

Instituto Superior Politécnico “José Antonio Echeverría”

Facultad de Ingeniería Industrial

Ingeniería en Informática



**SISTEMA DE GESTIÓN ACADÉMICA: MÓDULO DE  
GESTIÓN DE PROFESORES**

**Trabajo para optar por el título de ingeniería en Informática**

**AUTOR: YIRCY DIEM COLLAZO MARÍN**

**TUTOR: ING. EMIL LIMA VALDÉS**

**Ciudad de la Habana**

**Junio de 2005**



*A mis padres...*

## **AGRADECIMIENTOS**

A mi abuela Silvia, que aunque ya no este la quiero mucho, la extraño y siempre va a formar parte de mi.

A mi mamá y a mi papá, por su paciencia y dedicación. Gracias por su confianza y ayuda. Los quiero.

A Betty, mi segunda mamá porque aunque lejos siempre has estado muy cerca de mi.

A mis demás abuelos, que siempre estarán presentes.

A Yulier, por todo este tiempo juntos y por regalarme tantos momentos que no voy a olvidar.

A mi prima Betsy, por su cariño y por sus consejos.

A mi mejor amiga Tania, por apoyarme siempre y por los años que hemos compartido juntas. Gracias por todo.

A Alexander, por aceptarme como soy y por darme la oportunidad de conocerlo. Gracias por estar a mi lado en estos últimos tiempos.

A Aliett, Sandra y Leticia por su amistad, a Maikel por permitirme confiar en él, a Lachy por ser tan buen compañero, a Yannia, Dialen, Yanelis, Gema, Yoselin, Jandrich, Yoiset.

A Emil mi tutor, por su ejemplo y apoyo.

A Paco por su afecto y confianza.

A todos mis compañeros de cinco años de estudio y esfuerzo, por ser tan buenos amigos.

A Pepe, Frank, Yanet, Ivis, Patricia, Saray, Reiner, Grency, Elieser, Rudel, Rafael, Sandy, Joel mis compañeros de Gastex por compartir conmigo todos estos días de trabajo y sacrificio.

## **DECLARACIÓN DE AUTORÍA**

Declaro que soy la única autora de este trabajo y autorizo a la Universidad de Ciencias Informáticas de Cuba (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los 20 días del mes de junio de 2005.

---

Firma del Autor

---

Firma del Tutor

# Avales

## **OPINIÓN DEL TUTOR**

## RESUMEN

La Universidad de Ciencias Informáticas (UCI), se encuentra en proceso de informatización. En este sentido se está trabajando en el desarrollo de sistemas informáticos que sean capaces de sustituir los procesos que hasta el momento se desarrollan de forma manual o semi automatizada. El presente trabajo tiene como objetivo el desarrollo, dentro del Sistema de Gestión Académica, Akademos, de un Módulo de Gestión de los Profesores, capaz de encargarse de la gestión de la información relacionada con los docentes de la Universidad. El profesor es el recurso más importante con que cuenta un centro de estudios para llevar a cabo su labor formativa. La explotación de un sistema automatizado como el propuesto en este trabajo, permitirá, a todos los involucrados en la gestión de los profesores, realizar un seguimiento del desempeño de los docentes y de esta forma contar con información fiable para tomar decisiones acertadas en pos de aumentar la calidad de los profesionales formados en el centro. Para su desarrollo se siguieron los pasos que propone el Proceso Unificado. Está implementado sobre la plataforma Microsoft .NET, específicamente en el lenguaje C#.



# ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>1</b>
Situación problemática .....	1
Actualidad y necesidad del trabajo.....	2
Objetivos generales.....	3
Objetivos específicos .....	4
Tareas desarrolladas para cumplir los objetivos .....	5
Estructuración del contenido .....	5
<b>Capítulo 1 FUNDAMENTOS TEÓRICOS.....</b>	<b>7</b>
Introducción.....	7
1.1    Sistemas de Gestión Académica.....	7
1.2    El Proceso de Gestión Académica en la UCI .....	7
1.3    Sistema de Gestión Académica. Akademos.....	9
1.3.1    Módulos de Akademos.....	9
1.4    Gestión de Profesores.....	12
1.5    Aplicaciones Web .....	12
1.6    Servicios Web.....	13
1.7    Microsoft .NET .....	14
1.7.1    Visual Studio.NET .....	17
1.7.2    ASP .NET.....	18
1.7.3    Arquitectura basada en componentes.....	20
1.8    Modelo Cliente-Servidor .....	24
1.9    Sistemas de Gestores de Bases de Datos .....	26
1.9.1    SQL Server .....	27
Conclusiones.....	28
<b>Capítulo 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....</b>	<b>29</b>
Introducción.....	29
2.1    Estado actual del negocio.....	29
2.2    Reglas del negocio a considerar .....	31

2.3	Modelo del Negocio .....	32
2.3.1	<i>Modelo de casos de uso del negocio</i> .....	32
2.3.2	<i>Definición de actores y trabajadores del negocio</i> .....	32
2.3.3	<i>Expansión de los casos de uso del negocio</i> .....	36
2.3.4	<i>Diagramas de actividades</i> .....	41
2.3.5	<i>Diagrama de clases del modelo de objetos</i> .....	45
2.4	Solución propuesta .....	46
2.5	Requerimientos funcionales .....	48
2.6	Requerimientos no funcionales .....	50
2.7	Beneficios Tangibles.....	52
2.8	Beneficios Intangibles.....	52
2.9	Análisis costo – beneficios.....	53
2.10	Modelo del sistema.....	53
2.10.1	<i>Modelo de casos de uso del sistema</i> .....	53
2.10.2	<i>Diagrama de casos de usos del sistema</i> .....	54
2.10.3	<i>Casos de uso del sistema</i> .....	56
	Conclusiones.....	68
<b>Capítulo 3 Construcción de la Solución Propuesta .....</b>		<b>70</b>
	Introducción.....	70
3.1	Diagramas de clases del diseño Web.....	70
3.2	Diseño de la base de datos .....	70
3.3	Diagrama de clases persistentes.....	70
3.4	Modelo de datos .....	71
3.5	Principios de diseño.....	73
3.5.1	<i>Estándares en la interfaz de la aplicación</i> .....	73
3.5.2	<i>Tratamiento de excepciones</i> .....	73
3.5.3	<i>Estándares de codificación</i> .....	74
3.6	Diagrama de despliegue .....	74
3.7	Modelo de implementación.....	75
3.8	Diagramas de Componentes.....	75

Conclusiones.....	75
Conclusiones.....	77
<b>Recomendaciones .....</b>	<b>78</b>
<b>Bibliografía .....</b>	<b>81</b>
<b>Glosario de Términos.....</b>	<b>82</b>
<b>Anexos.....</b>	<b>83</b>

## **INTRODUCCIÓN**

En la Universidad de Ciencias Informáticas (UCI) se tiene aspiración de hacer de ella una gran ciudad digital donde la mayor cantidad de información posible se encuentre automatizada, para esto las tareas fundamentales de la institución deben tener un alto nivel de informatización por lo que es necesario el desarrollo de sistemas informáticos que sustituyan los procesos que actualmente se realizan de forma manual.

### **Situación problemática**

En la actualidad en la Universidad de las Ciencias Informáticas (UCI) el proceso de Gestión de Profesores se realiza totalmente de forma manual, esto implica un gran esfuerzo por parte de los involucrados en este proceso ya que no cuentan con una herramienta que facilite el trabajo con los profesores, excepto algunos reportes y el caso del plan de trabajo del profesor que se realiza con la ayuda de la herramienta Microsoft Word.

En la UCI actualmente se hace una tarea engorrosa mantener un control de las plantillas de profesores por lo que es necesario realizar un procesamiento de la información de manera automatizada que permita obtener un balance de la carga docente de estos. También se debe recoger y almacenar además toda la información referente a su trayectoria en el centro, resultados de participación en las actividades, cargos que desempeña, trabajo investigativos que realiza y responsabilidades en las tareas que continuamente forman parte del quehacer educativo.

No se cuenta actualmente con una aplicación Web que organice y gestione toda esta información. Solo con sistemas sobre Excel que permiten recoger una pequeña parte de la información preliminar de los profesores, requerida en la entrevista que realizan los directivos previamente al proceso de ingreso del profesor en el centro. Esto provoca que la información obtenida en la entrevista, no esté organizada adecuadamente y la forma en que se guarda no ofrece la mejor protección, además se dificulta en gran medida la actualización y corrección de los datos almacenados.

Ante esta situación se presenta el siguiente problema:

La automatización del proceso de gestión de profesores que permita llevar el control de toda la información referente a la planificación de los profesores en la Universidad.

### ***Actualidad y necesidad del trabajo***

En la actualidad en la institución no se cuenta con un sistema informatizado, capaz de ayudar a realizar el proceso de planificación del profesor; el cual se lleva a cabo de forma manual lo que conlleva a un mayor trabajo y esfuerzo a todos los implicados en este proceso. La ausencia de este sistema automatizado provoca que esta tarea en la Universidad pueda llegar a ser ineficiente y muchas veces con errores, revirtiéndose en la insatisfacción de las necesidades de la Universidad. Por esto que se hace evidente la realización de un Sistema para la Gestión de los Profesores que pueda además brindar información tanto a Directivos como a otros sistemas del centro.

Para el desarrollo de este sistema surgió la necesidad de investigar como trabajan los Sistemas de Gestión Académica y dentro de ellos específicamente los que de una forma u otra se relacionaban con la Gestión de los Profesores. Para ello fue necesario investigar como ocurría todo el proceso de planificación del profesor actualmente en la institución y se realizaron varias entrevistas al Director de Formación y a varios Jefes de Departamento de las Facultades que son los máximos involucrados en este proceso, para conocer como se realizaba esta función. Se analizaron los documentos necesarios para llevarlo a cabo y luego se pudo especificar las funcionalidades y características que debía tener el sistema para poder satisfacer las necesidades del usuario.

De la realización de este sistema se espera:

- Una herramienta de trabajo capaz de facilitar y gestionar de forma rápida toda la información referente a la Gestión de Profesores.

- Centralización de toda la información relacionada con el proceso de la Gestión de los Profesores.
- Que este integrado a otros sistemas del centro.
- Información actualizada de los profesores.
- Reducción de errores y del volumen de información a almacenar en papel.

En el objeto de estudio del presente trabajo se encuentra:

- Aspectos que presentan otros Sistemas de Gestión Académica.
- Aspectos que debe presentar un Sistema de Gestión de Profesores.
- Además de un estudio general de la Universidad y la gestión académica.
- Tecnologías Web para el desarrollo de aplicaciones cliente/servidor.
- Estudio del Sistema de Bases de Datos que será utilizado.
- Integración con otros sistemas.

Y como campo de acción de la investigación llevada a cabo para el desarrollo del sistema que se presenta se encuentra:

- El proceso de gestión académica en la Universidad.
- La Gestión de la Planificación del Profesor en el centro.
- Sistemas de gestión de bases de datos.
- La tecnología Microsoft .NET, en específico ASP .NET para el desarrollo de aplicaciones web.

### ***Objetivo general***

Desarrollar un Sistema Automatizado que facilite el trabajo a todos los implicados en el proceso de Gestión de Profesores y además permita llevar así un control de toda la información referente al profesor.

### ***Objetivos específicos***

- Analizar y estudiar algunos aspectos teóricos que tienen que ver con la Gestión Académica.
- Investigar sobre la Gestión de Profesores.
- Crear una herramienta que permita a todos los Directivos del centro así como al propio profesor, tener acceso a cualquier información sobre este.
- Realizar la modelación de la solución planteada siguiendo los pasos que propone el Proceso Unificado de Desarrollo de Software.

## ***Tareas desarrolladas para cumplir los objetivos***

Dentro de las tareas que se debieron realizar para garantizar el cumplimiento de los objetivos se encuentran:

- Realizar entrevistas a los implicados en el proceso, para obtener la mayor cantidad de información referente a la planificación del profesor en la institución actualmente, abordando aspectos muy importantes a tener en cuenta para el sistema, resultó extremadamente importante analizar el trabajo de los profesores en la institución pues estos son los mayores implicados en este proceso. Toda esta información se analizó y se utilizó para iniciar la modelación del sistema de Gestión de Profesores para la universidad.
- Analizar otros Sistemas de Gestión de Profesores que se utilizan en otras universidades fuera del país para así apoyarnos y que nos brinden información que pueda ayudarnos en la realización del mismo.
- Realizar un estudio de las tecnologías Web para el desarrollo de aplicaciones cliente/servidor.
- Analizar los objetivos fundamentales, las ventajas y desventajas de los sistemas gestores de bases de datos y decidir cual es el más óptimo a utilizar en nuestro sistema.

## ***Estructuración del contenido***

El contenido está estructurado en cuatro capítulos fundamentales. El primero, *Fundamentos Teóricos* muestra aspectos generales de los Sistemas de Gestión Académica y su funcionamiento, así como el porqué de la informatización. También se tratan otros aspectos importantes relacionados con el estudio de las tecnologías a utilizar en el desarrollo de la aplicación.

En el segundo capítulo, *Descripción de la Solución Propuesta*, que según el Proceso Unificado de Desarrollo de Software, se muestra la modelación del negocio realizada con el fin de entender el contexto del sistema a desarrollar, además, se



recogen los requerimientos que debe cumplir la aplicación en forma de casos de uso. Además se tratan los beneficios involucrados en la realización de la misma.

El tercer capítulo, *Construcción de la Solución Propuesta*, aborda los flujos de trabajo de diseño e implementación donde se modelan un grupo de artefactos.

# Capítulo 1 FUNDAMENTOS TEÓRICOS

## ***Introducción***

En este capítulo se presenta un análisis teórico de los principales conceptos asociados al dominio del problema con el objetivo de entender con más claridad los procesos del negocio, se ofrece un análisis teórico de las herramientas de desarrollo de aplicaciones Web, ASP .NET y del servidor de base de datos SQL Server.

### ***1.1 Sistemas de Gestión Académica***

La gestión académica constituye un elemento de gran importancia para el buen funcionamiento de cualquier centro educacional. Los sistemas de gestión académica tienen como tarea fundamental almacenar y procesar toda la información referente al proceso docente de un centro de estudios. Por lo general engloban todo lo relacionado a la matrícula y prematrícula de los estudiantes, al control de las evaluaciones y la asistencia de los mismos así como otros datos académicos importantes, además abarca toda la gestión de los planes de estudio y de los horarios de clases y de la información de los profesores. Estos sistemas constituyen una poderosa herramienta de trabajo que le permiten a cualquier centro de estudios hacer eficientes sus procesos, optimizando así sus recursos.

### ***1.2 El Proceso de Gestión Académica en la UCI***

La Universidad de las Ciencias Informáticas (UCI) constituye un centro de estudios dedicado a la formación de nuevos profesionales, con solo tres años de creación. Es uno de los proyectos más ambicioso de nuestro país, que en estos momentos cuenta con alrededor de 6 mil estudiantes número debe crecer a 10 mil, a los que se incorporan casi mil profesores; lo que demuestra la necesidad de creación de un sistema automatizado para la gestión académica en la Universidad.

En un centro de estudios es de vital importancia para su buen funcionamiento la eficiencia de la gestión académica. Los procesos que envuelve la gestión académica deben adaptarse a las condiciones actuales en que se desarrolla la labor formativa, estos involucran gran cantidad de actores, los cuales deben verse implicados en la gestión automatizada directamente con el objetivo de agilizar el proceso docente ya que el desenvolvimiento de este constituye el objetivo principal de cualquier centro y para lograrlo es necesario trazar metas y estrategias para planificar, organizar y controlar todos los medios necesarios .

La gestión académica en la UCI, constituye un arduo trabajo debido al alto número de implicados, de ahí que surja la necesidad de emplear sistemas informáticos que automaticen todo este proceso y garantice la eficiencia de la gestión académica en el centro.

En la Universidad han sido utilizados varios Sistemas para la Gestión Académica entre los que se encuentra:

**GESTACAD:** Sistema de Gestión Académica

Este sistema surgió con la necesidad de desarrollar un software que permitiera automatizar la gestión académica de las universidades en Cuba, el cual gestiona una parte de la información académica de los estudiantes universitarios y la información de los profesores que forman parte del proceso docente educativo.

Este sistema fue implementado por el Departamento de Informática de la Universidad de Matanzas "Camilo Cienfuegos", el cual permite la actualización de las informaciones y la obtención de resultados propios del trabajo de las áreas implicadas en la gestión académica.

### **1.3 Sistema de Gestión Académica. Akademos**

En la actualidad los sistemas de gestión académica limitan la interacción con la aplicación al personal de secretaría, excluyendo a otros involucrados como Directivos, profesores, etc. Es por esta razón que existen varios sistemas de gestión académica que lejos de ayudar a aliviar la labor del personal de secretaria, de directivos y docentes hacen el proceso más complejo. Entre las dificultades más frecuentes está no poder adaptarse a los cambios que ocurran y al no permitir la participación de todos los involucrados en la gestión académica, hacen que la información no se introduzca al sistema por los que la producen.

Para dar solución a esta problemática, la Dirección de Informatización de la Universidad de las Ciencias Informáticas (UCI) ha desarrollado Akademos, un sistema automatizado para la gestión académica. Con el cual se han obtenido buenos resultados ya que ha logrado la participación de directivos y docentes agilizando así los procesos de la gestión académica.

Akademos se ha desarrollado como una aplicación Web haciendo uso de la tecnología Microsoft.NET y los XML WebServices logrando la creación de sistemas que permitan informatizar las distintas áreas de esta Universidad. Consta de siete módulos, los cuales abarcan todos los procesos involucrados en la gestión docente. Estos módulos interactúan entre sí para llevar a cabo cada una de las tareas que automatiza el sistema [1].

#### **1.3.1 Módulos de Akademos**

##### **Módulo de plan de estudio**

El plan de estudios es la entidad fundamental del sistema. Este se define como una secuencia de niveles, divididos en momentos que los estudiantes deben vencer después de concluidas las asignaturas, estas son ubicadas en un orden de precedencia dado en los distintos momentos del plan de estudio. Este módulo también permite definir de los diferentes perfiles y disciplinas en las que se agrupan las asignaturas.

### **Módulo de matrícula**

Son los estudiantes la razón principal de cualquier centro de estudios. El objetivo fundamental del módulo de matrícula es el control de los datos de los estudiantes así como la gestión de los movimientos a que estos son sometidos en su paso por la universidad. Se define como movimiento cualquier acción que se realiza sobre un estudiante que provoca un cambio de su estado dentro de la universidad, y generan un conjunto de información que debe quedar registrada en su expediente.

### **Módulo de expediente**

En un centro de estudios, en su trayectoria por el mismo el estudiante deja un conjunto de información relacionada con su desempeño académico y otros aspectos sobre su comportamiento, lo cual conforma el expediente del estudiante. El conjunto de información que van dejando las generaciones de estudiantes debe ser almacenada con el objetivo de sacar conclusiones históricas que contribuyan al mejoramiento de la actividad del centro.

### **Módulo de registro del profesor**

Este módulo es el encargado de permitir el control del progreso de un período académico. Este módulo está muy relacionado con el de plan de estudio, ya que, de él toma la información que necesita referente a las evaluaciones y formas de calificación de las asignaturas.

Una vez, que el sistema cuente con esta información, es posible que puedan elaborarse una gran cantidad de reportes para determinar el desempeño de un estudiante, un grupo o una escuela completa.

## **Módulo de Planificación**

Este módulo es el encargado de facilitar la planificación de un periodo docente. Específicamente da la posibilidad de planificar tanto la carga docente de los profesores como la gestión de los horarios.

Este módulo permitirá a los jefes de departamento, decanos docentes de facultades entre otros mantener un control de la plantilla de profesores y realizar un balance de la carga docente de estos, permite la validación de los horarios atendiendo a varios criterios, entre ellos el solapamiento de profesores y locales. También, es un mecanismo eficaz de publicación de los mismos.

## **Módulo de reportes**

Generar reportes es quizás la tarea fundamental de todo sistema de gestión de datos. Akademos proporciona este módulo de reportes que cuenta con diferentes herramientas, las cuales permiten el diseño y publicación de nuevos reportes. Con esta funcionalidad el usuario puede diseñar reportes que relacionen varios aspectos de los estudiantes, convirtiéndose en una poderosa herramienta para el estudio de la comunidad estudiantil.

## **Seguridad**

Akademos tiene tres niveles de seguridad:

- Seguridad de los datos:
  - Para el almacenamiento de los datos se ha optado por la utilización de SQL Server 2000, el cual es capaz de garantizar la seguridad tanto lógica como de integridad de los datos almacenados en la base de datos.
- Seguridad de los recursos lógicos:
  - Se deja al sistema operativo, Windows en este caso, la responsabilidad de restringir el acceso a cada uno de los recursos lógicos de la aplicación como páginas Web, Servicios Web, archivos XML, etc.

- Seguridad de los elementos específicos del negocio que automatiza la aplicación:
- Akademos implementa varios niveles de acceso para restringir las acciones que puede realizar un usuario determinado.

#### **1.4 Gestión de Profesores.**

Los profesores son el eslabón principal con que cuenta un centro de estudios para desarrollar con éxito su labor formativa ya que ocupan un papel importante en el sistema de influencias que propician la formación del estudiante. La Universidad actualmente no cuenta con un sistema que automatice el trabajo con los profesores, de ahí la importancia de crear este Módulo de Gestión de Profesores que permita a los jefes de departamento, decanos docentes de facultades, entre otros, mantener un control de la plantilla de profesores y realizar un balance de la carga docente de estos así como la gestión de toda la información referente al mismo, de forma tal que se puedan utilizar de forma óptima a los profesores.

Actualmente los profesores no tienen acceso a conocer directamente su planificación y otras actividades a cumplir durante el curso, sino que deben ser solicitadas en la facultad. Resulta importante que los profesores puedan interactuar con el sistema de gestión de profesores y conocer en el momento deseado tanto su planificación docente como otros datos de interés como profesor del centro.

También otros Directivos del centro pueden acceder en un momento deseado a parte de la información relacionada con los profesores de la Universidad. La UCI cuenta actualmente con un alto número de profesores por lo que se vuelve engorrosa todo el proceso de gestión y planificación de los mismos, sobre todo en el caso del plan de trabajo del profesor ya que es mucha información almacenada en papel, la cual debe estar libre de errores y puede ser automatizada.

#### **1.5 Aplicaciones Web**

En la actualidad con la introducción de Internet y del Web en concreto, se han abierto infinidad de posibilidades en cuanto al acceso a la información desde casi

cualquier sitio. Esto representa un desafío a los desarrolladores de aplicaciones, ya que los avances en tecnología demandan cada vez aplicaciones más rápidas, ligeras y robustas que permitan utilizar el Web.

Las aplicaciones web constituyen formularios pre-formateados que envían mensajes instantáneos a usuarios o a un servidor. Están diseñadas para interactuar con bases de datos con el fin de recoger, almacenar, organizar y distribuir información [2].

## **1.6 Servicios Web**

“Los Servicios Web son una tecnología emergente impulsada por el deseo de exponer de forma segura la lógica de negocios en Internet. A través de los Servicios Web las empresas pueden encapsular sus procesos de negocios existentes, publicarlos como servicios, suscribirse a otros servicios e intercambiar información entre empresas” [3].

Los servicios web son un modo normalizado de integrar aplicaciones basadas en la web usando las normas abiertas XML, SOAP, WSDL y UDDI sobre un protocolo de Internet.

Los servicios web permiten que varias aplicaciones de diferentes fuentes se comuniquen entre sí sin tener que recurrir a la difícil codificación personalizada y, al ser todas las comunicaciones en XML, los servicios Web no están relacionados a ningún sistema operativo ni lenguaje de programación.

Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar [4].

### **Entre los beneficios principales de los Servicios Web se encuentran:**

- Pueden ser brindados y pagados en línea y pueden ser accedidos desde cualquier plataforma. Permiten el encapsulamiento. Los componentes pueden



ser aislados de forma tal que queda expuesta sólo al nivel de la capa de negocios lo que permite sistemas más flexibles y estables.

- Pueden ser completamente descentralizados y distribuidos sobre Internet y accedidos a través de una gran variedad de dispositivos.
- Se basan en especificaciones estándar para el intercambio de datos, mensajería, búsqueda, descripción de la interface y coordinación de los procesos.
- Mayor agilidad y flexibilidad debido a una mejor integración con los sistemas existentes.

### **1.7 Microsoft .NET**

Microsoft .NET es una plataforma de desarrollo integrado para arquitecturas de aplicaciones web, de escritorio o distribuidas. Permite crear y mantener aplicaciones a gran escala a los desarrolladores con un enfoque sencillo y sólido, de forma fácil y sin complicaciones. Además permite integrar aplicaciones escritas en distintos lenguajes de programación y ofrece un entorno seguro para su ejecución y distribución.

Esta tecnología .NET permite la creación y uso de aplicaciones, procesos y sitios Web basados en XML, como servicios que comparten y combinan información y funcionalidad. Esto lo permite por diseño, sobre cualquier plataforma o dispositivo inteligente, para proveer soluciones para organizaciones e individuos.

Se define como:

“El conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando estos últimos años con el objetivo de mejorar tanto su sistema operativo como su modelo de componentes (COM) para obtener una plataforma con la que sea sencillo el desarrollo de software en forma de servicios Web” [5].

Desarrollado con base en los estándares de Servicios Web XML, .NET permite que los sistemas y aplicaciones, ya sea nuevos o existentes, conecten sus datos y transacciones independientemente del sistema operativo, tipo de computadora o dispositivo móvil que se utilice, o del lenguaje de programación empleados para crearlo. Además ofrece la capacidad de desarrollar, implementar, administrar y utilizar soluciones conectadas a través de Servicios Web XML, de manera rápida, económica y segura. Estas soluciones permiten una integración más rápida y ágil entre las empresas y el acceso a información a cualquier hora, en cualquier lugar y a través de cualquier dispositivo.

El elemento más importante de la plataforma .NET es NET framework, una suite de herramientas que incluye COM+ (Component Object Model), un entorno de ejecución común, un compilador JIT (just-in-time) y un conjunto de librerías de sistema que dan acceso a un amplio conjunto de servicios.

Otro elemento fundamental es el CLR (Common Language Runtime), que es una aplicación similar a una máquina virtual que es el encargado de gestionar la ejecución de las aplicaciones para ella escritas. A estas aplicaciones les ofrece numerosos servicios que facilitan su desarrollo y mantenimiento y favorecen su fiabilidad y seguridad.

Entre ellos los principales son:

- Modelo de programación consistente y sencillo, completamente orientado a objetos.
- Eliminación del temido problema de compatibilidad entre DLLs, conocido como "infierno de las DLLs".
- Ejecución multiplataforma.
- Ejecución multilenguaje.
- Recolección de basura.

- Gestión del acceso a objetos remotos que permite el desarrollo de aplicaciones distribuidas de manera transparente a la ubicación real de cada uno de los objetos utilizados en las mismas.
- Seguridad avanzada, hasta el punto de que es posible limitar los permisos de ejecución del código en función de su procedencia (Internet, red local, CD-ROM, entre otros), el usuario que lo ejecuta o la empresa que lo creó.
- Interoperabilidad con código preexistente, de manera que es posible utilizar con facilidad cualquier librería de funciones u objetos COM y COM+ creados con anterioridad a la aparición de la plataforma .NET.
- Adecuación automática de la eficiencia de las aplicaciones a las características concretas de cada máquina donde se vaya a ejecutar [6].

Para el desarrollo y ejecución de aplicaciones en este nuevo entorno tecnológico Microsoft proporciona el conjunto de herramientas conocido .NET Framework SDK e incluye compiladores de lenguajes como C#, Visual Basic.NET, Visual C++ .NET y JScript.NET específicamente diseñados para crear aplicaciones para él. Ver Figura 1.1.

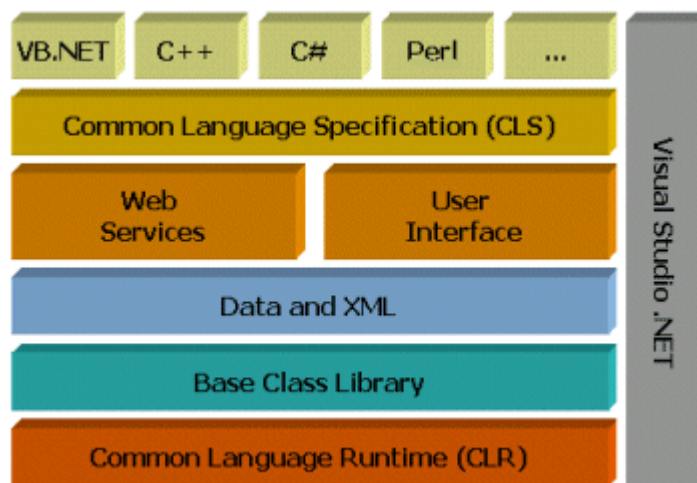


Figura 1.1. Componentes de Microsoft .NET. Framework.

### 1.7.1 Visual Studio.NET

Visual Studio .NET es una completa herramienta para generar e integrar con rapidez aplicaciones y servicios Web XML, lo que mejora notablemente la productividad del programador y abre las puertas a nuevas oportunidades empresariales. Es una de las herramientas de desarrollo más completa creadas hasta el día de hoy, integra en un mismo entorno múltiples lenguajes y todos los recursos necesarios para la construcción rápida de aplicaciones Windows, componentes, aplicaciones para la Web y servicios Web [7].

Ofrece una funcionalidad superior para optimizar procesos empresariales, abriendo así las puertas a nuevas oportunidades para las empresas. Visual Studio .NET también mejora notablemente la productividad del programador, permitiendo generar soluciones para una amplísima gama de clientes, incluidas aplicaciones para el Web, Windows y dispositivos de cliente ligero.

#### Principales ventajas:

- Visual Studio .NET está basado en la más reciente plataforma de servidor de Microsoft Windows®, lo que incorpora escalabilidad, confiabilidad y seguridad a las aplicaciones.
- El entorno de Visual Studio.NET, es abierto y personalizable, en el que pueden integrarse otros lenguajes y herramientas.
- Puede generar aplicaciones controladas por datos de alto rendimiento. También puede utilizar herramientas ADO.NET incorporadas para una gran variedad de bases de datos, incluidos Microsoft SQL Server™, Oracle o cualquier origen de datos basado en XML. Gracias a su compatibilidad intrínseca con XML, ADO.NET permite a los programadores compartir datos a través de diferentes plataformas.
- Permite a los programadores crear e implementar lógica empresarial importante basada en un servidor.
- Pueden componer visualmente componentes de nivel medio utilizando el Diseñador de componentes visual, que permite a los programadores arrastrar

objetos no visuales —como colas de mensajes, temporizadores y registros de eventos— hasta una superficie de diseño desde el Explorador de servidores, una nueva ventana de herramientas que descubre automáticamente todos los recursos necesarios basados en servidor.

Existen cuatro lenguajes que forman parte de Visual Studio.NET: Visual C#, Visual Basic.NET, Visual C++.NET y JScript.NET. Usando cualquiera de estos lenguajes, se pueden crear aplicaciones basadas en formularios Windows y formularios y servicios Web.

### 1.7.2 ASP .NET.

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

- **Mejor rendimiento.** ASP.NET es un código de Common Language Runtime compilado que se ejecuta en el servidor. A diferencia de sus predecesores, ASP.NET puede aprovechar las ventajas del enlace anticipado, la compilación just-in-time, la optimización nativa y los servicios de caché desde el primer momento. Esto supone un incremento espectacular del rendimiento antes de siquiera escribir una línea de código.
- **Compatibilidad con herramientas de primer nivel.** El marco de trabajo de ASP.NET se complementa con un diseñador y una caja de herramientas muy completos en el entorno integrado de programación (Integrated Development Environment, IDE) de Visual Studio. La edición WYSIWYG, los controles de servidor de arrastrar y colocar y la implementación automática son sólo algunas de las características que proporciona esta eficaz herramienta.
- **Eficacia y flexibilidad.** Debido a que ASP.NET se basa en Common Language Runtime, la eficacia y la flexibilidad de toda esa plataforma se

encuentra disponible para los programadores de aplicaciones Web. La biblioteca de clases de .NET Framework, la Mensajería y las soluciones de Acceso a datos se encuentran accesibles desde el Web de manera uniforme. ASP.NET es también independiente del lenguaje, por lo que puede elegir el lenguaje que mejor se adapte a la aplicación o dividir la aplicación en varios lenguajes. Además, la interoperabilidad de Common Language Runtime garantiza que la inversión existente en programación basada en COM se conserva al migrar a ASP.NET.

- **Simplicidad.** ASP.NET facilita la realización de tareas comunes, desde el sencillo envío de formularios y la autenticación del cliente hasta la implementación y la configuración de sitios. Por ejemplo, el marco de trabajo de página de ASP.NET permite generar interfaces de usuario, que separan claramente la lógica de aplicación del código de presentación, y controlar eventos en un sencillo modelo de procesamiento de formularios de tipo Visual Basic. Además, Common Language Runtime simplifica la programación, con servicios de código administrado como el recuento de referencia automático y el recolector de elementos no utilizados.
- **Facilidad de uso.** ASP.NET emplea un sistema de configuración jerárquico, basado en texto, que simplifica la aplicación de la configuración al entorno de servidor y las aplicaciones Web. Debido a que la información de configuración se almacena como texto sin formato, se puede aplicar la nueva configuración sin la ayuda de herramientas de administración local. Esta filosofía de "administración local cero" se extiende asimismo a la implementación de las aplicaciones ASP.NET Framework. Una aplicación ASP.NET Framework se implementa en un servidor sencillamente mediante la copia de los archivos necesarios al servidor. No se requiere el reinicio del servidor, ni siquiera para implementar o reemplazar el código compilado en ejecución.
- **Escalabilidad y disponibilidad.** ASP.NET se ha diseñado teniendo en cuenta la escalabilidad, con características diseñadas específicamente a

medida, con el fin de mejorar el rendimiento en entornos agrupados y de múltiples procesadores. Además, el motor de tiempo de ejecución de ASP.NET controla y administra los procesos de cerca, por lo que si uno no se comporta adecuadamente (filtraciones, bloqueos), se puede crear un proceso nuevo en su lugar, lo que ayuda a mantener la aplicación disponible constantemente para controlar solicitudes.

- **Posibilidad de personalización y extensibilidad.** ASP.NET presenta una arquitectura bien diseñada que permite a los programadores insertar su código en el nivel adecuado. De hecho, es posible extender o reemplazar cualquier subcomponente del motor de tiempo de ejecución de ASP.NET con su propio componente escrito personalizado. La implementación de la autenticación personalizada o de los servicios de estado nunca ha sido más fácil.
- **Seguridad.** Con la autenticación de Windows integrada y la configuración por aplicación, se puede tener la completa seguridad de que las aplicaciones están a salvo [8].

### 1.7.3 Arquitectura basada en componentes.

Diseñar aplicaciones empresariales complejas compuestas por gran cantidad de componentes requiere de la separación de estos de la solución en capas, de forma tal que exista alta cohesión entre los componentes de cada capa y a la vez cada capa esté desacoplada de la otra. Por lo general este tipo de aplicación requiere de la exposición de ciertas funcionalidades como servicios que puedan ser usadas por otras aplicaciones, de la misma forma esta aplicación puede consumir servicios expuestos por otra aplicaciones.

Es necesario, entonces, determinar como dividir una aplicación orientada a servicios y ubicar los componentes que van en cada capa, por lo tanto lo primero es hacer particiones lógicas a la aplicación de que separen la presentación, de la lógica del negocio, de los datos.

Es posible crear aplicaciones con tres capas lógicas pero que estén compuestas de un solo ejecutable o en un mismo ordenador, no obstante los mejores resultados se obtienen cuando se usan para crear aplicaciones empresariales distribuidas.

Microsoft presenta su estrategia para desarrollar aplicaciones distribuidas con esta arquitectura. El advenimiento de Microsoft .NET junto a la ya madura arquitectura de COM+, intentan imponerse como el estándar para desarrollar aplicaciones empresariales distribuidas basadas en componentes y orientada a los servicios.

Siguiendo con el modelo que propone Microsoft .NET, básicamente las tres capas son:

**Presentación.** La capa de presentación contiene los elementos de interfaz de usuario. Típicamente esto incluye Windows Form y tecnologías ASP NET.

**Negocio.** La capa de negocio implementa las funcionalidades del negocio de la aplicación. Esta capa esta compuesta por componentes que pueden ser implementados usando cualquiera de los lenguajes disponibles en .Net, Las prestaciones de estos componentes pueden incrementarse con el uso de tecnologías como *Microsoft .Net* Componentes Empresariales (COM+) para soluciones de componentes distribuidos y el uso de Microsoft Biztalk Server para la orquestación de Flujos de Trabajos Empresariales.

**Datos:** La capa de datos provee el acceso a sistemas externos como las bases de datos. .Net involucra en esta capa primeramente a ADO. NET. Sin embargo no es poco frecuente usar algunas de las capacidades de .NET XML en esta capa.

A continuación se muestra una lista de los componentes de software mas comúnmente encontrados en soluciones distribuidas [9].



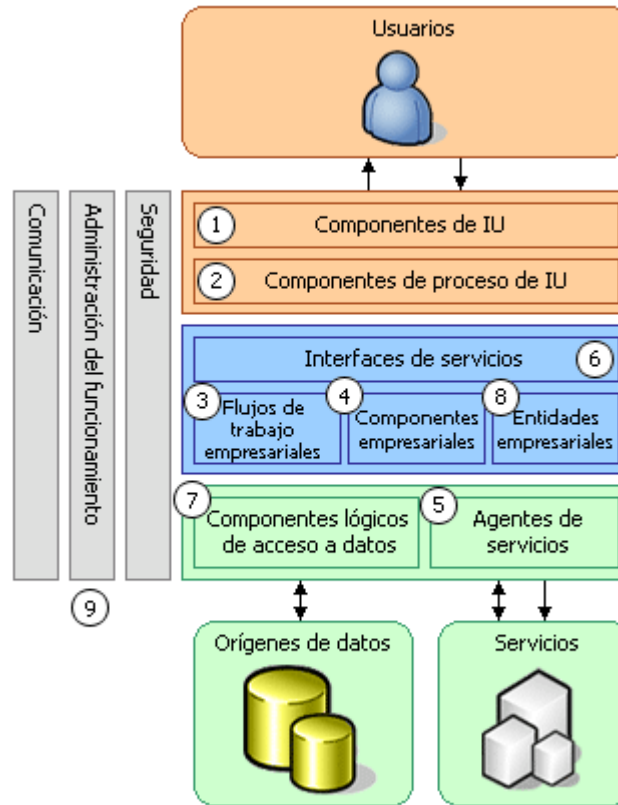


Figura 1.2. Capas lógicas de una aplicación .NET.

1. **Componentes de interfaz de usuario (IU).** La mayor parte de las soluciones necesitan ofrecer al usuario un modo de interactuar con la aplicación. Las interfaces de usuario se implementan utilizando formularios de Windows Forms, páginas Microsoft ASP.NET, controles u otro tipo de tecnología que permita procesar y dar formato a los datos de los usuarios, así como adquirir y validar los datos entrantes procedentes de éstos.
2. **Componentes de proceso de usuario.** En un gran número de casos, la interacción del usuario con el sistema se realiza de acuerdo a un proceso predecible. Para facilitar la sincronización y organización de estas interacciones con el usuario, resulta útil utilizar componentes de proceso de usuario individuales. De este modo, el flujo del proceso y la lógica de administración de estado no se incluye en el código de los elementos de la interfaz de usuario, por lo que varias interfaces podrán utilizar el mismo "motor" de interacción básica.

3. **Flujos de trabajo empresariales.** Una vez que el proceso de usuario ha recopilado los datos necesarios, éstos se pueden utilizar para realizar un proceso empresarial. Gran parte de los procesos empresariales conllevan la realización de varios pasos, los cuales se deben organizar y llevar a cabo en un orden determinado. Los flujos de trabajo empresariales definen y coordinan los procesos empresariales de varios pasos de ejecución larga y se pueden implementar utilizando herramientas de administración de procesos empresariales.
4. **Componentes empresariales.** Independientemente de si el proceso empresarial consta de un único paso o de un flujo de trabajo organizado, la aplicación requerirá probablemente el uso de componentes que implementen reglas empresariales y realicen tareas empresariales. Los componentes empresariales implementan la lógica empresarial de la aplicación.
5. **Agentes de servicios.** Cuando un componente empresarial requiere el uso de la funcionalidad proporcionada por un servicio externo, tal vez sea necesario hacer uso de código para administrar la semántica de la comunicación con dicho servicio. Se podría utilizar un agente de servicios estos permiten aislar las idiosincrasias de las llamadas a varios servicios desde la aplicación y pueden proporcionar servicios adicionales, como la asignación básica del formato de los datos que expone el servicio al formato que requiere la aplicación.
6. **Interfaces de servicios.** Para exponer lógica empresarial como un servicio, es necesario crear interfaces de servicios que admitan los contratos de comunicación (comunicación basada en mensajes, formatos, protocolos, seguridad y excepciones, entre otros) que requieren los clientes. Las interfaces de servicios también se denominan *fachadas empresariales*.
7. **Componentes lógicos de acceso a datos.** La mayoría de las aplicaciones y servicios necesitan obtener acceso a un almacén de datos en un momento determinado del proceso empresarial. Por tanto, es razonable abstraer la lógica necesaria para obtener acceso a los datos en una capa independiente de

componentes lógicos de acceso a datos, ya que de este modo se centraliza la funcionalidad de acceso a datos y se facilita la configuración y el mantenimiento de la misma.

8. **Componentes de entidad empresarial.** La mayoría de las aplicaciones requieren el paso de datos entre distintos componentes. Los datos se utilizan para representar entidades empresariales del mundo real, como productos o pedidos. Las entidades empresariales que se utilizan de forma interna en la aplicación suelen ser estructuras de datos, como conjuntos de datos, DataReader o secuencias de lenguaje de marcado extensible (XML), aunque también se pueden implementar utilizando clases orientadas a objetos personalizadas que representan entidades del mundo real necesarias para la aplicación, como productos o pedidos.
9. **Componentes de seguridad, administración operativa y comunicación.** La aplicación probablemente utilice también componentes para realizar la administración de excepciones, autorizar a los usuarios a que realicen tareas determinadas y comunicarse con otros servicios y aplicaciones

### **1.8 Modelo Cliente-Servidor**

La programación cliente-servidor es utilizada cuando se quiere realizar aplicaciones que utilicen redes y que comuniquen entre sí a varios ordenadores. Consiste en que el programa se divide en dos partes:

La parte Cliente, que reside en el equipo donde está el usuario y se encarga de la interacción con éste y la parte Servidor, que reside en un ordenador conectado a la red permanentemente y se encarga de manipular los datos.

#### **Funciones Comunes del Cliente:**

- Mantener y procesar todo el dialogo con el usuario.
- Manejo de pantallas.
- Menús e interpretación de comandos.

- Entrada de datos y validación.
- Procesamiento de ayudas.
- Recuperación de errores.

### **Funciones Comunes del Servidor:**

- Acceso, almacenamiento y organización de datos.
- Actualización de datos almacenados.
- Administración de recursos compartidos.
- Ejecución de toda la lógica para procesar una transacción.
- Procesamiento común de elementos del servidor (Datos, capacidad de CPU, almacenamiento en disco, capacidad de impresión, manejo de memoria y comunicación).

Ambas partes de la aplicación se comunican entre sí utilizando algún protocolo de red TCP/IP. La justificación de este paradigma es la minimización del tráfico de red, sobre todo para evitar ralentizaciones y economizar el ancho de banda. Esta tecnología denominada Cliente -Servidor es utilizada por todas las aplicaciones de Internet/Intranet. Un único servidor típicamente sirve a una multitud de clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente.

### **Entre las principales ventajas de este modelo se encuentran:**

- Con el uso de este esquema, se reducen los costos de producción de software y se disminuyen los tiempos requeridos.
- Reduce el costo del hardware requerido, llevando las aplicaciones a plataformas más baratas, aprovechando el poder de cómputo de los diferentes elementos de la red, y facilitando la interacción entre las distintas aplicaciones de la organización.

- Contribuye a una disminución de los costos de entrenamiento de personal, pues favorecen la construcción de interfaces gráficas interactivas, las cuales son más intuitivas y fáciles de usar por el usuario final.
- Facilita el suministro de información a los usuarios.
- Permite llevar más fácilmente la información a donde se necesita, contribuye a aumentar su precisión pues se puede obtener de la fuente (el servidor) y no de una copia en papel o en medio magnético.
- Favorece la adaptación a cambios en la tecnología, pues facilita la migración de las aplicaciones a otras plataformas y, al aislar claramente las diferentes funciones de una aplicación, hace más fácil incorporar nuevas tecnologías en ésta [11].

### **1.9 Sistemas de Gestores de Bases de Datos**

Un sistema gestor de base de datos (SBGD) es un conjunto de programas que permiten crear y mantener una Base de Datos, además se puede definir también como “el conjunto de herramientas que suministra a todos, ya sea un administrador, analista, programador, usuario, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base de datos, manteniendo la seguridad, integridad y confidencialidad de los mismos”[12].

#### **Características de un Sistema Gestor de Base de Datos:**

- Los datos están organizados de forma independiente de las aplicaciones que los vayan a utilizar (independencia lógica) y de los ficheros en que vayan a almacenarse (independencia física).
- Tanto los usuarios como las aplicaciones pueden tener acceso a los datos a través del uso de los lenguajes de consulta.
- La gestión de los datos ocurre de forma centralizada e independiente de las aplicaciones.
- Consistencia e integridad de los datos.

- Fiabilidad (protección frente a fallos) y seguridad (no todos los datos deben ser accesibles a todos los usuarios) [13].

Ejemplos de SGBD son Oracle y SQL Server de Microsoft. IBM DB2, Interbase

### 1.9.1 SQL Server

En la actualidad se requiere de aplicaciones y bases de datos empresariales que puedan acumular la información recolectada por los sistemas de negocios, que soporten a una cantidad cada vez mayor de usuarios simultáneos, además de procesar y analizar eficientemente cantidades masivas de datos en formas cada vez más complejas. SQL Server 2000 Enterprise Edition ofrece una plataforma de datos escalable con herramientas para apoyar a las empresas en el análisis de grandes cantidades de datos y poder tomar así decisiones informadas.

SQL Server es el servidor de bases de datos de Microsoft, seguro, robusto y con las más avanzadas prestaciones: transacciones, procedimientos almacenados, triggers.

“Un servidor de bases de datos es un programa que almacena datos estructurados en forma de tablas relacionales, atiende un puerto TCP/IP a través del cual acepta conexiones de clientes autenticados, admite comandos en lenguaje SQL, y devuelve al cliente a través de la red los datos resultantes del procesamiento de los comandos SQL”[14].

El SQL posee dos características muy apreciadas: que presenta una potencia y versatilidad notable y su accesibilidad de aprendizaje. Trabaja con estructura cliente/servidor sobre una red de ordenadores. El ordenador cliente es el que inicia la consulta; el ordenador servidor es que atiende esa consulta. El cliente utiliza toda su capacidad de proceso para trabajar; se limita a solicitar datos al ordenador servidor, sin depender para nada más del exterior. Estas peticiones y las respuestas son transferencias de textos que cada ordenador cliente se encarga de sacar por pantalla, presentar en informes tabulados, imprimir, guardar, etc., dejando el servidor libre.

El SQL permite:

- Definir una base de datos mediante tablas
- Almacenar información en tablas.
- Seleccionar la información que sea necesaria de la base de datos.
- Realizar cambios en la información y estructura de los datos.
- Combinar y calcular datos para conseguir la información necesaria.
- SQL es el lenguaje de comunicación entre el programa cliente y programa servidor.

### **Conclusiones**

En este capítulo se mostró la información que se maneja en los procesos de la Gestión Académica y la labor que realizan todos los implicados en el proceso de la Gestión Académica en el centro, específicamente en la gestión y planificación de los profesores lo que demuestra la necesidad del tratamiento informatizado de este proceso.

Por otra parte, se realizó un estudio de las aplicaciones Web, de las tecnologías para su desarrollo, así como de los gestores de bases de datos, decidiéndose el uso de ASP .NET, C# como lenguaje de programación y SQL Server 2000 como gestor de bases de datos por las facilidades que ofrecen.

## Capítulo 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

### ***Introducción***

En este capítulo se aborda lo referente a la modelación del negocio y del sistema, de la solución propuesta: Akademos, las descripciones de los casos de uso del negocio y del sistema aparecen en formatos expandidos, y los últimos muestran las interfaces correspondientes, además se describe el estado actual del negocio y la solución propuesta.

Además se muestran los requisitos funcionales y no funcionales que presenta el sistema que se propone así como los beneficios tangibles e intangibles y el análisis costo-beneficios, lo que permite hacer una concepción general del sistema, además identificar mediante el Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el sistema, y las diferentes acciones que realizan.

### ***2.1 Estado actual del negocio***

En la Universidad de las Ciencias Informáticas (UCI) no se utiliza actualmente ningún sistema automatizado que permita todas las actividades relacionadas con la gestión académica, específicamente el proceso de Gestión y Planificación de los profesores, de forma tal que se facilite el trabajo a los involucrados en esta labor, que constituye una de las tareas más prioritarias de cualquier centro de estudios.

Este proceso se inicia cuando el profesor se presenta en la Dirección de Formación, donde el Director es el encargado de realizar la entrevista previa al ingreso del mismo en el centro con el objetivo de recoger información preliminar del profesor y ubicarlo así en un Departamento Central. Para esto deben cumplir con varios requisitos fundamentales que permitan estar en plenas condiciones para entrar en la Universidad. Luego de terminada la entrevista y la recogida de los datos correspondientes se envía al profesor al Departamento Central al que fue asignado, donde se le hará la ubicación en una Facultad y se le hace una propuesta de la asignatura a impartir en el curso que luego debe ser aceptada por el Jefe de Departamento de la Facultad culminando así el proceso de asignación del profesor.



Actualmente en la Universidad todo este proceso se realiza de forma manual lo que impide una correcta manipulación de la información relacionada con los profesores. Luego que el profesor se presenta a la facultad solicita su plan de trabajo al Jefe de Departamento, este inicialmente acepta la propuesta de la asignatura que debe impartir el profesor la cual debe pertenecer a la disciplina indicada por el Director de Formación, y se le asignan los grupos a los que debe impartir clases. Todos estos datos son registrados por la Facultad.

Después de realizado este proceso el Jefe de Colectivo de la asignatura le confecciona su plan de trabajo donde se abordan todas las diferentes actividades que deben ser cumplidas por el profesor en el transcurso del curso. Esto conlleva a una gran carga de trabajo a los implicados en esta labor además de carecer de una fuente centralizada, que almacene los datos de todos los profesores del centro.

Entre las diferentes actividades que puede realizar el profesor a lo largo del curso se encuentran:

- Ejecutar las acciones previstas en los Proyectos Educativos
- Contribuir a la formación de valores.
- Trabajar en la orientación política-ideológica de los estudiantes.
- Participar en actividades políticas convocadas por la brigada, facultad o la UCI; sindicales, de la FEU y demás.
- Participación en grupos de investigación – producción.
- Presentar trabajos en el Forum de Ciencia y Técnica a nivel de Centro.
- Participar en actividades de orientación vocacional y profesional.
- Participar en los festivales culturales y deportivos, directa o indirectamente apoyando a la representación de la Facultad.
- Participar en las actividades metodológicas del departamento y colectivo de asignaturas, profundizando en las didácticas y especificidades de las mismas.

- Cumplir con las tareas encomendadas por la dirección de la facultad o la universidad.
- Participar en la creación de grupos científicos.
- Estar involucrado en un proyecto investigativo.

El Director de Formación y el Jefe de Departamento de la facultad pueden tener acceso a toda la información referente a este proceso, también el propio profesor y otros Directivos del centro podrán tener acceso a una parte de la información docente de los profesores así como su planificación y el plan de trabajo del mismo.

Es por estas razones que se resolvió efectuar la implementación de un Sistema de Gestión de Profesores automatizado en la Universidad de Ciencias Informáticas, que permita aliviar la labor de sus implicados de manera eficiente, sustituyendo el trabajo manual por el automatizado. Además brinde información referente al profesor en el momento deseado tanto a los propios profesores como a otros Directivos del centro permitiendo así a los usuarios una mejor utilización de la información que se almacena en este sistema.

## ***2.2 Reglas del negocio a considerar***

Todos los procesos del negocio presentan reglas que deben tenerse en cuenta a la hora de la automatización de los mismos, de forma tal que el sistema a desarrollar cumpla con todas las restricciones necesarias para este.

Las reglas del negocio a tomarse en cuenta para la elaboración del sistema son las siguientes:

- Un profesor para ingresar en la Universidad debe ser entrevistado inicialmente con el propósito de conocer el estado del mismo y si cuenta con las características necesarias para entrar en el centro, además de la recogida de todos sus datos los cuales son almacenados con posterioridad.
- Un profesor recibe al inicio del curso su ubicación y se le confecciona el plan de trabajo.

- La información registrada sobre el cumplimiento de las actividades expuestas en el plan de trabajo de los profesores puede ser accedida por el propio profesor y otros Directivos del centro.

### 2.3 Modelo del Negocio

El conocimiento de las características y actividades que se llevan a cabo en el contexto que se desea automatizar, es un aspecto importante para la obtención de los requerimientos que debe cumplir el sistema a desarrollar. Con este objetivo se realiza el modelo del negocio, en el que se modelan los trabajadores y clientes del negocio así como su interacción con los procesos y sus objetos. Este modelo está formado por el modelo de casos de uso del negocio y el modelo de objetos del negocio [14].

#### 2.3.1 Modelo de casos de uso del negocio

El modelo de casos de uso representa los procesos del negocio en forma de casos de uso del negocio y actores del negocio, representando en estos últimos a los clientes. Dicho modelo se representa mediante diagramas de casos de uso que muestran cómo los actores del negocio y los casos de uso del negocio se encuentran relacionados.

#### 2.3.2 Definición de actores y trabajadores del negocio

Los actores del negocio representan individuos o grupos con los que el negocio interactúa obteniéndose un resultado. Descripción de los actores del negocio.

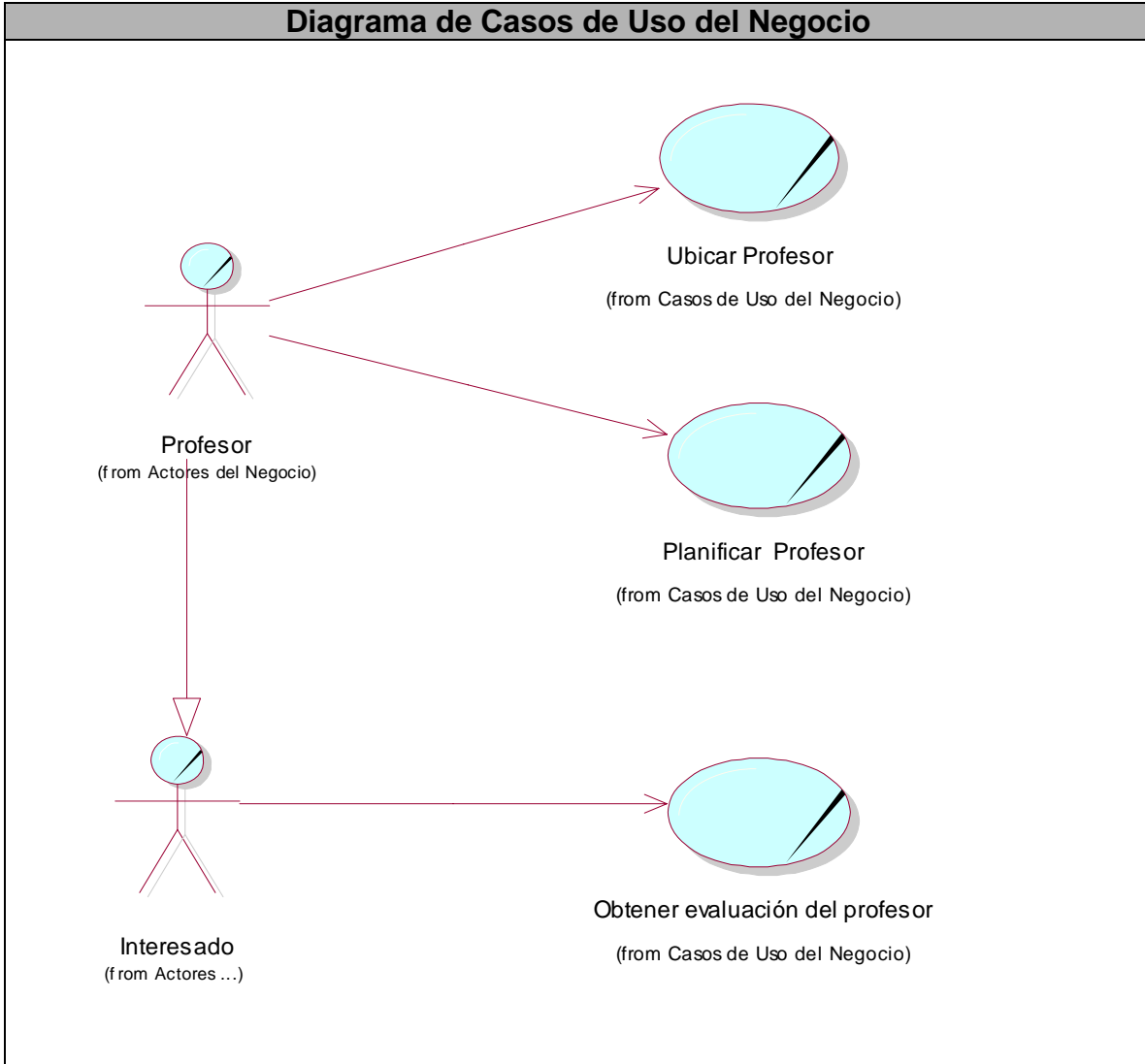
Actores del Negocio	Justificación
Profesor	Es quien inicia las diferentes acciones a realizar ya que solicita la entrevista para ser aceptado como profesor y se le ubique en un Departamento Central y en una Facultad y después asignarle la asignatura a impartir y conformar su plan de trabajo. Además el profesor puede solicitar ver su planificación.

Interesado	Aquel que pueda estar interesado en conocer la evaluación final del profesor que puede ser el propio profesor u otros Directivos del Centro además de la planificación actual del profesor.
------------	---

Los trabajadores del negocio son los que actúan en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Descripción de los trabajadores del negocio.

Trabajadores del negocio	Justificación
Director de Formación	Es aquel que está encargado de ubicar al profesor en un Departamento Central.
Jefe de Departamento Central	Es aquel que se encarga de ubicar al profesor en una facultad y asignarle la asignatura que debe impartir que deberá ser aceptada luego por el Jefe de Departamento de la Facultad.
Jefe de Departamento de la Facultad	Es aquel que acepta la propuesta de la asignatura a impartir por el profesor y el plan de trabajo del mismo así como los grupos que debe atender. También debe mediante el plan de trabajo confeccionado dar a conocer la planificación actual del profesor a cualquier interesado en la misma y llevar el control de la evaluación final del mismo indicando si cumplió con su plan de trabajo o no, además de encargarse de la gestión de la información de los profesores.

Jefe de Colectivo	Es aquel que se encarga de confeccionar el plan de trabajo del profesor y llevar el control del cumplimiento del mismo.
-------------------	---



### 2.3.3 Expansión de los casos de uso del negocio

Cada uno de los casos de uso del negocio se describe textualmente en formato expandido y se han utilizado además diagramas de actividad, artefactos que representan la secuencia de actividades que tienen lugar en cada uno de los diferentes casos de uso.

<b>Nombre del caso de uso:</b>	Ubicar Profesor
Actores del negocio:	Profesor (Inicia)
Propósito:	Ubicar al profesor en un Departamento Central y asignarle la asignatura a impartir así como los grupos que debe atender.
<p>Resumen: El caso de uso se inicia cuando el profesor se presenta en el centro con el propósito de ser aceptado como tal. Si el profesor cumple con todos los requisitos principales para poder entrar en el centro entonces es aceptado y se le ubica en un Departamento y se guardan todos los datos del mismo. Posteriormente se presenta en el Departamento Central para recibir su planificación. El caso de uso termina cuando se ha ubicado al profesor en una facultad y se le han asignado la asignatura que deberá impartir en la Facultad así como los grupos que debe atender.</p>	
Casos de uso asociados:	-
<b>Acción del actor</b>	<b>Respuesta del negocio</b>
1. El profesor solicita ser aceptado como profesor en el centro y se presenta en la entrevista.	2. El Director de Formación verifica que el profesor esté en condiciones para entrar en el centro y que cumpla con todos los requisitos.  a) Si no está en condiciones para ingresar recibe la información de que no ha sido aceptado y finaliza el proceso.

	b) Si está en condiciones se prosigue al paso 3.
	3. El Director de Formación archiva todos sus datos y ubica al profesor en un Departamento.
	4. El jefe de Departamento Central recibe los datos del profesor y lo ubica en una facultad.
	5. El jefe de Departamento Central propone la asignatura que debe impartir el profesor.
	6. El Jefe de departamento de la facultad recibe la propuesta : -Si este acepta la propuesta continuar el proceso de asignación del profesor. -Si no lo acepta regresar al paso 5.
	7. El Jefe de departamento de la facultad define los grupos que debe atender el profesor.
8. El profesor recibe su asignación.	
<b>Prioridad:</b>	
<b>Mejoras:</b>	La automatización de este proceso permite tener un adecuado control de la información de todos los profesores que ingresan en la Universidad.
<b>Cursos alternos: -</b>	



<b>Nombre del caso de uso:</b>	Planificar profesor.
<b>Actores del negocio:</b>	Profesor
<b>Propósito:</b>	Confeccionar el plan de trabajo del profesor para así llevar el control de las actividades que deberá realizar el mismo durante el curso.
<p><b>Resumen:</b> El caso de uso se inicia cuando el profesor se presenta en la facultad solicitando su plan de trabajo. El Jefe de Colectivo de la asignatura conjuntamente con el profesor realiza la planificación del profesor la cual debe ser aceptada después por el Jefe de Departamento Facultad.</p>	
<b>Casos de uso asociados:</b>	
<b>Flujo de trabajo</b>	
<b>Acción del actor</b>	<b>Respuesta del negocio</b>
1. El profesor se presenta en la facultad y hace la solicitud.	2. El Jefe de Departamento de la Facultad indica al Jefe de Colectivo confeccionarle el plan de trabajo al profesor.
	3. El Jefe de Colectivo realiza el plan de trabajo del profesor.
	4. El Jefe de Departamento de la Facultad recibe el plan de trabajo del profesor: -Acepta la propuesta. -No acepta la propuesta regresar al paso 3.

5. El profesor recibe su plan de trabajo.	
<b>Prioridad:</b>	Este paso se hace necesario para tener un control de todas las actividades que debe cumplir el profesor durante el curso.
<b>Mejoras:</b>	El plan de trabajo del profesor mejorará el control que se tiene de las diferentes actividades que debe cumplir el profesor durante el curso. La automatización de este proceso permitirá al Jefe de Departamento de la Facultad mejorar el proceso de gestión de los profesores.
<b>Cursos alternos:</b>	

<b>Nombre del caso de uso:</b>	Obtener evaluación del profesor
<b>Actores del negocio:</b>	Interesado
<b>Propósito:</b>	Mantener un control de todas las actividades que debe realizar el profesor para cumplir con su plan de trabajo.
<b>Resumen:</b> El caso de uso se inicia cuando un interesado, puede ser algún Directivo o el propio profesor, solicita en la facultad la evaluación final de un profesor en el curso. El caso de uso termina cuando se le haya entregado la evaluación final al interesado y se hayan guardado los datos.	
<b>Casos de uso asociados:</b>	
<b>Flujo de trabajo</b>	

<b>Acción del actor</b>	<b>Respuesta del negocio</b>
1. El interesado solicita su evaluación en la facultad.	2. El Jefe de Departamento de la Facultad recibe la solicitud y busca los resultados:  -Si no tiene la evaluación ir al paso 3.  -Si tiene la evaluación ir al paso 6.
	3. El Jefe de Departamento de la Facultad solicita al Jefe de Colectivo de la asignatura los resultados del plan de trabajo del profesor en el curso.
	4. El Jefe de Colectivo recibe la solicitud y busca y entrega los resultados correspondientes.
	5. El Jefe de Departamento de la facultad recibe los datos y registra los resultados.
	6. El Jefe de Departamento de la facultad entrega los datos solicitados al interesado.
7. El interesado recibe la evaluación final del profesor después de transcurrido el curso.	
<b>Prioridad:</b>	
<b>Mejoras:</b>	La automatización de este proceso

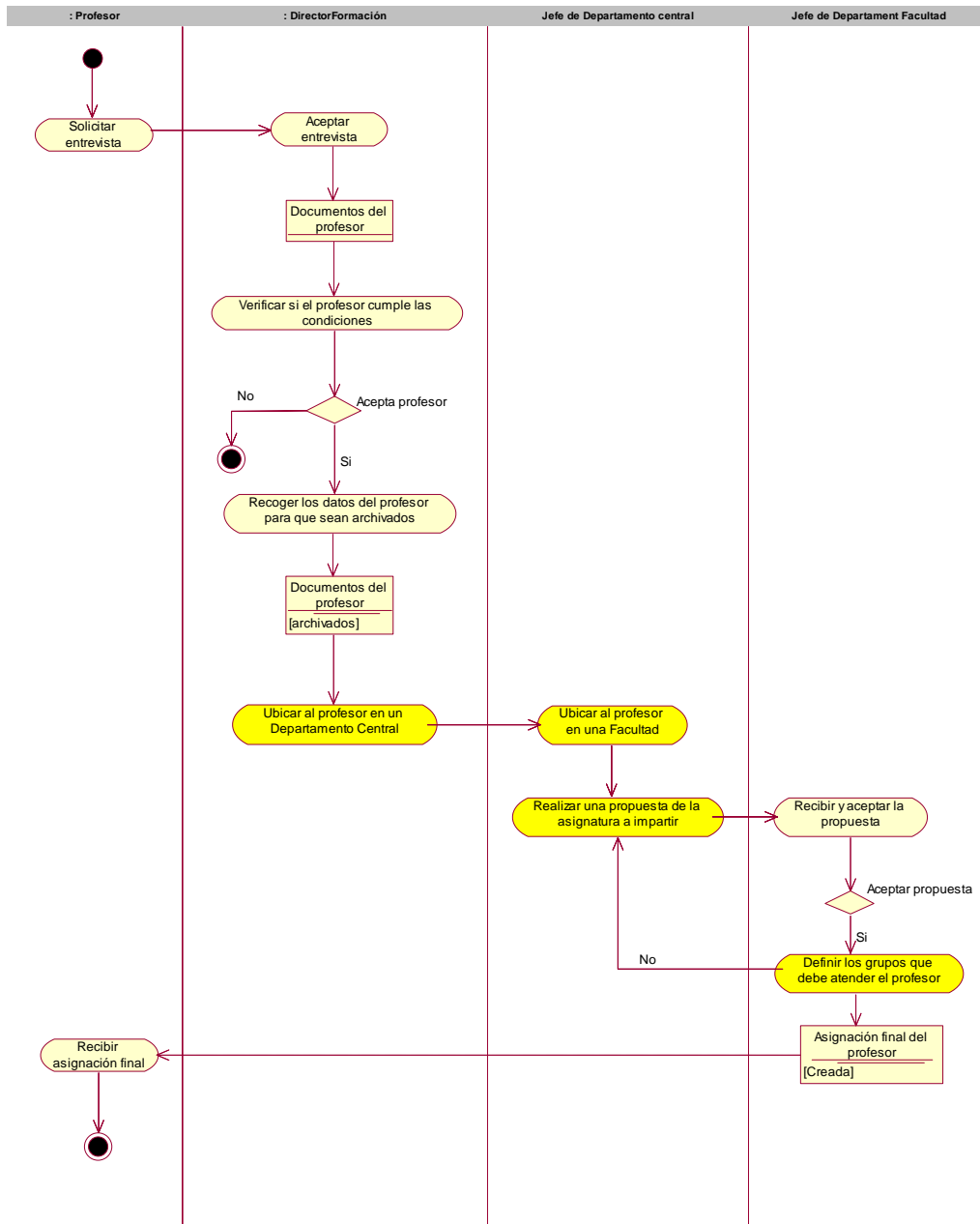
	<p>permitirá mantener un control eficiente del cumplimiento de todas las actividades abordadas en el plan de trabajo de los profesores.</p>
<p><b>Cursos alternos:</b></p>	

### 2.3.4 Diagramas de actividades

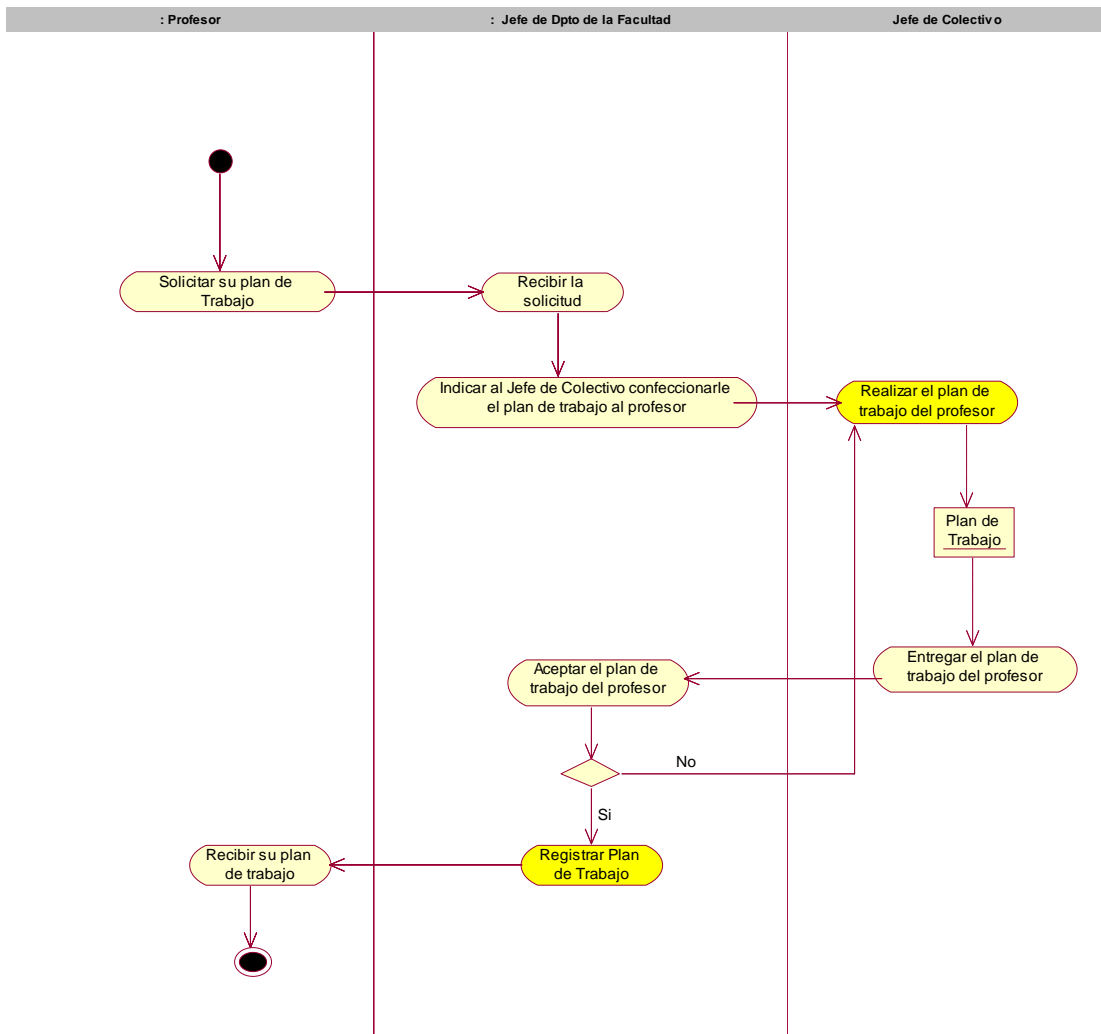
Los diagramas de actividades describen el flujo de trabajo asociado a los procesos de negocio. Muestran a través de las calles las responsabilidades de los trabajadores del negocio y a través del flujo de objetos cómo se utilizan las entidades del negocio.

Las entidades del negocio representan los objetos que los trabajadores del negocio utilizan o generan durante la realización de los procesos de negocio. Las actividades que aparecen sombreados en amarillo con una tonalidad más intensa son las actividades a automatizar en el sistema.

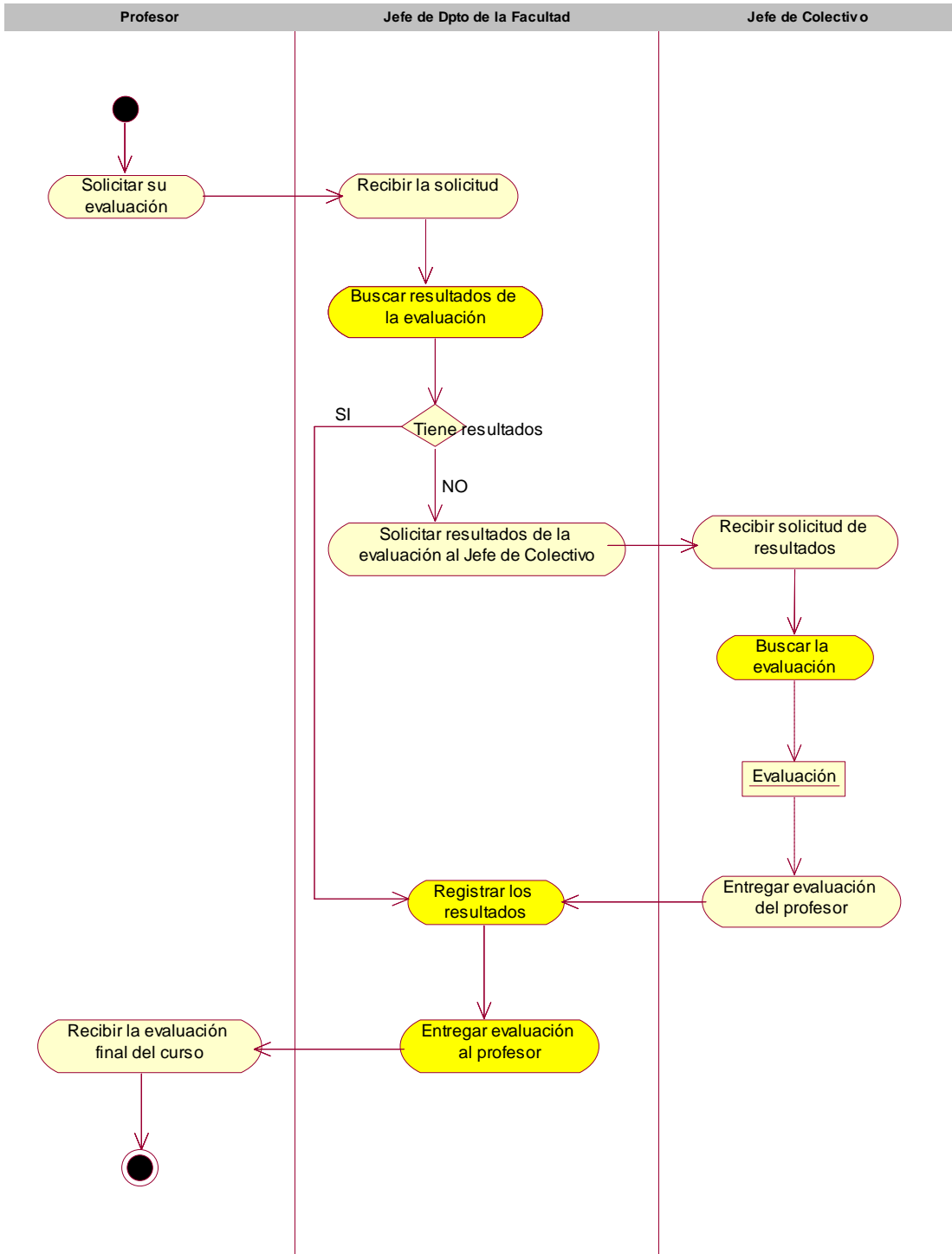
#### Caso de Uso Ubicar Profesor



### Caso de Uso Planificar Profesor



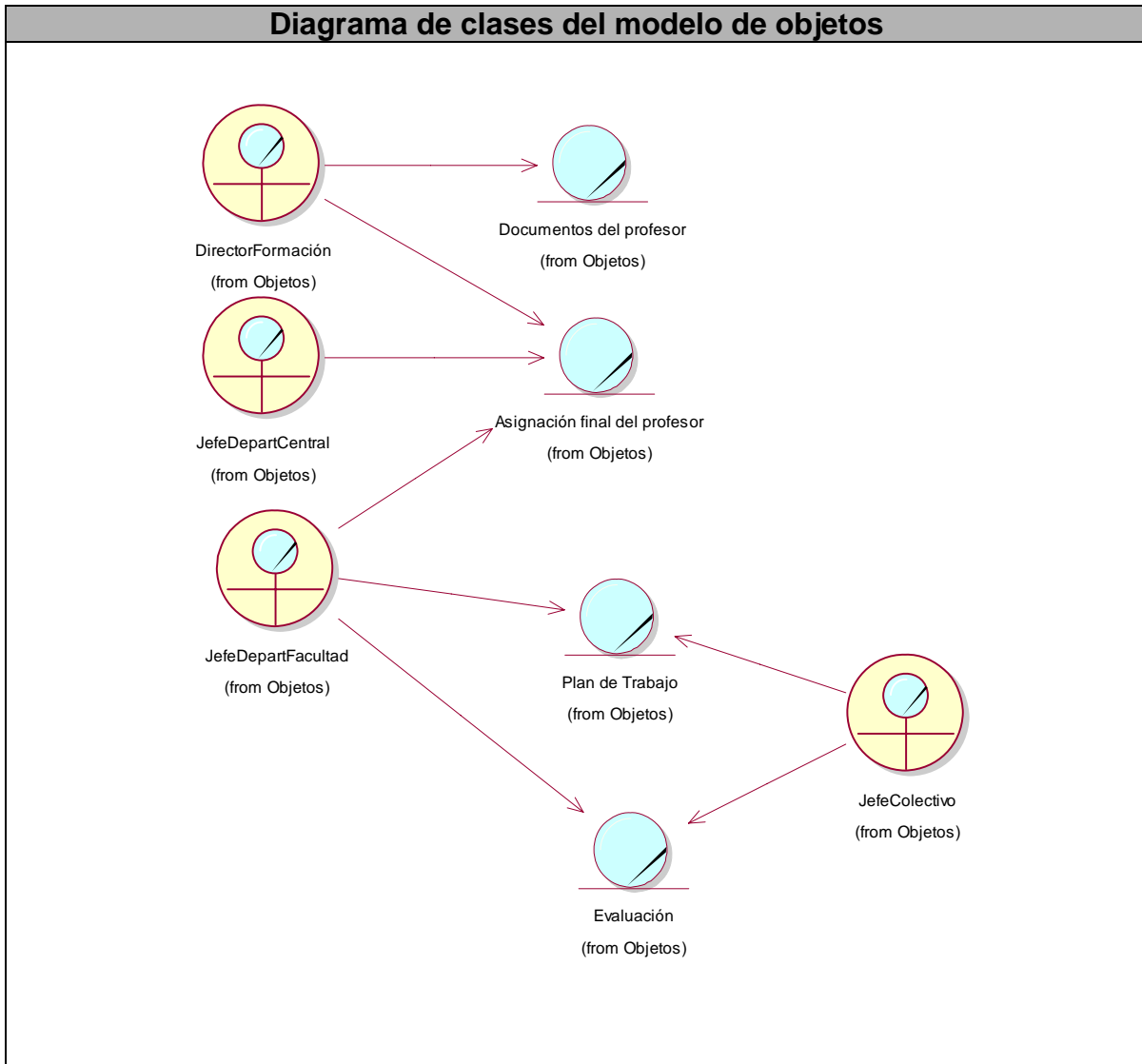
**Caso de Uso Obtener evaluación del profesor**



### **2.3.5 Diagrama de clases del modelo de objetos**

Los diagramas de clases del modelo de objetos describen el modelo de objetos del negocio, muestran la participación de los trabajadores y entidades del negocio y la relación que se establece entre ellos. Describe cómo cada caso de uso del negocio es llevado a cabo por un conjunto de trabajadores que utilizan un conjunto de entidades y unidades de trabajo del negocio.





## 2.4 Solución propuesta

La solución que se propone es la realización de un Módulo de Gestión de Profesores que corresponda al Sistema de Gestión Académica (AKADEMOS) y se relacione con los otros módulos que forman parte de este.

Dentro de las funcionalidades con que contará este módulo está permitir a los profesores interactuar directamente con el mismo de forma tal que puedan consultar

información referente a su planificación, así como los grupos a los que debe atender y las asignaturas a impartir durante el curso, además contará con un expediente en el cual estará registrada parte de la información del profesor, estarán incluidos otros documentos que pertenecen al profesor como su evaluación final después de haber culminado el curso.

Las secretarías docentes de la facultad también podrán consultar y registrar profesores pertenecientes a su facultad.

El sistema será capaz de realizar el proceso de asignación del profesor que se realiza con el ingreso de un nuevo profesor en el centro. Para llevar a cabo este proceso la secretaria docente de cada facultad deberá inicialmente indicar en el sistema la facultad a la cual va pertenecer el profesor, el sistema permite realizar una búsqueda del mismo y mostrar sus datos, se dará la opción a la secretaria de poder registrar al profesor en esa facultad, además podrá asignarle un grupo académico y en dependencia de este la asignatura a impartir, también eliminar un profesor de su facultad. En AKADEMOS se define la relación estudiante-profesor-asignatura que representa los estudiantes que están matriculados en una asignatura dada y la imparten en un grupo administrativo específico y con determinado profesor. Ver Figura 2.1.

El sistema permite realizar el plan de trabajo del profesor donde quedarán registradas todas las actividades con las que debe cumplir un profesor durante el curso, a este puede tener acceso el propio profesor así como otros Directivos de la Universidad, además de las secretarías docentes de cada facultad. El sistema permite dar a conocer la planificación actual de un profesor, pero además conocer también la de años anteriores a través del plan de trabajo del profesor.

Este módulo da la posibilidad de registrar un histórico de profesores donde quedará almacenada parte de la información de la planificación de los profesores.

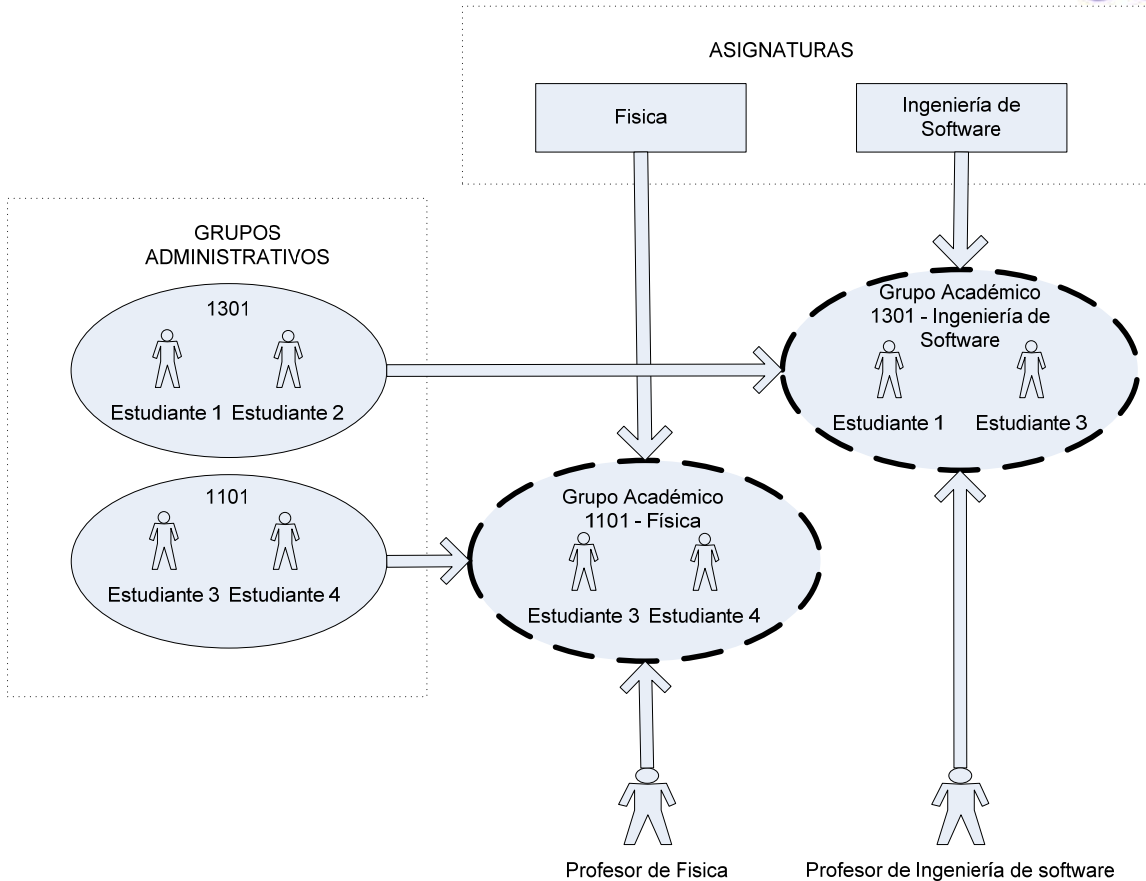


Figura 2.1. Grupo académico profesor.

## 2.5 Requerimientos funcionales

Los requisitos o requerimientos se pueden clasificar en: funcionales y no funcionales. Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, en cambio, los no funcionales se refiere a cualidades del sistema.

### 1. Distribuir profesores

#### 1.1. Ubicar al profesor:

1.1.1. El sistema debe permitir ubicar al profesor en un Departamento Central.

1.1.2. El sistema debe permitir ubicar al profesor en una Facultad.

#### 1.2. Realizar asignación al profesor.

1.2.1. Asignar asignatura que debe impartir el profesor.

- 1.2.2. Asignar los grupos que debe atender el profesor.
    - 1.2.3. Asignar cargos al profesor.
  - 1.3. Gestionar datos del profesor:
  - 1.4. Mostrar Histórico:
    - 1.4.1. Se deben mostrar un reporte de todas las asignaturas que ha impartido el profesor en el centro con el objetivo de facilitar el proceso de asignación de la misma.
2. Elaborar el plan de trabajo del profesor:
  - 2.1. El sistema debe permitir elaborar el plan de trabajo del profesor:
    - 2.1.1. Insertar el plan de trabajo de un profesor.
    - 2.1.2. Actualizar el plan de trabajo de un profesor.
3. Gestionar expediente del profesor:
  - 3.1. El sistema debe permitir poder realizar un expediente de profesor donde se tratan los datos más importantes referentes al profesor.
  - 3.2. Mostrar expediente del profesor:
  - 3.3. Mostrar resultados de la evaluación final del profesor en el curso.
4. Mostrar el plan de trabajo del profesor:
  - 4.1. El sistema deberá mostrar el plan de trabajo actual del profesor a cualquier interesado ya sea el propio profesor u otros Directivos del centro así como su planificación en cursos anteriores.
5. Registrar histórico de profesores:
  - 5.1. El sistema debe permitir almacenar algunos datos del profesor en un histórico al cambiar de momento para conocer así las asignaturas impartidas por el profesor en el centro.
6. Obtener evaluación del profesor.

6.1. El sistema debe permitir recoger los resultados de la evaluación final del profesor después de haber concluido el curso.

7. Mostrar la planificación del profesor:

7.1. El sistema deberá mostrar la planificación actual de un profesor a cualquier interesado ya sea el propio profesor u otros Directivos del centro.

## **2.6 Requerimientos no funcionales**

### **Apariencia o interfaz externa**

- Debe tener un modelo claro y simple de utilizar para los usuarios, para que se sientan confiados con él.
- De no poderse ejecutar una acción por un usuario, visualizar un mensaje de error que especifique porque no se pudo ejecutar.
- Debe ser sencilla y mostrar claramente todas las opciones con el objetivo de agilizar todo el proceso.

### **Usabilidad**

- No todas las personas usarán el sistema de la misma forma y con el mismo objetivo, ya que existirán algunas, que sólo podrán consultar la información de los profesores si cuentan con ese permiso.

### **Rendimiento**

- Se debe garantizar que la respuesta a solicitudes de los usuarios del sistema sea en un tiempo breve para evitar la acumulación de trabajo por parte de los implicados.

### **Seguridad**

- Utilizar la seguridad de integrada de Window.
- Mantener actualizada la información de los profesores.

### **Implementación**

- Será implementada haciendo uso de la tecnología :Net y Sql Server como Sistema de Gestión de Base de Datos.

### **Software**

Cliente:

- Sistema operativo con interfaz grafica y soporte para red.
- Navegador Web.

Servidor:

- Windows 2000 Professional Service Pack 2.0 o superior.
- .NET Framework. 1.1.
- SQL Server 2000.

### **Hardware**

Clientes:

- Pentium de 133 MHz o superior.
- 128 MB de memoria RAM mínima, 256 MB de memoria RAM recomendada.

Servidor Web y de Correo:

- Dual PIII de 933MHz.
- 512 MB de memoria RAM.
- 60 GB de disco duro.

Servidor de Base de Datos:

- Dual PIII de 933MHz.
- 512 MB de memoria RAM.
- 100 GB de disco duro.

## **2.7 Beneficios Tangibles**

El beneficio fundamental del sistema es contar con una aplicación web flexible y dinámica que permita realizar la gestión de la planificación de los profesores así como mantener actualizada la información de los mismos y dar a conocer de la una forma más precisa otros datos de interés de los profesores desde que inician su vida en el centro. Para la gestión académica existen varios aspectos a tener en cuenta relacionadas con la información que debe manipularse. Los profesores u otros Directivos del centro pueden acceder directamente desde cualquier punto en la Universidad, esto incurre en un ahorro de materiales de oficina tales como papel, toner para la impresión, etc.

Además existen en el mundo otros sistemas que tienen objetivos equivalentes a los de Akademos, uno de ellos es el Ágora el cual brinda un conjunto de funcionalidades muy similares a las de Akademos, cuenta con un precio de 3160€. Con la implementación de este sistema se incurre en un ahorro por la compra de un sistema igual en el extranjero.

## **2.8 Beneficios Intangibles**

Con la realización de este sistema para la Universidad de las Ciencias Informáticas (UCI) se facilitará parte del proceso de gestión académica de la universidad, específicamente encargarse de la gestión de la información relacionada con los profesores del centro esto permite a los profesores acceder directamente a la aplicación desde cualquier computadora de la universidad que se halle conectada a la red. Por otra parte aliviará el trabajo manual que realizan los implicados en esta labor, ya que con este sistema automatizado se dará solución a una de las tareas más prioritarias de la universidad.

La información que manipula el sistema, se encuentra actualizada, permitirá mostrar todo lo relacionado con la planificación de los profesores y así llevar un control de toda la gestión de los profesores, aspecto importante para la toma de decisiones en aras de lograr la calidad de los mismos.

De esta forma se simplificará el trabajo de los implicados en este proceso y se creará una nueva y rápida vía de gestión y actualización de información.

## **2.9 Análisis costo – beneficios**

Debido al gran papel importante que juega el profesor dentro de un centro de estudios como la UCI, el uso de una herramienta que automatice el proceso de gestión y planificación de profesores es imprescindible. Sin embargo para la puesta en marcha del proyecto es necesario tomar en cuenta los beneficios que aporte y los costos asociados a al mismo.

Analizando los beneficios tangibles e intangibles adquiridos por el sistema que se propone y teniendo en cuenta que la UCI cuente con las condiciones necesarias para la elaboración y explotación del mismo se considera factible el desarrollo del sistema.

## **2.10 Modelo del sistema**

### **2.10.1 Modelo de casos de uso del sistema**

El modelo de los casos de uso del sistema representa un esquema donde se recogen las funcionalidades del negocio que se automatiza y determina cómo será utilizado desde la perspectiva del usuario (Actor) pues se construye sobre la base de sus necesidades.

Los actores representan los usuarios del sistema. Una vez identificados se tiene delimitado el entorno externo del sistema. Los actores para esta aplicación son los siguientes:

#### **Definición de actores del sistema**

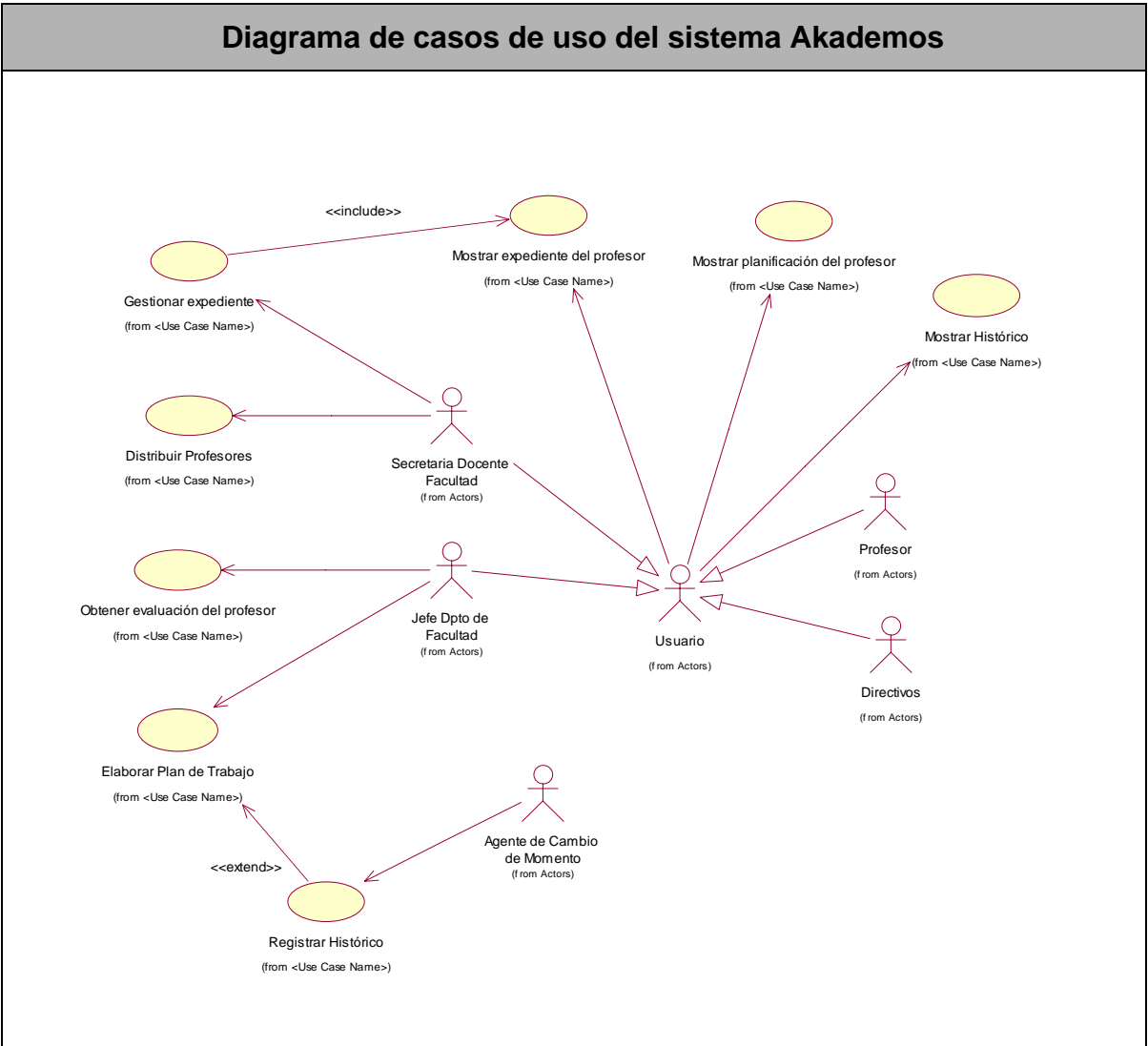
<b>Actores</b>	<b>Justificación</b>
Secretarias docentes de Facultad	Son las encargadas de realizar todo el proceso de asignación del profesor, ubicar al profesor en una Facultad, asignarle la asignatura a impartir y los grupos que debe atender. Además podrá tener acceso a toda



	la información del profesor así como su historial, planificación actual y plan de trabajo. Se encarga de la gestión de los profesores y del expediente del mismo.
Jefe de Departamento de la Facultad	Es el encargado de confeccionar el plan de trabajo del profesor y llevar el control del cumplimiento del mismo durante el curso. Además podrá tener acceso a toda la información referente al profesor.
Agente de cambio de Momento	Este actor no es humano, es un sistema que se encarga de enviar un aviso indicando que se ha cambiado de momento y por tanto se debe actualizar el histórico.
Profesor	Puede tener acceso a toda la información del profesor así como su historial, planificación actual y plan de trabajo.
Directivos	Puede tener acceso a toda la información del profesor así como su historial, planificación actual y plan de trabajo.
Usuario	Es cualquier persona que puede tener acceso a toda la información del profesor así como su historial, planificación actual y plan de trabajo. Puede ser el propio profesor, secretarías docentes de Facultad u otros Directivos del centro.

### 2.10.2 Diagrama de casos de usos del sistema

El diagrama de casos de usos del sistema: Artefacto narrativo que describe, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario [14].



### 2.10.3 Casos de uso del sistema

Caso de Uso	Distribuir Profesores
Actores	Secretarias docentes de facultad
Descripción	Con este caso de uso las secretarias docentes de cada facultad podrán registrar al profesor actualizando así el listado de profesores a los que tenga acceso de acuerdo a la facultad a la que pertenezca. Podrán adicionar los grupos académicos al profesor correspondiente y en dependencia de este, la asignatura a impartir por el mismo, eliminar un profesor y conocer de cada uno las asignaturas y grupos que atiende y si presta servicios como profesor en otra facultad, además de tener acceso al histórico de profesores.
Referencias	RF 1

Caso de Uso	Gestionar Expediente
Actores	Secretarias docentes de facultad
Descripción	Con este caso de uso las secretarias docentes de cada facultad podrán registrar el expediente de un profesor, a los que tenga acceso de acuerdo a la facultad a la que pertenezca. Podrán crear el expediente al profesor seleccionado y permite atachar documentos al mismo como la evaluación final del profesor. Permite crear un nuevo documento del profesor y gestionar expedientes.
Referencias	RF 3,Caso de Uso Mostrar Expediente (include)

Caso de Uso	Elaborar el Plan de Trabajo
Actores	Jefe de Departamento de la Facultad
Descripción	Con este caso de uso el Jefe de Departamento de la Facultad podrán crear el plan de trabajo del profesor accediendo directamente al listado de profesores de su Facultad, donde quedarán registradas todas las actividades que debe cumplir el profesor durante el curso, manteniendo así un control del cumplimiento de las mismas. Podrá modificar el plan de trabajo del profesor.
Referencias	RF 2,Caso de Uso Registrar Histórico (extend)

Caso de Uso	Registrar Histórico
Actores	Agente de Cambio de Momento
Descripción	Permite actualizar el registro histórico de profesores, este caso de uso se inicia al ocurrir un cambio de momento y debe pasarse todos los registros corrientes de las planificaciones a los registros históricos del profesor.
Referencias	RF 5

Caso de Uso	Mostrar planificación del profesor
Actores	Usuario
Descripción	Podrá mostrar la planificación del profesor en un curso mostrando todas las actividades a cumplir por el mismo, también permite

	mostrar su planificación en cursos anteriores.
Referencias	RF 4

Caso de Uso	Mostrar expediente del profesor
Actores	Usuario
Descripción	Este caso de uso se inicia cuando el usuario solicita acceder al expediente de un profesor y en dependencia de quien se haya logueado en el sistema mostrar el expediente de un profesor seleccionado y de cada uno de la información que contiene incluido su evaluación durante el curso.
Referencias	RF 3

Caso de Uso	Obtener evaluación del profesor
Actores	Usuario
Descripción	Permite obtener la evaluación final del profesor al culminar el curso es decir al ocurrir un cambio de nivel de acuerdo al cumplimiento de las tareas tratadas en el plan de trabajo del mismo.
Referencias	RF 6

Caso de Uso	Mostrar Histórico
Actores	Usuario

Descripción	Este caso de uso se inicia cuando el usuario solicita acceder al histórico de profesores y en dependencia de quien se haya logueado en el sistema mostrar el histórico de profesores y de cada uno de la información que contiene incluida su planificación durante el curso y en otros anteriores.
Referencias	RF 5

### Descripción de los Casos de uso del Sistema en formato expandido

Caso de uso:	Registrar Histórico	
Actores:	Agente de cambio de nivel	
Propósito:	Pasar parte de la información corriente del registro de planificación del profesor a los registros históricos de profesores.	
Resumen:	El caso de uso se inicia cuando ocurre un cambio de momento y se debe pasar parte de la información relacionada con la planificación del profesor al registro histórico o cuando se le modifica el plan de trabajo del profesor porque ocurre un cambio en su planificación. El caso de uso termina cuando se haya actualizado el histórico de profesores.	
Referencias:	<b>R5</b>	
Precondición:	Debe haber ocurrido un cambio de momento u ocurrido un cambio en la planificación del profesor.	
Flujo de sucesos:		
Acción del Actor	Respuesta del Sistema	
1. Se avisa que ocurrió un cambio de momento o cuando haya ocurrido algún cambio en la planificación del profesor.	2. El sistema selecciona la última edición de momento para ese plan de estudio.	

	3. El sistema selecciona todas las planificaciones de los profesores para esa edición de momento de ese plan de estudio.
	4. El sistema busca la evaluación final en el plan de trabajo del profesor por cada una de las planificaciones.
	5. Inserta en el registro histórico estas planificaciones y la evaluación final por cada una.
	6. Elimina las planificaciones corrientes seleccionadas de todos los profesores para esa edición de momento.
Cursos alternos	
Poscondición	Han pasado al histórico de profesores todos los registros de planificación de los profesores.

Caso de uso:	Distribuir profesores
Actores:	Secretarias docentes de Facultad
Propósito:	Realizar el proceso de asignación del profesor, es decir permitir ubicar al profesor en una Facultad, asignarle la asignatura a impartir y los grupos a atender. Además se encarga de la gestión



	de la información de los profesores.	
Resumen:	El caso de uso se inicia cuando la secretaria docente de la facultad solicita registrar un profesor, para lo cual debe seleccionar inicialmente la facultad donde se muestra un listado con todos los profesores que pertenecen a la facultad seleccionada, luego se realiza una búsqueda del profesor a registrar donde se muestran los datos del mismo y se permite registrarlo en la facultad seleccionada, permite además adicionar un grupo académico y la asignatura a impartir así como eliminar un profesor.	
Referencias:	<b>R1</b>	
Precondición:	El usuario debe haberse logueado en el sistema.	
Flujo de sucesos:		
Acción del Actor	Respuesta del Sistema	
1. La secretaria docente de la facultad selecciona el vínculo al módulo de Profesores en la página principal.	2. El sistema muestra la página “Registrar Profesor” con todos los planes de estudio y la estructura administrativa raíz.	
3. La secretaria docente selecciona la estructura padre para poder llegar a la facultad a la que pertenece.	4. El sistema muestra todos los profesores que corresponden a la facultad seleccionada y los que prestan servicios a la misma.	
5. La secretaria docente solicita la opción “Registrar profesor”	6. El sistema muestra la página “Buscar profesor” y le indica que debe introducir el nombre del profesor a registrar.	

<p>7. La secretaria introduce el nombre del profesor a registrar en la facultad.</p>	<p>8. El sistema muestra un listado de todos los trabajadores del centro con ese nombre y muestra la opción de adicionar profesor.</p>
<p>9. La secretaria selecciona el profesor a registrar y da clic en esta.</p>	<p>10.El sistema muestra el listado actualizado con los profesores correspondientes a la facultad seleccionada y los que prestan servicios.</p>
<p>11.La secretaria docente de la facultad selecciona al profesor registrado y da clic en la foto correspondiente para inicial así el proceso de asignación.</p>	<p>12.El sistema muestra la página del profesor y las diferentes tareas a seleccionar.</p>
<p>13.La secretaria elige una opción en específico.</p>	<p>a) Si elige Adicionar Grupo Académico ver sección “Adicionar grupo académico”.</p> <p>b) Si elige Eliminar profesor ver sección “Eliminar Profesor”</p>
<p>Sección: “Adicionar Grupo Académico”</p>	
<p>1.La secretaria docente elige la opción “Adicionar grupo Académico”.</p>	<p>2. El sistema muestra la página “Adicionar grupo académico” e indica introducir todos los datos necesarios: nivel, momentos, grupos y asignaturas.</p>
<p>3.La secretaria docente de la facultad introduce todos los datos necesarios.</p>	<p>4. El sistema verifica que el grupo académico seleccionado exista y que el profesor seleccionado no se encuentre ya</p>

	asignado a ese grupo con dicha asignatura.
	5. El sistema registra toda la información especificada anteriormente.
<b>Sección: "Eliminar profesor".</b>	
1. La secretaria docente elige la opción Eliminar Profesor.	2. El sistema verifica y muestra los datos del profesor:  a) Si no tiene grupos académicos asignados se elimina el profesor.  b) Si tiene grupos académicos asignados ir al paso 3.
3. La secretaria selecciona los grupos a eliminar y da clic en los mismos.	4. El sistema elimina los grupos académicos seleccionados conjuntamente con el profesor a eliminar.
<b>Caminos alternativos</b>	
<p>En el paso 3 si elige la opción Registrar Profesor sin seleccionar la facultad a la que pertenece se le informa que debe seleccionar una facultad para poder continuar.</p> <p>En el paso 4 de la sección "Adicionar grupo Académico" el grupo académico asignado no existe entonces se le informa que el grupo especificado no existe.</p> <p>En el paso 4 de la sección "Adicionar grupo Académico" ya el profesor seleccionado se encuentra asignado a ese grupo académico con dicha asignatura.</p>	
<b>Poscondición</b>	Ha sido registrado el profesor en una facultad.

Prototipo de Interfaz	Ver anexo 1
-----------------------	-------------

Caso de uso:	Mostrar Histórico	
Actores:	Usuario	
Propósito:	Mostrar reporte de todas las asignaturas impartidas por cada uno de los profesores del centro.	
Resumen:	El caso de uso comienza cuando un usuario del sistema entra al módulo del profesor. Selecciona el plan de estudio, la facultad y el profesor para el cual desea el historial.	
Referencias:	<b>R5</b>	
Precondición:	El usuario debe haberse logueado en el sistema.	
Flujo de sucesos:		
Acción del Actor	Respuesta del Sistema	
1. El usuario selecciona el vínculo al módulo de Profesores en la página principal.	2. El sistema muestra la página “Histórico Profesor” con todos los planes de estudio y la estructura administrativa raíz.	
3. El usuario selecciona la estructura padre para poder llegar a la facultad a la que pertenece.	4. El sistema muestra todos los profesores que corresponden a la facultad seleccionada y los que prestan servicios a la misma.	

<p>5. El usuario selecciona la estructura padre para poder llegar a la facultad a la que pertenece.</p>	<p>6. El sistema muestra todos los profesores que corresponden a la facultad seleccionada y los que prestan servicios a la misma.</p>
<p>7. El usuario selecciona el profesor correspondiente.</p>	<p>8. El sistema muestra la opción "Consultar Histórico".</p>
	<p>9. El sistema muestra un listado de las asignaturas impartidas por un profesor en el centro.</p>
<p><b>Caminos alternativos</b></p>	
<p><b>Poscondición</b></p>	<p>Ha sido mostrado el histórico de profesores con todos los registros de planificación de los profesores.</p>
<p><b>Prototipo de Interfaz</b></p>	<p>Ver anexo 2</p>

<p><b>Caso de uso:</b></p>	<p>Mostrar Planificación del profesor</p>
<p><b>Actores:</b></p>	<p>Usuario</p>

Propósito:	Mostrar la planificación actual de un profesor en el centro.	
Resumen:	El caso de uso comienza cuando un usuario del sistema entra al módulo del profesor. Selecciona el plan de estudio, la facultad y el profesor para el cual desea el historial.	
Referencias:	<b>R4</b>	
Precondición:	El usuario debe haberse logueado en el sistema.	
Flujo de sucesos:		
Acción del Actor	Respuesta del Sistema	
1. El usuario selecciona el vínculo al módulo de Profesores en la página principal.	2. El sistema muestra la página “Planificación Profesor” con todos los planes de estudio y la estructura administrativa raíz.	
3. El usuario selecciona la estructura padre para poder llegar a la facultad a la que pertenece.	4. El sistema muestra todos los profesores que corresponden a la facultad seleccionada y los que prestan servicios a la misma.	
5. El usuario selecciona la estructura padre para poder llegar a la facultad a la que pertenece.	6. El sistema muestra todos los profesores que corresponden a la facultad seleccionada y los que prestan servicios a la misma.	
7. El usuario selecciona el profesor correspondiente.	8. El sistema muestra la opción “Planificación Profesor”.	

	<p>9. El sistema muestra la planificación actual de un profesor en el centro, es decir grupos y asignaturas a impartir.</p>
<p>Caminos alternativos</p>	
<p>Poscondición</p>	<p>Ha sido mostrado el histórico de profesores con todos los registros de planificación de los profesores.</p>
<p>Prototipo de Interfaz</p>	<p>Ver anexo 3.</p>

### **Conclusiones**

En el presente capítulo, y sobre la propuesta de RUP para el desarrollo de software, se realizó una modelación del negocio donde se pudo comprender la estructura y dinámica del proceso de planificación de los profesores en la Universidad de las Ciencias Informáticas (UCI) y se determinaron las mejoras que puede aportar, al negocio, el sistema a desarrollar. Como resultado de esto se obtuvo un diagrama de casos de uso del negocio donde se definieron los actores y casos de uso del negocio

así como las relaciones entre los mismos. Cada caso de uso fue descrito en formato expandido y mediante diagramas de actividad donde fueron sombreadas las actividades a automatizar por el sistema. Además se trató la variante de automatización propuesta y se realizó la captura de los requerimientos funcionales del sistema. Se definió el diagrama de casos de uso del sistema con los actores y casos de uso del sistema y relaciones entre ellos, así como una descripción en formato expandido y los prototipos de interfaz de usuario para cada caso de uso. Además se obtuvo un listado de requisitos no funcionales a tener en cuenta para el desarrollo de la aplicación.



## Capítulo 3 Construcción de la Solución Propuesta

### ***Introducción***

El presente capítulo aborda los resultados de los flujos de trabajo de diseño e implementación, es decir, expone la construcción de la solución propuesta.

Muestra los diagramas de clases del diseño Web de Akademos y se modelan los nodos en los que se distribuye la aplicación mediante el Diagrama de Despliegue y se presenta el Diagrama de clases persistentes así como el modelo de datos obtenido a partir del anterior. También se modela en este capítulo el Diagrama de componentes de cada subsistema de implementación como artefacto de este flujo de trabajo, incluyendo la descripción de sus componentes. Aparecen además los estándares de programación y diseño de interfaz a seguir.

### ***3.1 Diagramas de clases del diseño Web***

La modelación de una aplicación Web facilita la manipulación y comprensión de las mismas. Incluye los componentes Web y los componentes tradicionales.

En el diagrama de clases del diseño se modelan las páginas, los enlaces entre estas, todo el código que irá creando las páginas, así como el contenido dinámico de estas una vez que estén en el navegador del cliente.

Diagrama de Clases Caso de Uso Distribuir Profesores. Ver anexo 4

Diagrama de Clases Caso de Uso Registrar Histórico. Ver anexo 5

Diagrama de Clases Caso de Uso Mostrar Histórico. Ver anexo 6

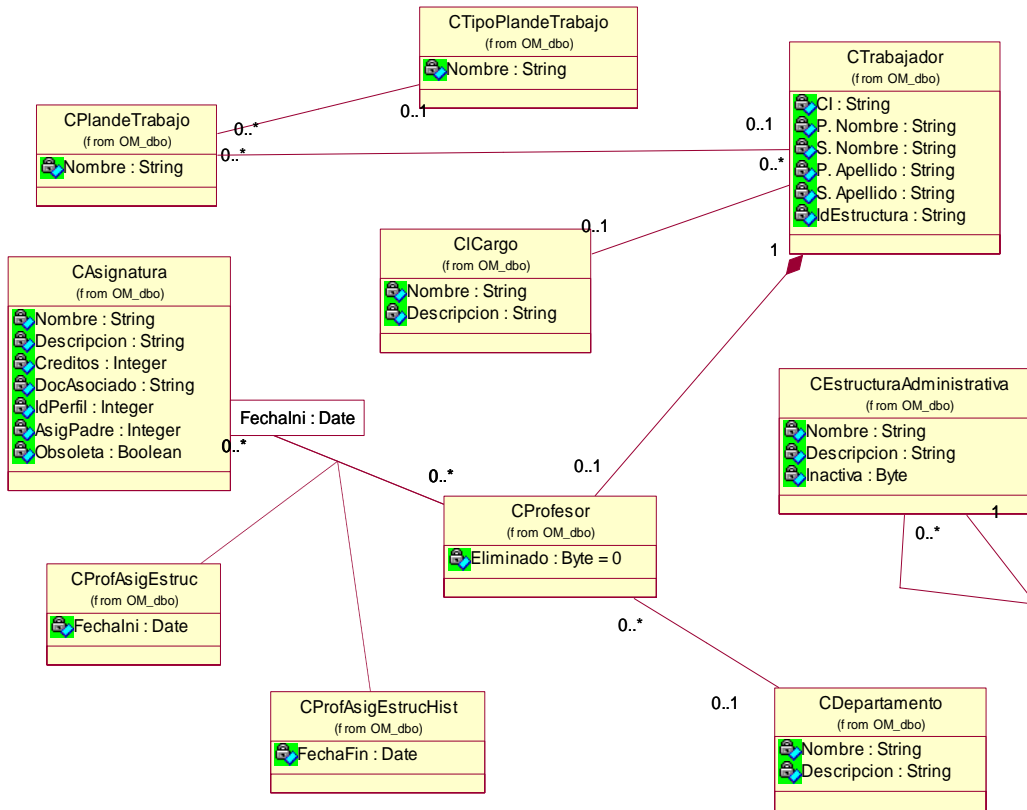
Diagrama de Clases Caso de Uso Mostrar planificación del profesor. Ver anexo 7

### ***3.2 Diseño de la base de datos***

#### ***3.2.1 Diagrama de clases persistentes***

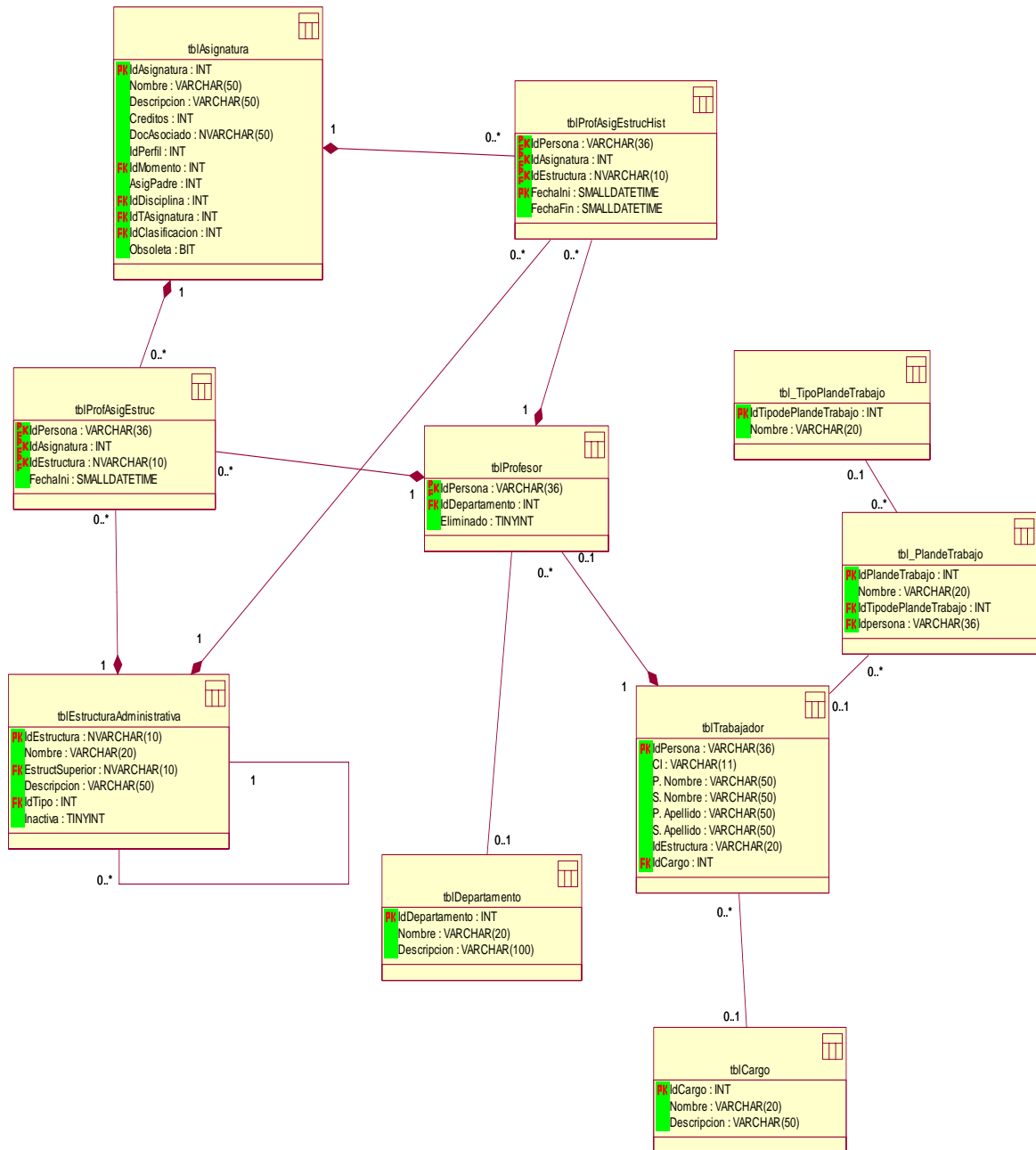
La persistencia es la propiedad que tiene los objetos de existir más allá del tiempo de vida de la aplicación. Por consiguiente las clases persistentes son las clases que

deben guardar su estado en un medio permanente. Estas clases en el modelo de aplicaciones de corte empresarial son comúnmente conocidas como clases entidad.



### 3.2.2 Modelo de datos

El modelo de los datos describe la representación lógica y física de datos persistentes en el sistema.



### **3.3 Principios de diseño**

#### **3.3.1 Estándares en la interfaz de la aplicación.**

La utilidad del diseño de la interfaz para una aplicación, es más que una finura filosófica, es esencial para el éxito de esta. El diseño de interfaz no solo está enfocado a apoyar y mantener el trabajo básico de los usuarios, ya que también puede estar orientado a extenderlo.

En el diseño de la interfaz del sistema primeramente se tuvieron en cuenta las operaciones necesarias que podían realizar los usuarios y se ubicaron como opciones en la parte izquierda de la pantalla, de forma tal que se garantizara la comodidad para su acceso. Además se tuvo en cuenta la organización de la información que se muestra, así como su distribución. Todas las pantallas tienen un diseño similar, con cambios solamente en las áreas que lo requieran, con el propósito de no desorientar al usuario. Por otra parte contienen únicamente la información necesaria evitando tener contenido excesivo en las páginas. Se ha usado para los textos un tamaño de letra estándar y un contraste entre las letras de color negro y el fondo verde claro para facilitar la lectura. De forma general se han tenido en cuenta el tamaño, espacio y colocación de los elementos para crear un ambiente visualmente consistente.

#### **3.3.2 Tratamiento de excepciones.**

Las excepciones permiten la notificación de errores ocurridos en un programa. El tratamiento de estas es imprescindible para la recuperación airosa de una aplicación en caso de condiciones que invaliden su funcionamiento permitiendo además notificar a los usuarios la ocurrencia de estas.

Para evitar errores, primeramente el sistema valida las entradas de los usuarios, indicando a nivel de interfaz los datos requeridos. Además en caso de error por valores incorrectos se señalan los datos a ser rectificadas.

Cualquier excepción que se eleve por condiciones inválidas para una operación, es capturada y mostrada como mensaje de error al usuario, incluso en ocasiones, con la posible solución para el problema.

### 3.3.3 Estándares de codificación.

Las características del proyecto requieren del trabajo en equipo. En la búsqueda de lograr una codificación coherente, correcta, incluso con un poco de estilo, y que sea entendible por todos los desarrolladores se especifican normas para escribir la aplicación. Los estándares o convenciones de código que se centran en la estructura física y la apariencia de este. En este caso se siguieron las convenciones para escribir código C#, y se especificaron normas para el grupo de desarrollo, tales como empezar con letra C las clases de acceso a datos, CN y WS para componentes empresariales y servicios Web respectivamente.

### 3.4 Diagrama de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. [14] El diagrama de despliegue para la aplicación que se desarrolla se muestra en la Figura 3.1.

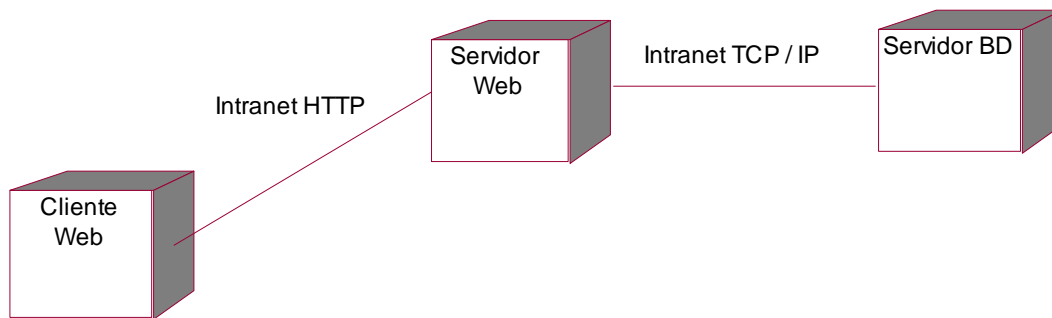


Figura 3.1. Diagrama de despliegue del sistema.

Nodos	Capacidad de memoria externa	de Memoria RAM	Potencia de Procesamiento
-------	------------------------------	----------------	---------------------------

<b>Ciente Web</b>	Puede ser la mínima posible para que tenga la instalación del Internet Explorer. (1 GB)	Puede ser la mínima posible para que tenga la instalación del Internet Explorer. (8 MB)	Puede ser la mínima posible para que tenga la instalación del Internet Explorer. (100 MHz)
<b>Servidor Web</b>	60 GB	512 MB	Dual PIII 933 MHz
<b>Servidor Base de Datos</b>	100 GB	512 MB	Dual PIII 933 MHz

### **3.5 Modelo de implementación**

El modelo de implementación describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros [14].

### **3.6 Diagramas de Componentes.**

El diagrama de componentes muestra el conjunto de componentes y sus relaciones. Un componente es una parte física y reemplazable del sistema que se conforma con un conjunto de interfaces y proporciona la realización de dicho conjunto. Se usan para modelar los elementos físicos que pueden hallarse en un nodo por lo que empaquetan elementos como clases, colaboraciones e interfaces. Ver anexo 8.

### **Conclusiones**

En el presente capítulo se han desarrollado los flujos de trabajo de diseño e implementación que propone RUP, modelándose un grupo de artefactos en cada uno de ellos. Como resultado de este capítulo se construyó el diagrama de clases

persistentes de la aplicación con el objetivo de obtener el modelo de datos y se definieron todos los elementos (componentes y nodos) que describen el modelo de implementación de la aplicación. Se modelaron los nodos en los que se distribuye la aplicación, especificando las conexiones de red y protocolos que los unen. Como parte de la implementación fue elaborado el diagrama de componentes en los que se estructura la aplicación.

Para mayor facilidad en cuanto a los componentes se muestra una tabla descriptiva de cada uno. En este capítulo, además, se abordó los estándares de programación y de diseño de interfaz que se siguen para el desarrollo del sistema.

## **Conclusiones.**

El desarrollo de este trabajo de tesis está orientado a la confección de una herramienta informática para la gestión de la planificación de los profesores en la UCI.

Con la implantación del mismo, se dará solución a uno de los problemas actuales existentes en la UCI, lo cual implica un mejoramiento en cuanto a la labor con los profesores, en las condiciones de trabajo de los involucrados en esel proceso así como en la reducción de errores y de información almacenada en papel.

Se espera que el documento haya logrado la comprensión teórica de la situación problemática existente y su solución, así como el desarrollo de las diferentes etapas de la aplicación usando la metodología RUP.

La utilización de este sistema no solo repercute en el mejoramiento del proceso de Gestión de Profesores de la UCI sino que le permite al propio profesor conocer sus resultados en las diferentes actividades abordadas en su plan de trabajo, además de otros datos de interés en su trayectoria por el centro.

El desarrollo del sistema se ha basado en el Proceso Unificado de Desarrollo de Software, se realizó una modelación del negocio con el propósito de entender el contexto del sistema y se presentó un grupo de artefactos para los flujos de trabajo de requisitos, diseño e implementación que propone dicho proceso.

Además con los beneficios tangibles e intangibles expuestos anteriormente se determinó que el desarrollo de la aplicación es realmente factible, siendo positiva la implantación de la misma en la Universidad de las Ciencias Informáticas de Cuba.



## Recomendaciones

- Se recomienda poner en práctica el sistema en las diferentes universidades del país con vista a facilitar el trabajo en las mismas.
- Lograr una nueva versión del software que perfeccione sus funcionalidades con nuevos elementos que surjan.
- Mantener el control de la actualización de la información del sistema Akademos.
- Analizar las sugerencias de los profesores en cuanto al sistema.
- Elaborar la ayuda para los usuarios del sistema.
- Se recomienda la migración a software libre de la aplicación.

## Referencias Bibliográficas

- [1] Valdés Lima Emil, Sistema Automatizado para la Gestión Académica, UCI 2005.
- [2] Tecnología Sonork. Aplicaciones Web.  
[http://www.sonork.com/esp/web\\_app.html](http://www.sonork.com/esp/web_app.html) (18/1/2005).
- [3] Make Multimedia, Servicios Web,  
<http://www.srt.com.ar/make2002-2/serviciosweb.asp> (8/2/2005).
- [4] Lizárraga Celaya Carlos, Universidad de Sonora, Servicios Web,  
<http://www.fisica.uson.mx/carlos/WebServices/WSRevolution.htm> (8/2/2005)
- [5] González Seco José Antonio, La Biblia de C# Totalmente en Español (.NET),  
<http://mentores.net/default.aspx?tabid=104&type=art&site=68&parentid=36>  
(18/1/2005).
- [6] Análisis Los Cimientos de las Aplicaciones Web.  
<http://www.cientec.com/analisis/loscimientos.asp> (18/1/2005).
- [7] Microsoft Corporation, Características de Visual Studio . Net.  
<http://www.microsoft.com/latam/vstudio/producto/caracteristicas.asp>(3/2/2005)
- [8] Microsoft Corporation. Tutoriales de Microsoft Asp .Net.  
<http://es.getdotnet.com/quickstart/aspplus/doc/whatisaspx.aspx> (4/2/2005)
- [9] Corporación Microsoft. “Arquitectura de aplicaciones de .NET: Diseño de aplicaciones y servicios”. Disponible en:  
<http://www.microsoft.com/spanish/msdn/arquitectura/default.asp>
- [10] Murillo Alfaro Felix, Instituto Nacional de Estadística e Informática, Perú, Tecnología Cliente Servidor.  
[http://www.pcm.gob.pe/portal\\_ongei/publicaciones/cultura/Lib5038/ven2.HTM](http://www.pcm.gob.pe/portal_ongei/publicaciones/cultura/Lib5038/ven2.HTM)  
(9/2/2005)
- [11] Norik, Aprenda Bases de Datos con MS SQL Server 2000.  
<http://usuarios.lycos.es/cursosgbd/UD2.htm> (21/2/2005)
- [12] Introducción a las Bases de Datos  
<http://elvex.ugr.es/decsai/windows/bd/1BD.pdf> (21/2/2005)
- [13] Bases de Datos SQL Server.  
<http://www.arsys.es/soporte/programacion/sql.htm> (21/2/2005).

- [14] JACOBSON, Ivar, BOOCH, Grady, RUMBAUGH, James. El proceso unificado de desarrollo de software, Pearson Educación S.A., 2000

## **Bibliografía**

JACOBSON, Ivar, BOOCH, Grady, RUMBAUGH, James. El proceso unificado de desarrollo de software, Pearson Educación S.A., 2000

Canaleta, Xavier y otros. Gestión académica y protección de datos. <http://web.salleurl.edu/~xavic/material/JENUI2003.pdf>, 2003.

Guerrero, Elpidio W. La importancia de la valoración de la gestión académica-administrativa y del estado de la infraestructura y el equipamiento. [www.congreso.unam.mx/ponsemloc/ponencias/1515.html](http://www.congreso.unam.mx/ponsemloc/ponencias/1515.html), 2003.

Miguel, Adoración de; Piatini, Mario G. “Fundamentos y modelos de bases de datos”. Madrid, 1997.

## **Glosario de Términos.**

Base de Datos (BD): Conjunto de datos interrelacionados, almacenados con carácter más o menos permanente en la computadora, puede ser considerada una colección de datos variables en el tiempo.

SQL (Structured Query Language): Conjunto estándar de comandos para gestionar bases de datos relacionales por sus mismas características relacionales.

SGBD: Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.

TCP/IP: Transmission Control Protocol/Internet Protocol "Es el conjunto de protocolos que definen a Internet. Originalmente diseñado para el sistema operativo UNIX, hoy en día existe software TCP/IP disponible para la mayoría de los sistemas operativos. Para poder utilizar la Internet, su computador debe tener software TCP/IP.

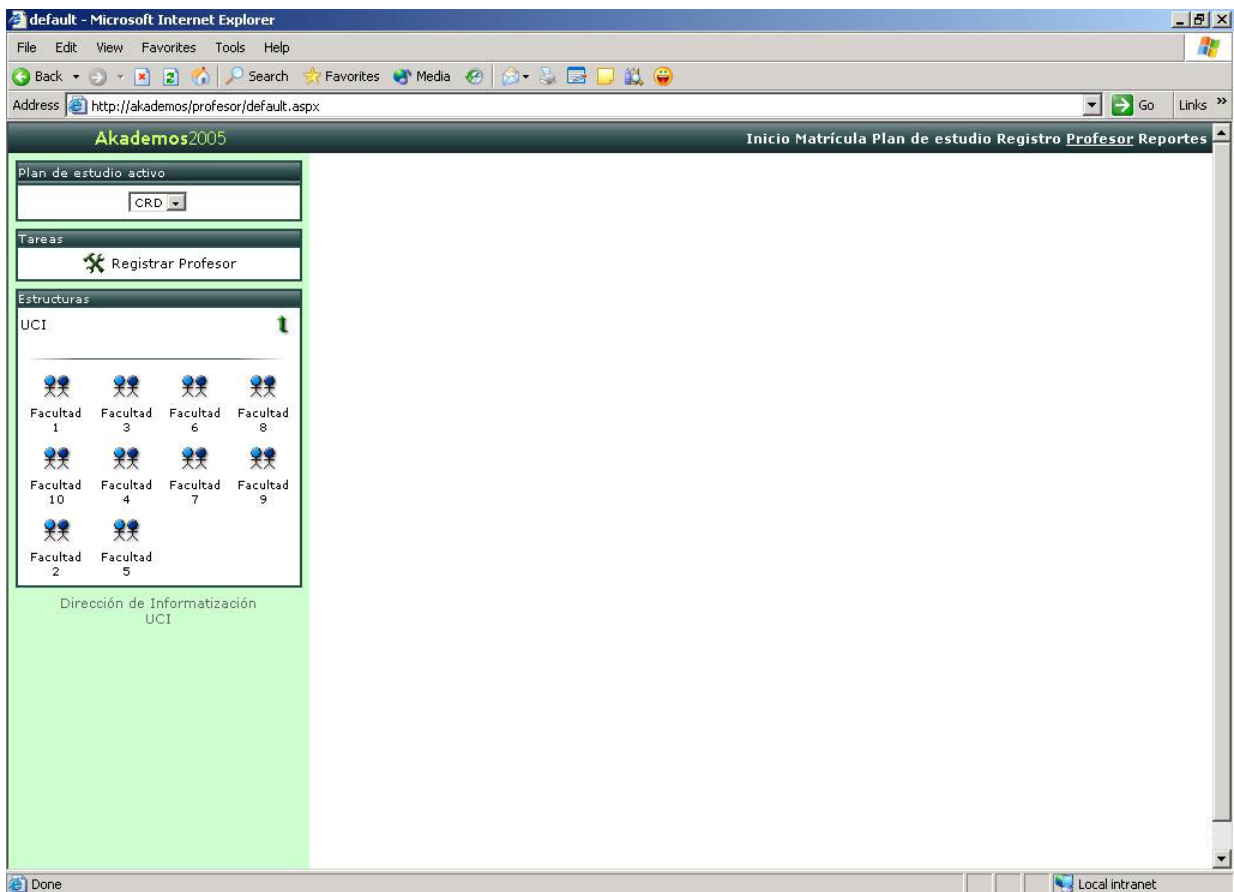
Servidor: Es un computador o un software que provee una clase especial de servicio a los software clientes que están corriendo en otros computadores y que lo accesan para realizar una función determinada. Un computador funcionando como servidor puede tener operando varios software servidores para prestar servicios, por ejemplo: servidor de WWW, servidor de FTP, de Mail, etc .

UML: "Unified Modeling Language" Lenguaje gráfico que brinda un vocabulario y reglas para especificar, construir, visualizar y documentar los artefactos de un sistema utilizando el enfoque orientado a objetos.

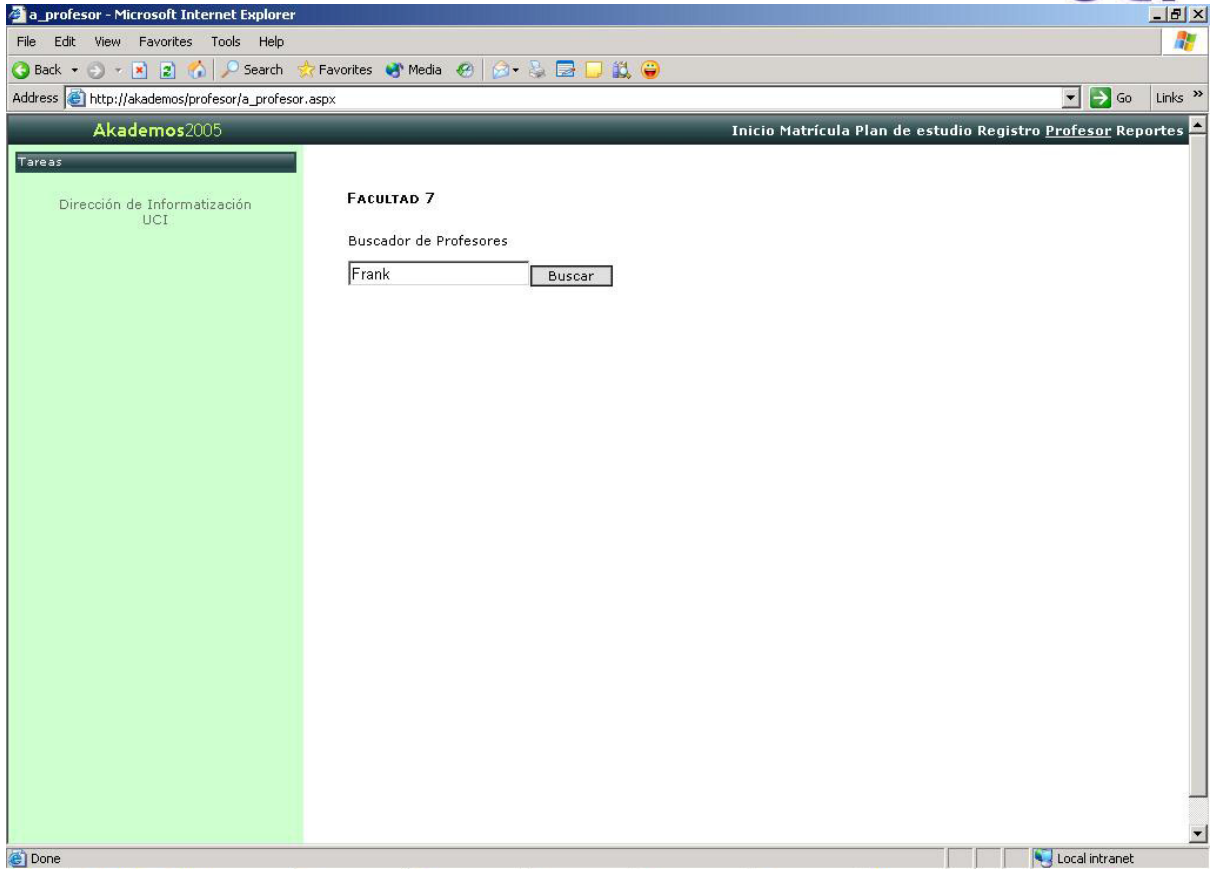
# Anexos

## Anexo 1

### Caso de Uso Distribuir profesores











**a\_profesor - Microsoft Internet Explorer**

File Edit View Favorites Tools Help

Address [http://akademos/profesor/a\\_profesor.aspx](http://akademos/profesor/a_profesor.aspx) Go Links

**Akademos2005** Inicio Matrícula Plan de estudio Registro Profesor Reportes

Tareas








Dirección de Informatización  
UCI

**FACULTAD 7**








Buscador de Profesores

Frank

7 Profesores

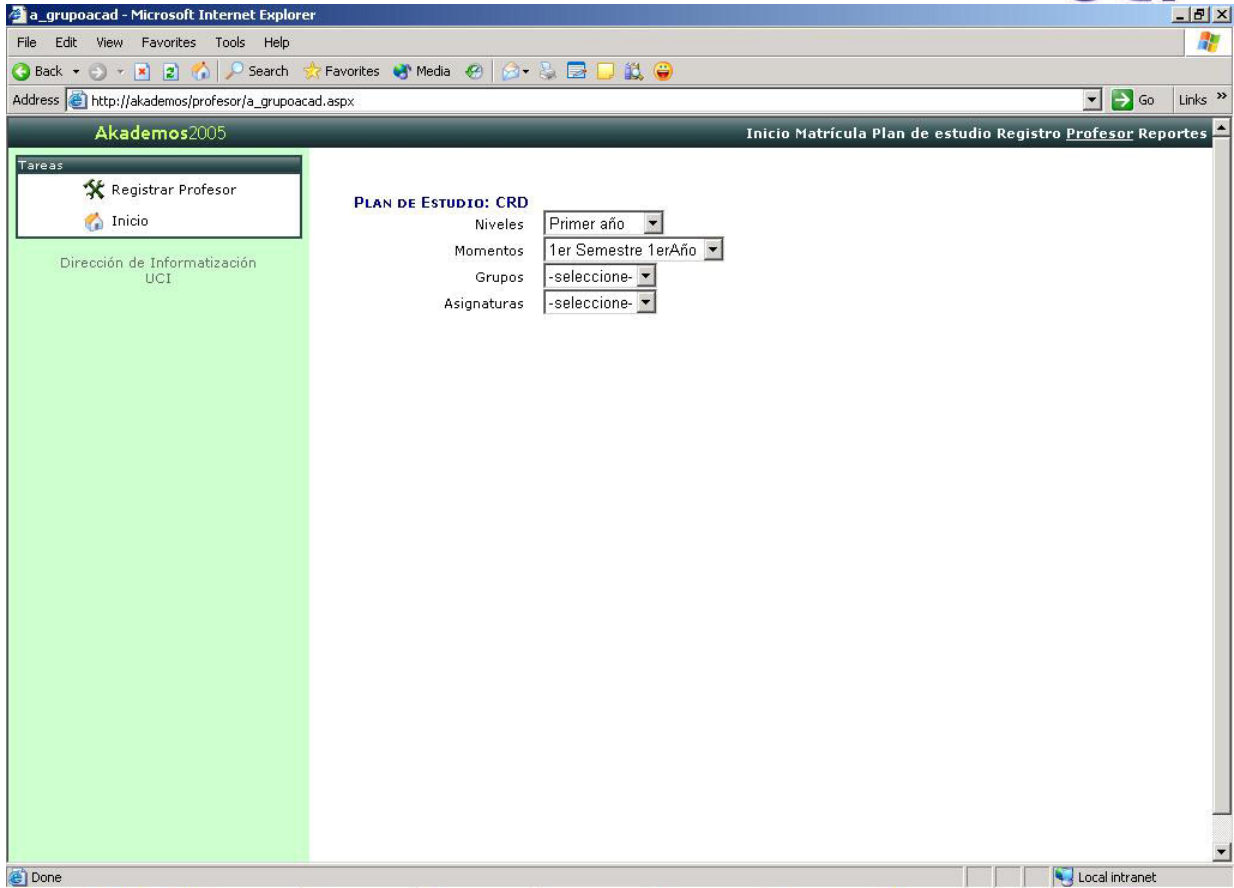
Foto	Nombre	C.I	
	FRANKLIN TEODORO AVILA DEDIEU	42120722542	+
	FRANK POMPA SOURD	64052401248	+
	FRANK ABEL ORTA CEDEÑO	79072507724	+
	FRANK DAVID SANCHEZ BARRANCO	79121816448	+
	Frank Valdez Foyaca	86091605469	+
	Frank Isabel Vega Bonne	87050206784	+
	Eduardo Frank Macheco Barzaga	79102123446	+

32 Estudiantes

Foto	Nombre	C.I	
	FRANKIEL HERNÁNDEZ DEL PINO	86072306187	+
	Franklin Yannier Garmendia Lopez	84110111901	+
	Frank Yunier Estevez Rodriguez	83071325628	+
	Frank Benavides Dalmendray	82032008824	+
	Frank Puig Placeres	82021006843	+
	Frank Ernesto Estevez Sanchez	85080118109	+
	Frank González Fernandez	85032101221	+

Done Local intranet





## Anexo 2



The screenshot shows a web browser window with the URL `http://akademos/profesor/default.aspx`. The page title is "Akademos2005" and the navigation menu includes "Inicio", "Matrícula", "Plan de estudio", "Registro", "Profesor", and "Reportes".

On the left side, there is a sidebar with the following sections:

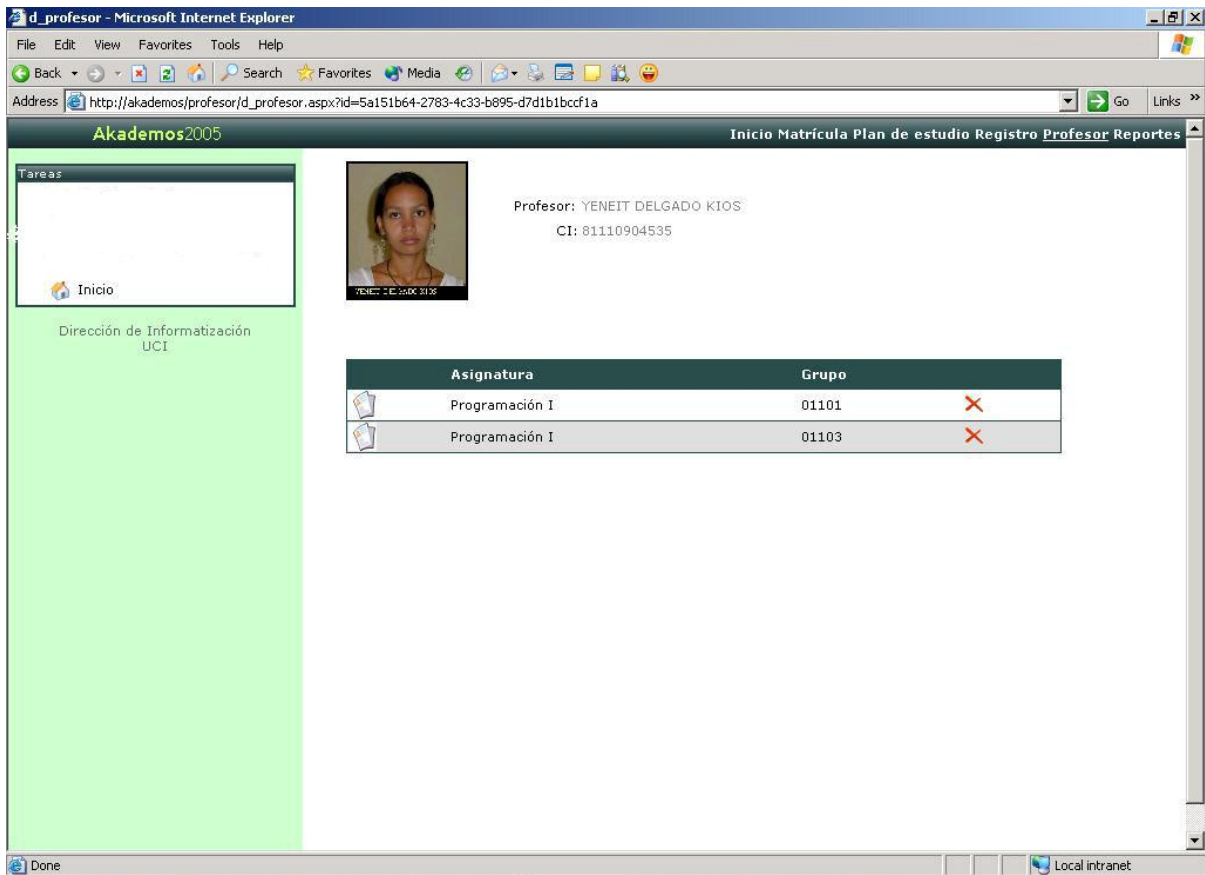
- Plan de estudio activo:** A dropdown menu currently showing "CRD".
- Tareas:** A button labeled "Registrar Profesor".
- Estructuras:** A section titled "Facultad 1" containing a grid of icons representing different courses or departments, with IDs such as 01101, 01106, 01203, 01301, etc.

The main content area displays a table titled "PROFESORES PLANTILLA DE LA FACULTAD". The table has three columns: "Nombre", "CI", and a red "X" icon. The table lists the following faculty members:

Nombre	CI	
JUANA ELENA ACOSTA GARCIA	54081301370	X
LUIS MANUEL ALONSO MARTÍNEZ	70110400487	X
JOEL ARENCIBIA RAMIREZ	78090314764	X
Yusniel Avila Malagón	79030900245	X
EMILIO FELIPE BAÑOBRE MARTÍNEZ	53072223249	X
RICARDO JOSUE BESTARD FLORES	73082010961	X
DARIEN BETANCOURT MARTINEZ	79090103845	X
RAFAEL BRITO GONZALEZ	52102401965	X

At the bottom of the page, there is a "Local intranet" icon.

## Anexo 3



**Akademos2005** Inicio Matriculación Plan de estudio Registro Profesor Reportes

Tareas

Inicio

Dirección de Informatización UCI

Profesor: YENEIT DELGADO KIOS  
CI: 81110904535

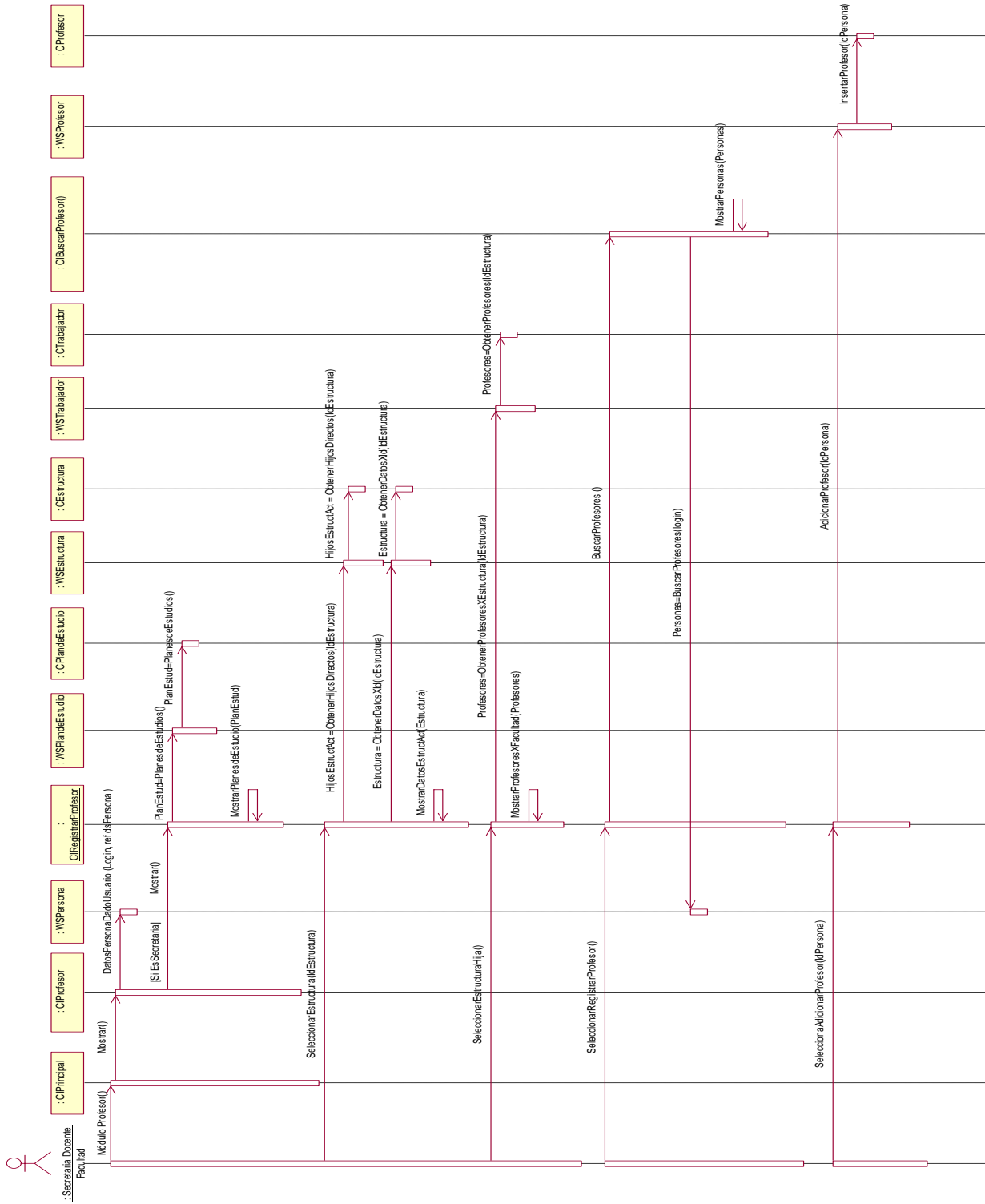
Asignatura	Grupo	
Programación I	01101	X
Programación I	01103	X

Done Local intranet



## Realización del caso de uso

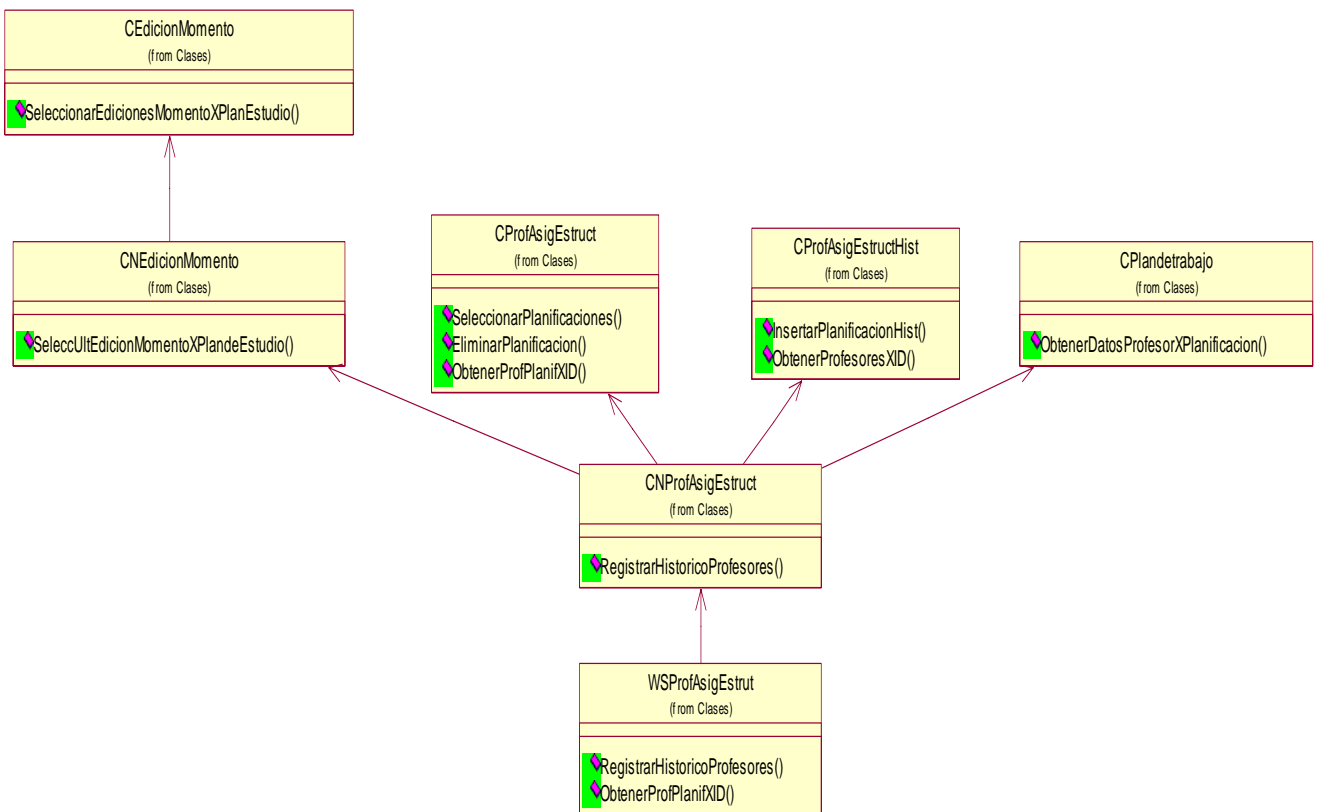




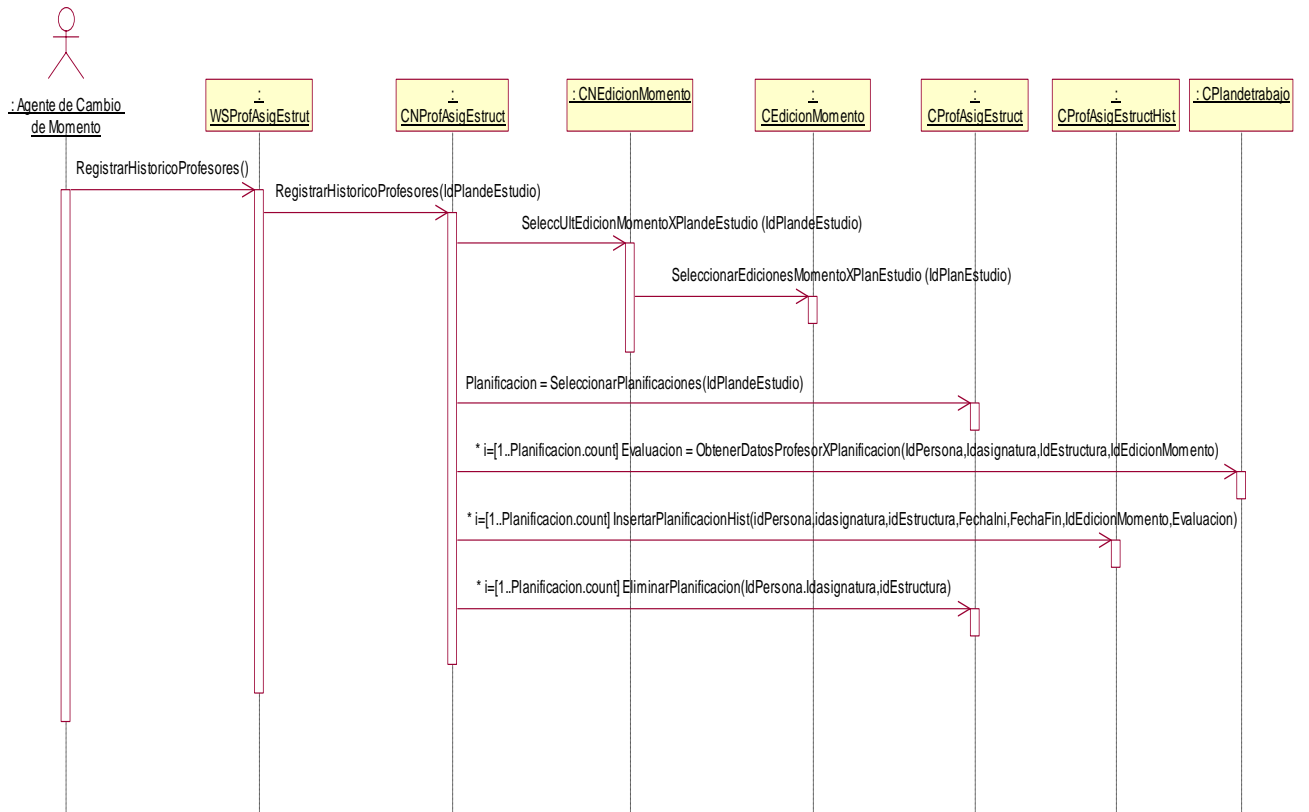
## Anexo 5

### Caso de Uso Registrar Histórico

#### Diagrama de clases



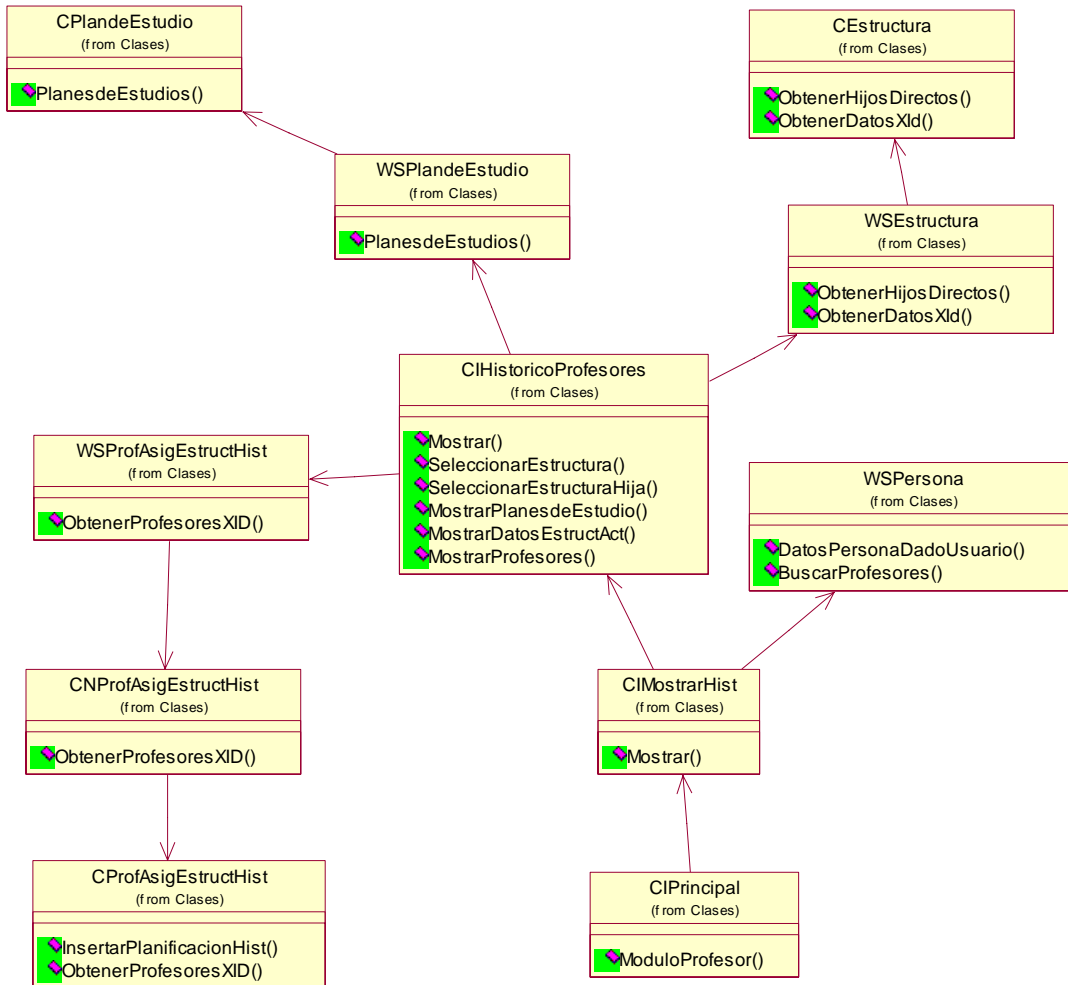
## Realización del caso de uso



## Anexo 6

### Caso de Uso Mostrar Histórico

#### Diagrama de clases

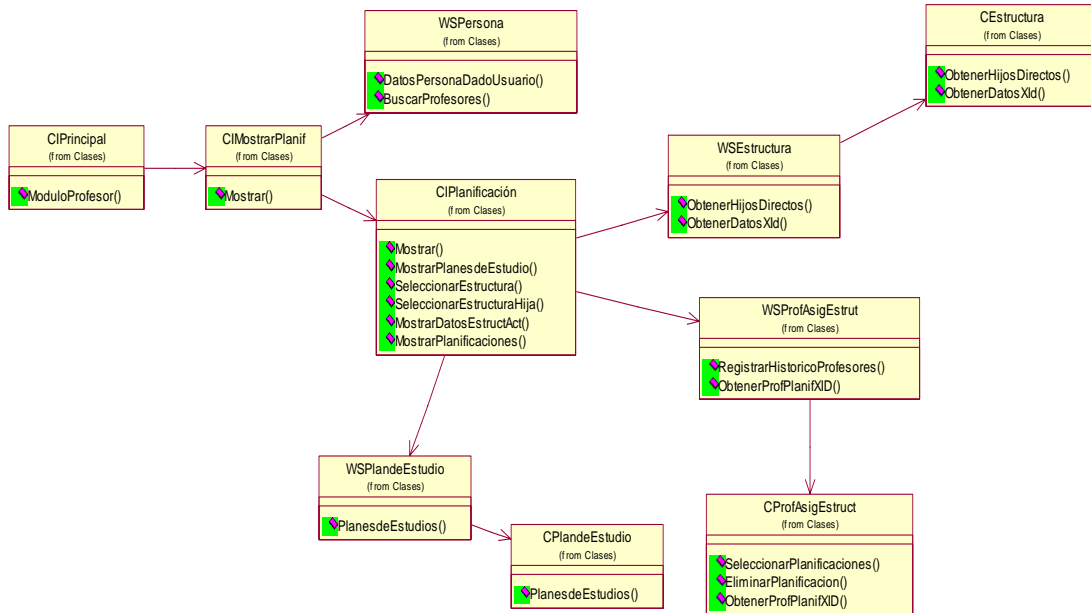




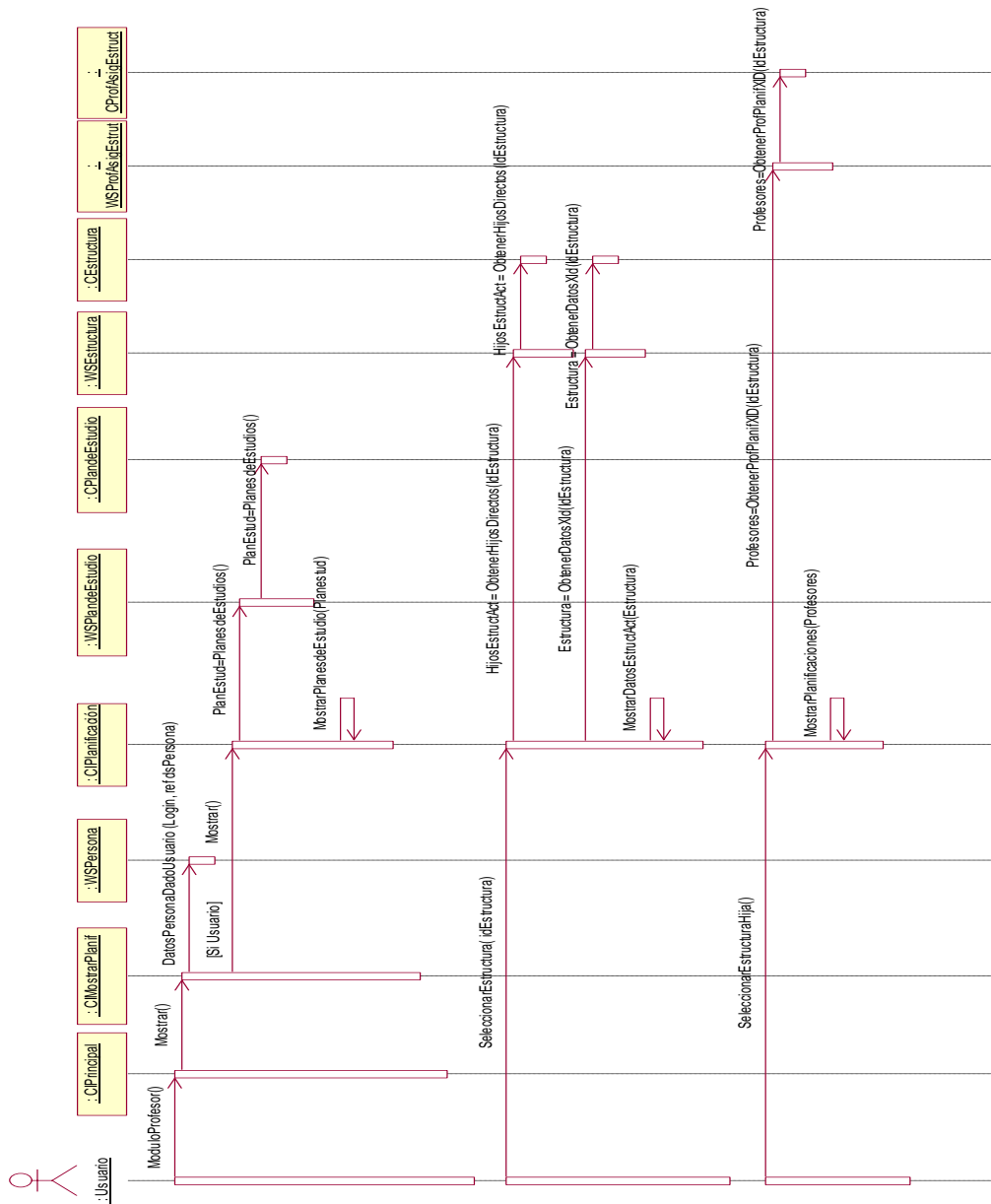
## Anexo 7

### Caso de Uso Mostrar planificación del profesor

#### Diagrama de clases

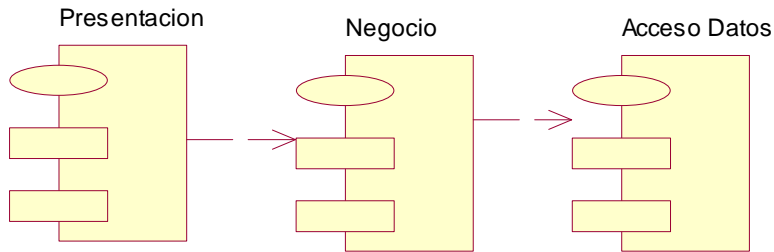


# Realización del caso de uso

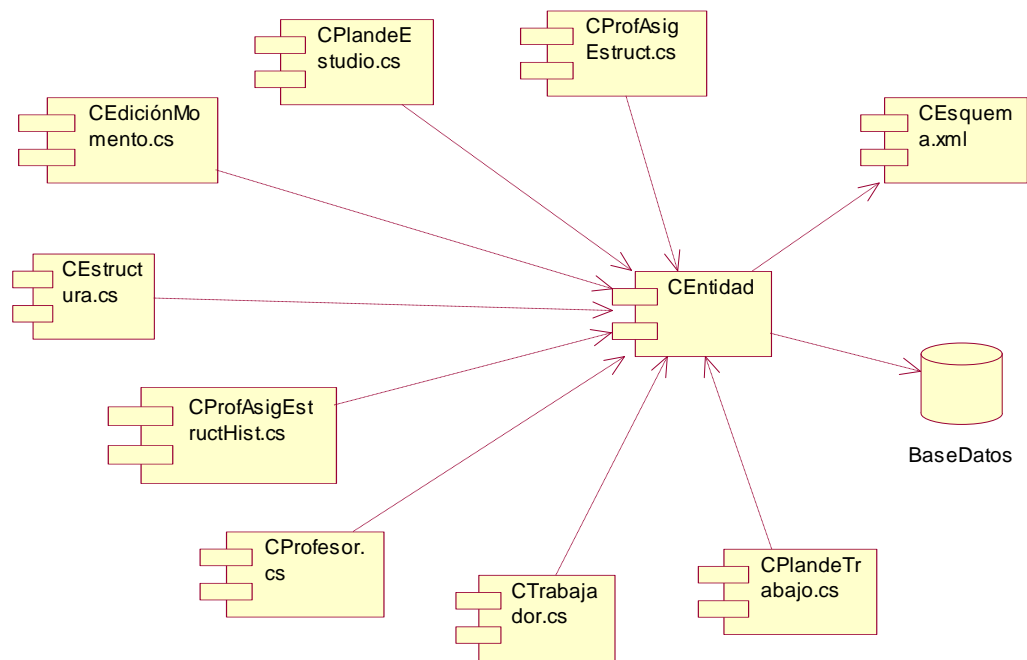


## Anexo 8

### Diagrama de Componentes



### Paquete Acceso a Datos







# Paquete Presentación

