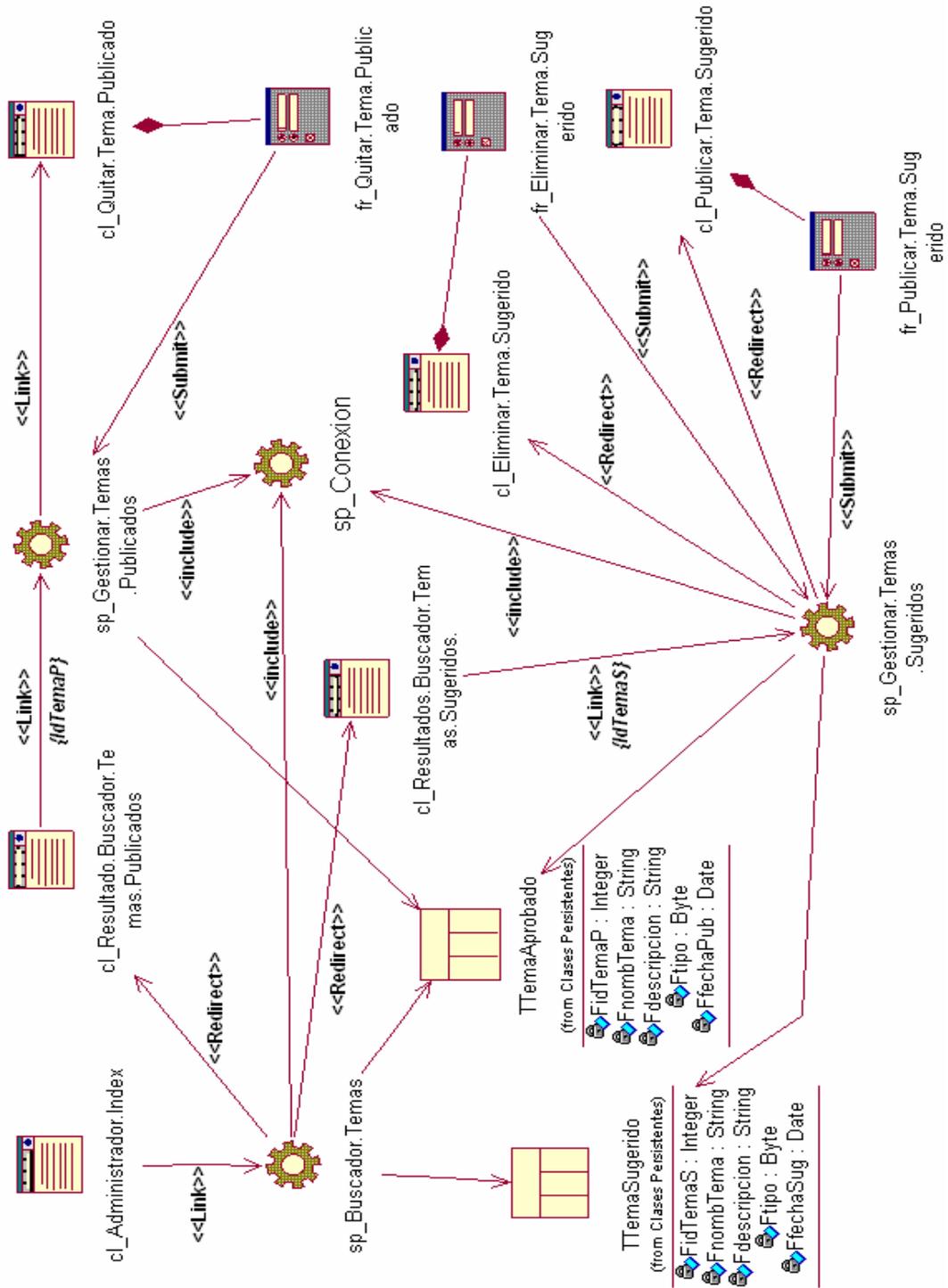
B.f) **CU6: Sugerir tema**



B.g) CU7: Realizar matrícula



B.h) **CU8: Gestionar temas**



B.i) **CU9: Listar cursos disponibles**

