

**UNIVERSIDAD NACIONAL DE INGENIERIA**  
**FACULTAD DE INGENIERIA INDUSTRIAL Y DE SISTEMAS**  
**SECCION DE POSGRADO**



**“ANÁLISIS Y DISEÑO DE UN SISTEMA DE GESTIÓN DE  
INFORMACIÓN ACADÉMICA DE UNA UNIVERSIDAD”**

**TESIS**

**PARA OPTAR EL GRADO ACADÉMICO DE MAESTRO EN CIENCIAS  
CON MENCIÓN EN INGENIERÍA DE SISTEMAS**

**ELABORADO POR**

**LIC. MARÍA ESTHER ARGANDOÑA CALIXTO**

**ASESOR**

**MBA. JORGE GUZMÁN YANGATO**

**LIMA - PERÚ**

**Digitalizado por:**

**2012**

### ***Dedicatoria***

*A mis queridos padres por ser la luz que  
que siempre iluminan mi sendero, y  
a mi querido y recordado Bushito  
por su eterna compañía.*

### ***Agradecimientos***

*A mis padres por su apoyo constante e infinito,  
a mis profesores por su dedicación entregada,  
por guiarnos y brindarnos sus experiencias  
para alcanzar nuestros objetivos.*

## **INDICE**

Contenido	
Resumen	
Abstract	
Descriptores	
Introducción	
	<b>Pág.</b>
<b>Capítulo I: Planteamiento de la investigación</b>	<b>1</b>
1.1 Formulación del problema	2
1.1.1 Estructura de una universidad según la ley universitaria	6
1.1.2 Problemas y deficiencias	10
1.2 Objetivos de la tesis	11
1.2.1 Objetivo general	11
1.2.2 Objetivo específico	11
1.3 Justificación y delimitación de la investigación	12
1.3.1 Justificación	12
1.3.2 Delimitación	13

<b>Capítulo II: Marco teórico</b>	<b>14</b>
2.1 Sistemas de Información	15
2.1.1 Definición de Sistemas de Información	15
2.1.2 Análisis y Diseño de Sistemas de Información	15
2.2 Métodos de Análisis y Diseño de Sistemas de Información	16
2.2.1 Método Estructurado	16
2.2.2 Método Orientado a Objetos	17
2.2.2.1 Características	18
2.2.2.2 Beneficios	19
2.3 Lenguaje de Modelado Unificado – UML	20
2.3.1 Definición	20
2.3.2 Características	21
2.3.3 Diagramas	21
2.3.3.1 Los diagramas estructurales	21
2.3.3.2 Los diagramas de comportamiento	22
2.4 Metodología RUP: Proceso Unificado de Rational	24
2.4.1 introducción	24
2.4.2 Historia	24
2.4.3 Características del Proceso Unificado	28
2.4.4 Estructura del Proceso Unificado	34
2.4.4.1 Actividades ( Workflows )	36
2.4.4.2 Fases de la metodología RUP	50
2.4.4.2.1 Fase de inicio	50
2.4.4.2.2 Fase de elaboración	54
2.4.4.2.3 Fase de construcción	58
2.4.4.2.4 Fase de transición	62

2.4.4.3 Comparación crítica del método análisis y diseño estructurado con el método orientado a objetos con la metodología RUP	63
2.4.4.4 Evaluación crítica y mejoras de la Metodología RUP	65
<b>Capítulo III: Análisis y Diseño de un Sistema de Gestión de Información Académica de una Universidad utilizando la metodología RUP</b>	<b>69</b>
3.1 Fase de inicio	70
3.1.1 Plan de proyecto	70
3.1.2 Flujo de trabajo de requisitos	70
3.1.2.1 Relación de requisitos candidatos	71
3.1.2.2 Requisitos funcionales representados en el Modelo de Negocios Inicial	71
3.1.2.3 Relación de requisitos no funcionales	74
3.1.3 Flujo de trabajo de análisis	74
3.1.4 Flujo de trabajo de diseño	77
3.1.4.1 Diagrama de clases	78
3.1.5 Relación inicial de riesgos y descripción de cómo mitigarlos	80
3.1.5.1 Relación inicial de riesgos	80
3.1.5.2 Plan de contingencia	80
3.2 Fase de elaboración	80
3.2.1 Flujo de trabajo de requisitos	80

3.2.1.1 Requisitos iniciales	80
3.2.2 Flujo de trabajo de análisis	86
3.2.2.1 Diagrama de clases	86
3.2.2.2 Modelo dinámico inicial	88
3.2.2.3 Extracción de clase borde	90
3.2.2.4 Extracción de clase control	92
3.2.2.5 Diagrama de paquetes	94
3.2.3 Flujo de trabajo de diseño	95
3.2.3.1 Refinamiento de los casos de uso	95
3.2.3.2 Refinamiento del diagrama de clases	115
3.2.3.3 Realización de los casos de uso	117
3.2.3.4 Diagrama de subsistemas	145
3.2.3.5 El modelo de despliegue (nodos)	146
3.2.4 Flujo de trabajo de implementación	147
3.2.4.1 Presentación de un prototipo del sistema de información	147
3.2.5 Flujo de trabajo de prueba	156
<b>Capítulo IV: Análisis de Costo Beneficio</b>	<b>157</b>
4.1 Factibilidad económica	158
4.1.1 Análisis de costo-beneficio	158
<b>Capítulo V: Análisis de Resultados</b>	<b>166</b>

<b>Conclusiones y Recomendaciones</b>	<b>169</b>
Conclusiones	169
Recomendaciones	171
Glosario de Términos	172
Bibliografía	174
Anexos	176

## **DESCRIPTORES**

1. Académica
2. Análisis y Diseño
3. Casos de Uso
4. Gestión
5. Objetos
6. RUP

## **RESUMEN**

El presente trabajo de investigación consiste, primero, en determinar el estado de las universidades peruanas en la gestión de su información de su información académica, y segundo en el análisis y diseño de un sistema para la gestión de información académica de una universidad (pública o privada) cuya estructura organizativa se ciña a lo establecido por el reglamento de la Ley Universitaria 23733.

La Universidad Nacional Mayor de San Marcos, por ceñirse al referido reglamento ha sido tomada como referencia para el presente trabajo; sin embargo los resultados del trabajo son aplicables a cualquier universidad cuya organización esté basada en el mencionado reglamento de la Ley Universitaria.

El análisis y el diseño son orientados a objetos y fueron efectuados con la metodología RUP y la herramienta Rational Rose.

Finalmente se presenta un prototipo del sistema de gestión académica.

## **ABSTRACT**

This research work has two major aims. The first one is to find the state in which peruvian universities are in what concerns to academic information management. The second one is to make the analysis and design for an information system to manage academic information of universities whose organizational structure is as defined by the Peruvian University Law 23733.

University **Universidad Nacional Mayor de San Marcos**, since its organizational structure is as defined by the university law, has been taken as reference model for this work. Nevertheless, the results of the work are applicable to any university whose organizational structure is as defined by the Peruvian university law.

The analysis and design of the information system are object oriented and was made applying RUP methodology and using the Rational Rose Tool.

Finally, a prototype of the information system is developed.

## INTRODUCCION

Al realizar un estudio de investigación referente a la gestión académica de las universidades se ha encontrado que en el país existen aproximadamente 102 universidades, y en una muestra de 25 universidades se observó que 13 de ellas no cuenta con un sistema de información que automatice las tareas de la gestión académica tales como registro de datos de docentes, asignación de carga lectiva, control de docentes en los semestres académicos, entre otros. El trabajo está enfocado en el análisis y el diseño de un sistema de información que de solución a los problemas anteriormente mencionados. El análisis y el diseño han sido efectuados con el enfoque orientado a objetos y aplicando la metodología RUP. Sin embargo, la investigación no se ha limitado a la simple aplicación de la metodología, sino que también se realizado una evaluación critica en cuanto a su aplicabilidad y eficacia en sistemas de tamaño mediano y pequeño (2.4.4.4).

El capítulo I presenta el planteamiento de la investigación; el capítulo II contiene el marco teórico; el capítulo III consiste la aplicación de la metodología RUP en el Análisis y Diseño de un Sistema de Gestión de Información para una Universidad aplicable a todas las universidades que se ciñan al reglamento de la Ley Universitaria 23733, el capítulo IV presenta el análisis de costo beneficio, y en el capítulo V se hace un análisis de resultados.

El presente trabajo pretende ser de utilidad para aquellas universidades que todavía no cuentan con un sistema de gestión académica y cuya estructura organizativa se base por el reglamento de la mencionada Ley Universitaria.

El análisis y diseño presentados abarcan las fases de inicio y elaboración de la metodología RUP, por lo que una universidad interesada en contar con el sistema de gestión tendría que continuar y completar con las fases de construcción y transición de la metodología.

## **CAPÍTULO I**

### **PLANTEAMIENTO DE LA INVESTIGACIÓN**

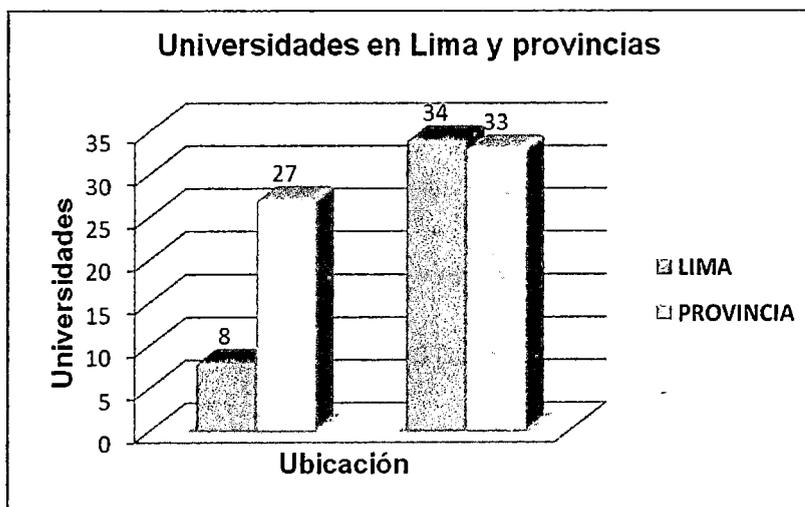
Se inicia con estudio del estado en que se encuentra con la formulación del problema, se hace la investigación de la estructuras de las universidades de nuestro país, observándose que la Universidad Nacional Mayor de San Marcos se ciñe al reglamento de La Ley Universitaria 23733, por lo que se toma como referencia para el desarrollo del trabajo, posteriormente se presentan los problemas y deficiencias, los objetivos y finalmente la justificación y delimitación del trabajo de investigación.

## 1.1 Formulación del problema.-

En la actualidad en nuestro país existen aproximadamente 102 universidades, 42 universidades en Lima y 60 universidades en provincia (información de la Asamblea Nacional de Rectores - ANR) como se muestra en las siguientes Tabla I.1 y Figura I.1:

UNIVERSIDADES	ESTATALES	PRIVADAS	TOTAL	TOTAL%
LIMA	8	34	42	41.17%
PROVINCIA	27	33	60	58.82%
TOTAL	35	67	102	100.00%

**Tabla I.1 Población de las universidades en Lima y provincias**



**Figura I.1 Representación gráfica de la población de las universidades en Lima y provincias**

Para la encuesta se considero un tamaño de muestra de 25 universidades proporcional a los datos de la población, como se observa en la Tabla I.2:

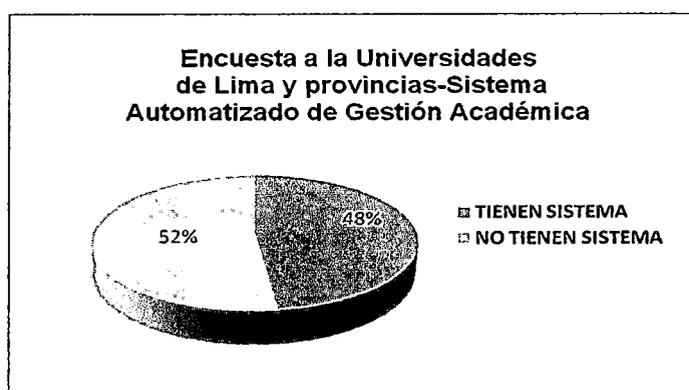
UNIVERSIDADES	ESTATALES	PRIVADAS	TOTAL
LIMA	2	8	10
PROVINCIA	7	8	15
TOTAL	9	16	25

**Tabla I.2 Muestra proporcional para la encuesta en Lima y provincias**

El resultado obtenido en la encuesta realizada fue el siguiente, como se muestra en la Tabla I.3 y Figura I.2:

UNIVERSIDADES	LIMA	PROVINCIA	TOTAL	TOTAL %
TIENEN SISTEMA AUTOMATIZADO G. A.	8	4	12	48%
NO TIENEN SISTEMA AUTOMATIZADO G.A.	2	11	13	52%
TOTAL	10	15	25	100%

**Tabla I.3 Resultado de la encuesta en Lima y provincias**



**Figura I.2 Representación gráfica del resultado de la encuesta en Lima y provincias**

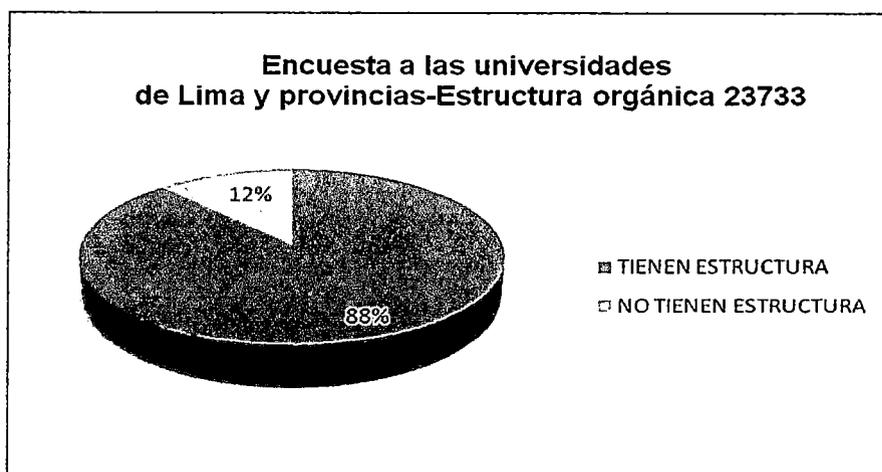
Como se observa en la Tabla I.3 de 25 universidades 13 de ellas, es decir el 52%, no cuenta con un sistema de información para la gestión académica, en el Anexo 01 se muestra el cuestionario de preguntas tomada en la encuesta. En estas universidades las tareas de la gestión académica son realizadas manualmente y la información es almacenada en archivos locales sin ninguna integración. Esta situación requiere una solución para la mejora del servicio que brindan nuestras universidades (públicas o privadas).

Se entiende por Sistema de Gestión Académica a los procesos integrados de designación de carga lectiva, almacenamiento de datos e información sobre las actividades de los docentes, actualización del plan de estudios, etc.

Como parte de la investigación se ha revisado la Ley Universitaria del Perú N° 23733, cuyos Artículos 10, 11, 12, 13 y 14, establecen que las universidades deben estar formadas por Facultades, Escuelas, Institutos, Escuelas de Postgrado, Departamentos Académicos, oficinas académicas, oficinas administrativas, oficinas de asesoramiento, entre otros. Se ha verificado que una buena parte de las universidades se ciñen a la estructura establecida por esta Ley según la encuesta el 88%, como se observa en la Tabla I.4 y Figura I.3.

UNIVERSIDADES	LIMA	PROVINCIA	TOTAL	TOTAL %
<b>TIENEN ESTRUCTURA ORGANICASEGÚN LA LEY 23733</b>	8	14	22	88%
<b>NO TIENEN ESTRUCTURA SEGÚN LA LEY 23733</b>	2	1	3	12%
<b>TOTAL</b>	10	15	25	100%

**Tabla I.4 Resultado de la encuesta en Lima y provincias**



**Figura I.3 Representación gráfica del resultado  
de la encuesta en Lima y provincias**

UNIVERSIDADES	PÚBLICAS	PRIVADAS	TOTAL	TOTAL %
TIENEN ESTRUCTURA ORGÁNICA SEGÚN LA LEY 23733	9	13	22	88%
NO TIENEN LA ESTRUCTURA SEGÚN LA LEY 23733	0	3	3	12%
<b>TOTAL</b>	<b>9</b>	<b>16</b>	<b>25</b>	<b>100%</b>

**Tabla 1.5 Resultado de la encuesta en Lima y provincias**

Entre estas tenemos a la Universidad Nacional Mayor de San Marcos, la Universidad Nacional del Callao, la Universidad Privada San Martín de Porres, etc. Sin embargo, hay también universidades que se han apartado de esta estructura. Puesto que la Universidad Nacional Mayor de San Marcos tiene una estructura organizativa basada en la ley universitaria, el presente trabajo la toma como su referencia o modelo de universidad.

La Universidad Nacional Mayor de San Marcos, conocida como la Universidad Decana de América, fue fundada el 12 de mayo de 1551, con su sede en Lima.

Desarrolla sus actividades académicas y administrativas en la Ciudad Universitaria y en otros locales del Cercado de Lima, Callao y el interior del país.

### **1.1.1 Estructura de una universidad según la ley universitaria.-**

En el Anexo 02 se presenta la estructura detallada de una universidad de acuerdo a lo establecido por la Ley Universitaria 23733.

En esta sección se presentan las entidades de dicha estructura organizativa que directamente están relacionadas con la tesis, tales como: Facultad, Departamento y Escuela Académico Profesional.

**La Facultad.-** Es un órgano responsable de las actividades académicas administrativas, de perfeccionamiento, investigación, de extensión universitaria y promoción de la cultura. Goza de autonomía de gobierno académica, económica y administrativa en el desarrollo de sus actividades dentro de la legislación vigente y de las disposiciones de los Órganos de Gobierno de la Universidad. Una Facultad agrupa especialidades o carreras afines.

Tiene como funciones principales:

- Coordinar las actividades académicas en pregrado y postgrado.
- Programar las actividades académicas para el dictado de clases.
- Ejecutar las actividades académicas de cada semestre.
- Controlar y evaluar el desarrollo de las actividades académicas.
- Promover y desarrollar la capacitación del docente y del administrativo.
- Coordinar entre facultades requerimientos de docentes.

Cada Facultad tiene la siguiente estructura orgánica siguiente:

- a. Órganos de Dirección y Gobierno
  - i. El Consejo de Facultad
  - ii. El Decano
  - iii. Comité de Gestión Académico Administrativo.
  - iv. docente y administrativo de la facultad en función de las necesidades académicas y de gestión administrativa.
- b. Órganos de Asesoramiento
- c. Órganos de Apoyo
  - i. Secretaria Académica
  - ii. Secretaria Administrativa
- d. De los Órganos de Apoyo Académico
  - i. Departamentos Académicos
- e. De los órganos de Línea
  - i. Escuelas Académico Profesionales
  - ii. Unidad de Postgrado
  - iii. Instituto de Investigación
  - iv. Centro de Extensión Universitaria y Proyección Social

En el Anexo 03 se presenta como ejemplo el organigrama de la Facultad de Ciencias Contables.

Pasaremos a describir a continuación los órganos más importantes de una Facultad.

**El Departamento Académico.-** Es la unidad de servicios de la facultad que reúne a docentes que cultivan la misma disciplina o disciplinas afines. Existe un Departamento Académico por cada especialidad. Coordinan la actividad académica de sus miembros y determinan y actualizan los syllabus de acuerdo con los requerimientos curriculares de la facultad. Los Departamen-

tos sirven a una o más facultades según su especialidad, y se integran a una Facultad sin pérdida de su capacidad funcional, según lo determine el Estatuto de la Universidad. Al Jefe del Departamento Académico también se le conoce también con el nombre de Coordinador Académico.

El Departamento Académico tiene las siguientes funciones:

- Atender requerimientos de profesores para el dictado de cursos de su facultad y los requerimientos de otras facultades. Es decir, brinda servicios al interior y exterior de la facultad a que pertenece.
- Recibir disponibilidad horaria y preferencias de cursos de los docentes para el Semestre Académico a iniciarse.
- Coordinar y atender sugerencias y pedidos de docentes hechos por Directores de las Escuelas Académicas Profesionales de la Facultad y Escuelas de Posgrado.
- Tener actualizado el cuadro de asignación de carga lectiva de la Facultad del Semestre Académico a iniciarse, que es elevado a Consejo de Facultad para su aprobación.
- Asignar la carga lectiva de cursos a los docentes de la Facultad.
- Controlar y registrar por medio de un historial las actividades del docente durante el Semestre Académico.
- Coordinar con el Instituto de Investigación y Centro de Extensión Universitaria y Proyección Social.

**Escuela Académico Profesional.-** Es la única unidad operativa a través de la cual se cumple la finalidad esencial de la formación profesional en una disciplina. Existe una Escuela Académica Profesional por cada carrera profesional. Tiene las siguientes funciones:

- Elaborar la Currícula o Plan de Estudios de la Escuela y elevarlo al Consejo de Facultad.
- Informar al consejo de Facultad sobre las actividades académicas de los docentes que sirven a la Escuela.
- Proponer al Consejo de Facultad el número de vacantes para el Concurso de Admisión.
- Conocer y coordinar los syllabus de los cursos que conforman el Plan de Estudios.
- Atender y resolver los problemas de los estudiantes de la Facultad.
- Presentar al Departamento Académico los requerimientos y sugerencias de docentes para los cursos del Semestre Académico a iniciarse.

**Unidad de Postgrado.-** Es un órgano que está sujeto funcionalmente a la Escuela de Postgrado de la Universidad con la finalidad de garantizar la armonía y coherencia de requisitos y exigencias para los estudiantes de segunda especialización, maestría y doctorado. Tiene las siguientes funciones:

- Implementar y ejecutar las políticas y normas que establezca la Escuela de Postgrado de la Universidad.
- Mantener coordinación permanente con la Escuela o Escuelas de Postgrado.
- Solicitar al Departamento o Departamentos Académicos de la Facultad los requerimientos de docentes para el dictado de clases.
- Planificar, organizar y dirigir los estudios de Segunda Especialización.
- Proponer los lineamientos sobre criterios técnico pedagógicos a aplicarse en la Unidad.

### **1.1.2 Problemas y deficiencias.-**

Habiendo estudiado diferentes universidades se han observado los siguientes problemas recurrentes:

- Los datos e información de los docentes en la Facultad se registran manualmente y la información se almacena en archivos no integrados de Word y Excel, etc.
- Las facultades de la universidad no cuentan con un sistema integrado. Cada facultad maneja de manera local los datos e información de sus actividades académicas.
- La disponibilidad de los docentes con respecto a sus horarios y preferencias de cursos se registra manualmente en el Departamento Académico.
- Los requerimientos de docentes en la Facultad y de otras Facultades se registran manualmente.
- El Cuadro de Horarios del Semestre Académico es elaborado por el Coordinador o Jefe del Departamento Académico usando Word, Excel, entre otros, en base al historial del docente que se archiva en documentos manuales.
- La elaboración y entrega de cargas lectivas por el Coordinador Académicos a los docentes se realiza manualmente con archivos como el Word y Excel.
- La información de los docentes durante cada Semestre Académico, sobre avances de cursos, asistencia a los cursos, participación en Semina-

rios, Talleres, Congresos entre otros y estudios de capacitación se registra manualmente.

- El Sistema Único de Matricula (SUM) es un sistema que permite realizar las matricula de los alumnos en cada semestre académico, para lo cual se deben tener actualizado los planes de estudios de las Facultades. Sin embargo, actualmente las actualizaciones se hacen actualmente y son enviados al área respectiva por medio de un oficio.

## **1.2 Objetivos de la tesis.-**

### **1.2.1 Objetivo general.-**

- Elaborar el análisis y diseño de una solución automatizada que permita la gestión académica de una Universidad que agilice los procesos de las Facultades.

### **1.2.2 Objetivo específico.-**

Diseñar un sistema informático que permita realizar de manera automatizada las siguientes tareas de gestión académica de una universidad.

1. Registro de datos de los docentes.
2. Registro de los planes de estudio de cada Facultad.
3. Registro de las evaluaciones de docentes por encuesta, asistencia a las clases programadas, etc.
4. Registro de disponibilidad horaria de los docentes.
5. Control de las actividades de los docentes.
6. Distribución de cargas lectivas.

7. Interacción con las Áreas de Personal (registro de docente y administrativo de la Universidad) y el SUM(Sistema Único de Matricula de la Universidad).
8. Generar reportes, informes de docentes de las actividades académicas de los docentes de los semestres académicos.
9. Emisión de informes de cargas lectivas aprobadas por Consejo de Facultad.
10. Mantenimiento en la base de datos de docentes información referente a capacitaciones, talleres congresos, grados o títulos obtenidos, etc.

### **1.3 Justificación y delimitación de la investigación.-**

#### **1.3.1 Justificación.-**

Al tener automatizadas las tareas de la gestión académica se obtendrán los siguientes beneficios:

- a) Agilizar la distribución de cargas lectivas a los docentes.
- b) Permitirá el intercambio de información entre diferentes Facultades, requerimientos de docentes a través de pedidos o solicitudes.
- c) Tener el registro o historial de los docentes que permita actualizar sus datos, así como sus evaluaciones, encuestas por cada semestre.
- d) Mejorará la calidad de la enseñanza por medio de asignación de cursos en base al historial del docente.
- e) Hay un número considerable de universidades que no cuenta con el sistema de gestión de información académica que podrían beneficiarse con los resultados de la tesis.

### **1.3.2 Delimitación.-**

El análisis y diseño desarrollado en el trabajo es aplicable a universidades con una estructura organizativa por la Ley 23733 (Anexo 04). Se ha tomado como referencia a la Universidad Nacional Mayor de San Marcos.

El trabajo se limita lo que tradicionalmente se llama al análisis y diseño del sistema; las otras etapas del ciclo de vida del sistema no forman parte de este trabajo y tendrán que ser completadas por las universidades interesadas.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

En este capítulo se presentan definiciones y conceptos sobre sistemas de información. Se hace un estudio del análisis y diseño de sistemas del método estructurado y el método orientado a objetos. Se presentan las características y beneficios de cada método. Se hace una descripción del Lenguaje de Modelado Unificado-UML. Se realiza un estudio analítico de la Metodología RUP, su estructura, actividades y fases que la componen. Finalmente se presenta una evaluación crítica y mejoras de la metodología RUP.

## **2.1 Sistemas de Información.-**

### **2.1.1 Definición de Sistemas de Información.-**

Un sistema de información es un conjunto de componentes de hardware, software, datos, personas, procedimientos, actividades que interactúan entre si para un o varios objetivos. El sistema de información recibe datos e instrucciones, procesa los datos de acuerdo con las instrucciones y produce los resultados.( Davis, 1987).

En los sistemas de información se presentan dos tipos de modelos:

- El modelo básico del sistema de información: entrada, proceso y salida.
- El modelo básico del sistema de información con almacenamiento: entrada, proceso, salida y almacenamiento de datos. El almacenamiento de datos es necesario para uso posterior de los datos.

### **2.1.2 Análisis y Diseño de Sistemas de Información.-**

El análisis y diseño de sistemas se refiere al proceso de examinar la situación de una organización con el propósito de mejorarla con métodos y procedimientos más adecuados.

El análisis del sistema comprende la recopilación de datos e interpretación de los hechos (levantamiento de información), estudio en detalle del sistema actual, diagnosticar problemas con la finalidad de mejorarlo y sugerir o recomendar las especificaciones necesarias para el nuevo sistema.

El diseño del sistema consiste en determinar los elementos que permitan cumplir con los requerimientos presentados en el análisis. Los elementos que se deben producir son: la estructura de datos necesaria, archivos o tablas (base de datos) para el almacenamientos de datos, las salidas (informes, reportes, panta-

llas) como resultado del procesamiento de datos, procedimientos para cálculos, y dispositivos de almacenamiento.

## **2.2 Métodos de Análisis y Diseño de Sistemas de Información.-**

Dentro del análisis y diseño de sistemas de información existen dos tipos de métodos más conocidos el Método Estructurado y el Método Orientado a Objetos. En la actualidad Método Estructurado no es muy utilizado, mientras que el Método Orientado a Objetos está siendo utilizado por los desarrolladores de Sistemas de Información, y también se ha observado que algunas organizaciones hacen una mixtura de ambos métodos. A continuación se describen ambos métodos.

### **2.2.1 Método Estructurado.-**

El método estructurado conocido como método tradicional, paradigma tradicional, metodología estructurada y paradigma estructurado, apareció en la década de 1970, se caracteriza porque el desarrollo del Sistema de Información se realiza de manera secuencial. Para el desarrollo del Sistema de información requiere de cinco fases más utilizadas:

- i. Fase de requisitos
- ii. Fase de análisis
- iii. Fase de diseño
- iv. Fase de implementación
- v. Fase de mantenimiento

El método consiste en desarrollar cada fase de manera secuencial, es decir se inicia con la fase de requisitos, luego se continúa con la fase de análisis y así sucesivamente hasta terminar con la última fase de mantenimiento.

A continuación se describirá cada fase:

- i. **Fase de análisis.-** Consiste en la recopilación de datos, diagnosticar problemas, analizar el sistema actual a fin de mejorarlo o cambiarlo por un nuevo sistema. Como producto se debe obtener los requerimientos del nuevo sistema.
- ii. **Fase de diseño.-** Se obtiene como producto los elementos de diseño del nuevo sistema que debe satisfacer los requerimientos obtenidos en la fase del análisis.
- iii. **Fase de implementación.-** Es el desarrollo del diseño en base a los módulos presentados en la fase de análisis utilizando un Lenguaje de Programación apropiado, Manejador de Base de Datos adecuado y cualquier otro programa necesario para el desarrollo de esta fase.
- iv. **Fase de mantenimiento.-** Luego de instalarse el sistema de información, es más probable que necesite modificarse, puede que presente errores o fallas, también se puede presentar algún requerimiento que no se haya considerado en la fase de análisis.

**2.2.2 Método Orientado a Objetos.-** El método análisis y diseño orientado a objetos aparece durante la década de 1980, se conoce como el paradigma orientado a objetos, debido a que el método tradicional solo funcionaba para sistemas de información de pequeña escala, mientras que el método orientado a objetos permite funcionar también para sistemas de información de mediana escala y grande escala. (Schach, 2005).

**2.2.2.1 Características.-** El Análisis y Diseño Orientado a Objetos tienen las siguientes características según (Martín, 1992):

- Cambian nuestra forma de pensar sobre los sistemas. Para la mayoría de las personas, la forma de pensar orientada a objetos es más natural que las técnicas del análisis y diseño estructurado. El mundo está formado de objetos, y desde nuestra infancia descubrimos que existen objetos y que sus comportamientos son diferentes.
- Los sistemas suelen construirse a partir de objetos ya existentes. Esto lleva un alto grado de reutilización, a un ahorro de dinero, un menor tiempo de desarrollo y una mayor confiabilidad del sistema.
- La complejidad de los objetos que podemos utilizar sigue en aumento, puesto que nuevos objetos se construyen a partir de otros. Estos a su vez están contruidos por otros objetos.
- El depósito Case debe contener una creciente biblioteca de tipos de objetos, algunos comprados y otros contruidos en casa. Es muy probable que estos tipos de objetos sean más poderosos conforme crezca su complejidad.
- La creación de sistemas con un funcionamiento correcto es más fácil con las técnicas orientadas a objetos. Se debe en parte, a que las clases orientada a objetos están diseñadas para reutilizarse; y en parte, a que las clases están autocontenidas y divididas en métodos.
- Las técnicas Orientadas a Objetos se ajustan de manera natural a la tecnología Case. Existen ciertas herramientas elegantes y poderosas para la

implantación orientada a objetos. Otras herramientas Case necesitan ciertas mejoras para controlar el análisis y diseño orientado a objetos.

**2.2.2.2 Beneficios.-** El Análisis y Diseño Orientado a Objetos presenta muchos beneficios, pasaremos a mencionar y detallar los más importantes.(Martín, 1992).

- **Reutilización.** Las clases están diseñadas para que se utilicen en muchos sistemas. Para maximizar la reutilización, las clases se construyen de modo que se puedan adaptar.
- **Estabilidad.** Las clases diseñadas para una reutilización repetida se vuelven estables de la misma manera que los microprocesadores y otros chips se hacen estables. Las aplicaciones se construyen a partir de chips de software cuando sea posible.
- **El diseñador piensa en términos del comportamiento de objetos y no en detalles de bajo nivel.** El encapsulado oculta los detalles y hace que las clases complejas sean fáciles de utilizar.
- **Se construyen clases cada vez más complejas.** Se construyen clases a partir de otras clases, las cuales a su vez se integran mediante clases.
- **Confiabilidad.** Es probable que el software construido a partir de clases estables ya probadas tenga menos fallas que el software elaborado a partir de cero.

- **Un diseño más rápido.** Las aplicaciones se crean a partir de componentes ya existentes. Muchos de los componentes están contruidos de modo que se puedan adaptar para un diseño particular.
- **Integridad.** Las estructuras de datos sólo se pueden utilizar con métodos específicos. Es importante en los sistemas cliente-despachador y los sistemas distribuidos, en los que los usuarios desconocidos podrían intentar el acceso al sistema.
- **Programación más sencilla.** Los programas se conjuntan a partir de piezas pequeñas, cada una de las cuales, en general, se pueden crear fácilmente. El programador crea un método para una clase a la vez.
- **Mantenimiento más sencillo.** El programador encargado del mantenimiento cambia un método de clase a la vez. Cada clase efectúa sus funciones independientemente de las demás.
- **Independencia de diseño.** Las clases están diseñadas para ser independientes del ambiente de plataformas, hardware y software.

## **2.3 Lenguaje de Modelado Unificado - UML. -**

### **2.3.1. Definición.-**

El Lenguaje Unificado de Modelado (Unified Modeling Language) es un Lenguaje estándar para escribir planos de software. Es utilizado para visualizar, especificar, construir documentar los artefactos de un sistema de software (Booch, G.,1999).

Es un lenguaje que permite visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software.

### **2.3.2 Características.-**

- Es un lenguaje muy expresivo utilizado en la documentación en el desarrollo de Sistemas de Información.
- Es sólo un lenguaje, es parte de un método de desarrollo de software.
- Es un lenguaje fácil de aprender.
- Es adecuado en sistemas de información empresarial, sistemas en tiempo real y en aplicaciones web.

### **2.3.3 Diagramas.-**

Un diagrama es la representación gráfica de un conjunto de elementos y la mayoría de las veces se dibuja como un grafo conexo de nodos (elementos) y arcos (relaciones). Cada diagrama es una proyección gráfica de los elementos que configuran un sistema.

Lenguaje Unificado de Modelado incluye trece tipos de diagramas, clasificados en dos grandes grupos: uno para modelar la estructura estática del sistema y otro para modelar el comportamiento dinámico.

**2.3.3.1 Los diagramas estructurales.-** Se emplean para visualizar, especificar, construir y documentar los aspectos estáticos de un sistema.

- **Diagrama de clases.-** Muestra un conjunto de clases, interfaces, colaboraciones y relaciones. Son los diagramas más comunes en el modelado

de sistemas orientado a objetos. Abarcan la vista de diseño estática de un sistema.

- **Diagrama de objetos.-** Muestra un conjunto de objetos y sus relaciones. Se utilizan para describir estructuras de datos, instantáneas estáticas de instancias de los elementos existentes en los diagramas de clases.
- **Diagrama de componentes.-** Representa las partes internas, los conectores y los puertos que implementan un componente.
- **Diagrama de estructura compuesta.-** Muestra la estructura interna de una clase o una colaboración.
- **Diagrama de despliegue.-** Muestra un conjunto de nodos y sus relaciones, se utilizan para describir la vista de despliegue estática de una arquitectura.
- **Diagrama de paquetes.-** Muestra la descomposición del propio modelo en unidades organizativas y sus dependencias.
- **Diagrama de artefactos.-** Muestra un conjunto de artefactos y sus relaciones con otros artefactos y con las clases a las que implementan, se utilizan para mostrar las unidades físicas de implementación del sistema.

**2.3.3.2 Los diagramas de comportamiento.-** Se emplean para visualizar, especificar, construir y documentar los aspectos dinámicos de un sistema.

- **Diagrama de casos de uso.-** Muestra
- un conjunto de casos de uso, actores y sus relaciones. Organizan y modelan el comportamiento de un sistema.

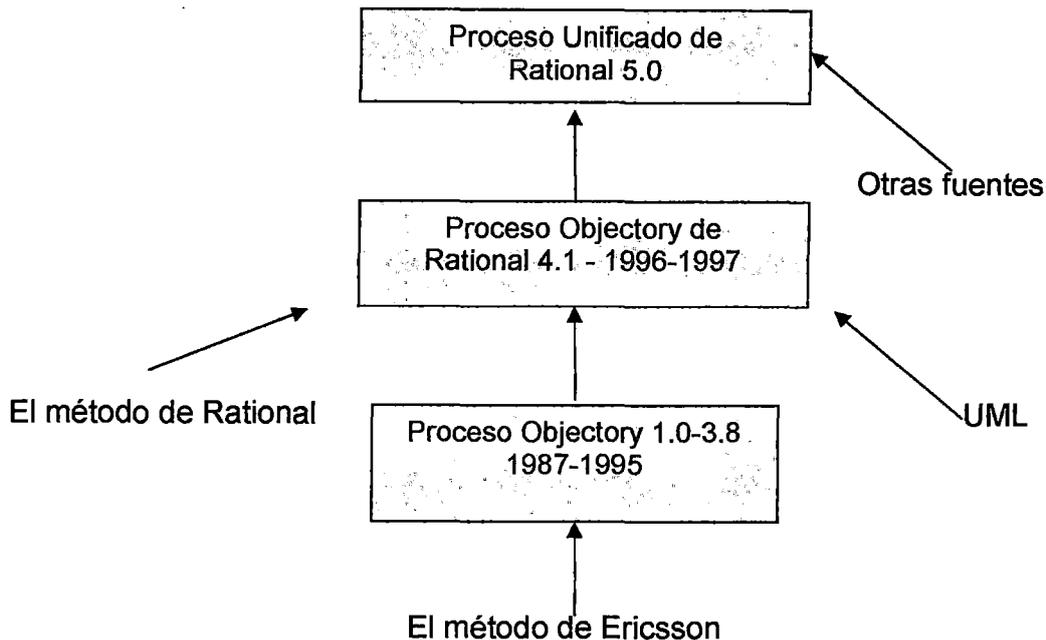
- **Diagrama de secuencia.-** Es un diagrama de interacción que resalta la ordenación temporal de los mensajes, presenta un conjunto de roles y los mensaje enviados y recibidos por las instancias que interpretan los roles. Se utilizan para describir la vista dinámica de un sistema.
- **Diagrama de estados.-** Representa una maquina de estados, que consta de estados, transiciones, eventos y actividades. Se utilizan para describir la vista dinámica de un sistema.
- **Diagrama de comunicación.-** Es un diagrama de interacción que resalta la organización estructural de los objetos que envían y reciben mensajes.
- Muestra un conjunto de roles, enlaces entre ellos y los mensajes enviados y recibidos por las instancias que interpretan esos roles. Se utilizan para describir la vista dinámica de un sistema
- **Diagrama de actividades.-** Muestra un conjunto de acciones, el flujo secuencial o ramificado de acción en acción, y los valores que son producidos o consumidos por las acciones. Se utilizan para cubrir la vista dinámica de un sistema.
- **Diagrama de colaboración.-** Muestra los objetos, así como los mensajes, numerados en el orden en el cual se envían en un escenario específico.

## **2.4 Metodología RUP: Proceso Unificado de Rational.-**

**2.4.1 Introducción.-** El trabajo se ha centrado en el análisis y diseño del sistema de información para la gestión académica de una universidad. En la sección 2.1.2.2 se ha descrito de manera general el análisis y diseño orientado a objetos. En esta parte del trabajo presentaremos un método que permite el desarrollo completo de un sistema de software: la metodología RUP (Rational Unified Process) o UP (Unified Process). La metodología RUP utiliza como herramienta y complemento el UML (Unified Modeling Language) para representar parte y transición. La descripción de la metodología que se presenta es una abstracción y síntesis de varios textos, libros, así como de información de internet, los cuales están indicados en la bibliografía. Sin embargo se debe resaltar que los autores toman como referencia al texto de (Booch,1999) para las definiciones, conceptos, modelos de análisis, modelos de diseño, etc.

#### **2.4.2 Historia.-**

El Proceso Unificado es un producto final de tres décadas de desarrollo y uso práctico. En la Figura II.2 se observa que se inicia con el proceso Objectory, que fue la primera publicación en 1987. Posteriormente paso por el Proceso Objectory de Rational que fue publicado en el año de 1997 hasta el Proceso Unificado de Rational publicado en el año 1998. El desarrollo del Proceso Unificado ha recibido influencia de los métodos de Ericsson y de Rational, y de otras fuentes que se explican a continuación.



**Figura II.3 El desarrollo del Proceso Unificado**

En el año 1967, Ericsson modelaba el sistema entero como un conjunto de bloques interconectados (en UML se le conoce subsistemas y se implementan mediante componentes). Ensamblaba los bloques de más bajo nivel en subsistemas de más alto nivel, haciéndolo al sistema más manejable. Identificaban los bloques estudiando los casos de negocio, ahora se conoce como casos de uso. El método que utilizaban era el que ahora se conoce como “desarrollo basado en componentes”, siendo su creador Ivar Jacobson.

En 1976 se publicó por CCITT, organismo internacional para la estandarización en el área de las telecomunicaciones del Lenguaje de especificación y Descripción (Specification and Description Language, SDL) para el comportamiento funcional de los sistemas de telecomunicación. El estándar fue influenciado por el método de Ericsson: el sistema se presentaba como un conjunto de bloques in-

método de Ericsson: el sistema se presentaba como un conjunto de bloques interconectados que se comunicaban unos con otros solo a través de mensajes, llamados señales.

El SDL era una estándar de modelado de objetos especializado que se actualizó periódicamente. En la actualidad es usado por más de 10,000 desarrolladores y recibe soporte de muchos fabricantes de herramientas. Se desarrollo hace 20 años, cuando el modelado de objetos no había madurado. Finalmente fue sustituido por UML, que se estandarizó en 1997.

En 1987 Ivar Jacobson dejó Ericsson y fundó Objectory AB en Estocolmo, después de ocho años junto con sus colaboradores desarrollaron un proceso denominado Objectory, es una abreviatura de Object Factory, fábrica de objeto. El concepto de caso de uso estuvo presente en el trabajo de Ericsson, sin embargo aquí el concepto se hizo más claro. La experiencia en el desarrollo de Objectory aportó ideas sobre cómo diseñar los procesos generales en un negocio.

En 1995 Rational Software compra Objectory AB, y unificó los principios de desarrollo Rational. Las contribuciones más importantes al proceso fueron los énfasis en la arquitectura y en el desarrollo iterativo.

Se añadieron la experiencia y prácticas de Rational para formar el proceso Objectory de Rational 4.1, adicionando las fases y la aproximación iterativa controlada y la arquitectura se hizo más explícita. El Proceso Objectory de Rational utilizó UML en todos sus modelos.

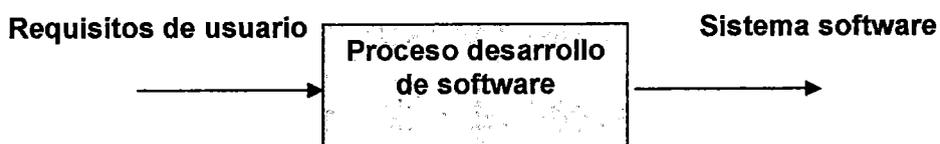
En 1998, en la mitad del año, el proceso Objectory de Rational se había convertido en proceso que podía soportar el ciclo de vida completo del desarrollo de software. En Junio de este mismo año Rational publica una versión del pro-

ducto: el Proceso Unificado de Rational 5.0. En el año 1999 los tres amigos Booch, Jacobson y Rumbaugh, publicaron una metodología completa del análisis y diseño orientado a objetos que unificara sus tres metodologías separadas. El Proceso Unificado de Rational (RUP), tenía este nombre al principio, era porque los tres amigos trabajan en ese entonces para Rational, Inc.. En el libro que ellos publicaron, se usó el nombre de Proceso Unificado de Desarrollo de Software, tal es así que actualmente se conoce como Proceso Unificado, una forma de abreviar el nombre.

En resumen, el proceso Unificado se formó a partir de muchas fuentes que fueron aportando en el transcurrir de los años, entre ellos los tres amigos. (Booch, Jacobson y Rumbaugh).

En las siguientes secciones se hace una definición de la metodología, se explica las características principales, se presenta la historia de la metodología y detalla la estructura del Proceso Unificado.

**Definición de RUP (Rational Unified Process).**- Es un proceso de desarrollo de software, constituido por el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software (Booch, 1999), la Figura II.2 se muestra la definición de manera gráfica.



**Figura II.2 Un proceso de desarrollo de software**

El Proceso Unificado se puede utilizar en los sistemas de software, en diferentes áreas, variedad de organizaciones y en proyectos de diversos tamaños.

El Proceso Unificado está basado en componentes, es decir que el sistema de software en construcción está formado por componentes de software interconectados a través de interfaces.

El Proceso Unificado utiliza el Lenguaje Unificado de Modelado descrito en la sección 2.2 (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema de software o sistema de información. UML es una parte muy importante en el Proceso Unificado.

**IBM Rational Rose 7.0 Enterprise.-** Es una herramienta para modelado del análisis y diseño Orientado a Objetos que permite dar soporte al UML(Lenguaje Unificado de Modelado), es el producto más completo de la familia Rational Rose. Se considera como la mejor elección para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Siendo un producto de Rational Rose, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente.

#### **2.4.3 Características del Proceso Unificado.-**

El Proceso Unificado tiene tres características principales:

- i) Dirigido por casos de uso.
- ii) Centrado en la arquitectura.
- iii) Iterativo e incremental.

Estas características son detalladas a continuación.

- i. **Dirigido por casos de uso.-** Los casos de uso guían a través de todo el proceso de desarrollo de software, en las actividades o flujos de trabajo requisitos, análisis, diseño, implementación y prueba, desarrollándose dentro de las cuatro fases : inicio, elaboración, construcción y transición. En el desarrollo de un sistema de software una de las actividades iniciales es la obtención de los requisitos, y deben ser representados de tal manera que sean fácil de comprender para los usuarios, clientes y desarrolladores; esta representación se hace a través de los casos de uso, y los actores que representan a los usuarios que pueden ser personas u otro sistema que interactúa o hace uso del sistema.

En un sistema los actores interactúan con los casos de uso, un caso de uso es una secuencia de acciones que el sistema los realiza para entregar un resultado a un actor. El modelo de caso de uso está compuesto por todos los actores y casos de uso del sistema, este modelo captura todos los requisitos funcionales del sistema, según (Booch, 1999, p. 39), se define un caso de uso:

*“Un caso de uso especifica una secuencia de acciones, incluyendo variantes que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto”.*

Finalmente, podemos concluir que los casos de usos dirigen y son como hilos conductores durante todo el proceso de desarrollo de un sistema, estando presentes en los flujos de trabajo: requisitos, análisis, diseño, implementación y prueba.

- ii. **Centrado en la arquitectura.-** Permite representar el sistema por medio de vistas. Las vistas consisten en representar el sistema desde diferentes perspectivas para su mejor entendimiento. El grupo de vistas repre-

sentan la arquitectura. Se tienen las vistas de los modelos de casos de uso, modelo de análisis, modelo de diseño, modelo de implementación y modelo de despliegue. La arquitectura es la visión común donde los empleados, usuarios, desarrolladores deberían estar de acuerdo y poder aceptarlo. También brinda una clara perspectiva del sistema completo, muy necesaria para tener un control del desarrollo del sistema de información. Como ejemplo podemos citar para la construcción de un edificio, el arquitecto crea los aspectos más significativos del diseño del edificio en su conjunto, se encarga de hacer los planos del edificio describiendo cada parte del edificio. En el desarrollo del sistema de información, se necesita una "arquitectura de sistemas de software", por tanto el arquitecto de software y los desarrolladores, deberán presentar el sistema desde diferentes perspectivas para entender mejor el diseño, estas perspectivas se conocen con el nombre de vistas del modelo del sistema y juntas representan la arquitectura.

La arquitectura de software según (Booch, 1999), comprende las siguientes decisiones importantes:

- La organización del sistema de software.
- Los elementos estructurales que comprendan el sistema y sus interfaces, junto con sus comportamientos, tal como se especifican en las colaboraciones entre estos elementos.
- La composición de los elementos estructurales y del comportamiento en subsistemas progresivamente más grandes.
- El estilo de la arquitectura que guía esta organización los elementos y sus interfaces, sus colaboraciones y su composición.

Habíamos mencionado que la arquitectura se representa mediante vistas del modelo: una vista del modelo de casos de uso, una vista del modelo de análisis, una vista del modelo de diseño, una vista del modelo de despliegue y una vista del modelo de implementación, la vista del modelo de prueba no se considera porque no desempeña ningún papel en la descripción de la arquitectura. Una vista es parte del modelo, por ejemplo una vista del modelo de casos de uso, es muy parecido al modelo de casos de uso, contiene los casos de uso más importantes. La vista del modelo del diseño es parecido al modelo del diseño, y también comprenden los casos de uso más importantes desde el punto de vista arquitectónico.

En los sistemas grandes y complejos se requiere de una arquitectura para lo siguiente:

- Comprender mejor el sistema.
- Organizar el desarrollo.
- Fomentar la reutilización.
- Hacer evolucionar el sistema.

El arquitecto especifica la descripción de la arquitectura. Esta descripción le permite controlar el desarrollo del sistema desde la perspectiva técnica. La arquitectura de software se centra en los elementos significativos: subsistemas, interfaces, colaboraciones, nodos y clases activas, describiendo los cimientos del sistema para desarrollarlo.

La arquitectura se obtiene durante la fase de elaboración pasando por los flujos de trabajo requisitos, análisis, diseño implementación y pruebas, utilizando los casos de uso significativos para la arquitectura entre otras entradas para la implementación de la arquitectura o el "esqueleto del sistema".

- iii. **Iterativo e incremental.-** En el desarrollo de un sistema de software es recomendable dividir el trabajo en partes o miniproyectos, en ese sentido cada miniproyecto es una iteración, y los incrementos resultan después de cada iteración, obteniéndose como resultado el crecimiento del producto.

Las iteraciones e incrementos se realizan en las cuatro fases : inicio, elaboración, construcción y transición. Una iteración es un miniproyecto, es decir pasa por casi todos los flujos de trabajo fundamentales: requisitos, análisis, diseño, implementación y prueba. Los incrementos son el resultado de una iteración. *“Un incremento es la diferencia entre la versión interna de una iteración y la versión interna de la siguiente”*. (Booch, 1999, p. 97).

El desarrollo iterativo e incremental tiene como finalidad la obtención de un software mejor, es decir:

- Tener un control de los riesgos críticos y significativos desde la fase de inicio.
- Proporcionar un marco de trabajo que gestione de la mejor manera cualquier cambio en los flujos de trabajo.
- Comenzar una arquitectura que guie el desarrollo de software.
- Construir el sistema a través del tiempo, y no al final del proyecto, porque al cambiar o modificar algo sería muy costoso.
- Proporcionar un proceso de desarrollo que permita a los desarrolladores trabajar eficazmente.

### **Manejo de riesgos.-**

Cuando se desarrolla un sistema de software información, nos enfrentamos a riesgos, los identificamos y deben de tratarse rápidamente. *“Un riesgo es una exposición que nos puede acarrear perdidas o daños. El riesgo es un factor, cosa, elemento o cambio que constituye un peligro, cuyo grado es incierto. En el desarrollo de software, podemos definir un riesgo como un asunto que tiene cierto grado de probabilidad de poner en peligro el éxito de un proyecto”.* (Booch, 1999, p. 85).

En las dos primeras fases: inicio y elaboración del Proceso Unificado, trata los riesgos importantes, y cualquier riesgo adicional al inicio de la fase de construcción, tratándolos por orden de prioridad.

Al ser identificados los riesgos, estos son tratados. El equipo de trabajo deberá contar con cuatro elecciones: evitarlos, limitarlos, atenuarlo o controlarlo.

- Pueden y deberían evitarse, mediante una replanificación del proyecto o un cambio en los requisitos.
- Deberían limitarse, restringirse de tal manera que afecte a una parte pequeña del proyecto.
- Pueden atenuarse ejercitándolos y observándolo si aparecen o no. Si aparecen nuevamente, entonces el equipo ha aprendido más sobre él, y podrá encontrar una forma de evitarlo, limitarlo o controlarlo.
- Pueden presentarse riesgos que no puedan atenuarse, en este caso lo que se debe hacer es controlarlos y observar si aparecen, de presentarse este caso el equipo debe aplicar su plan de contingencia. Como en toda actividad de la vida real, siempre se presentarán riesgos, lo importante es afrontarlos y tomar las medidas respectivas. Es

por ello que es muy importante el realizar las primeras iteraciones en las fases de inicio y elaboración porque se esperaban de alguna manera que estos riesgos podían aparecer, de no haber conocido los riesgos en el proyecto, los desarrolladores hubieran continuado con su trabajo. Sin embargo en la fase de construcción los riesgos son menores.

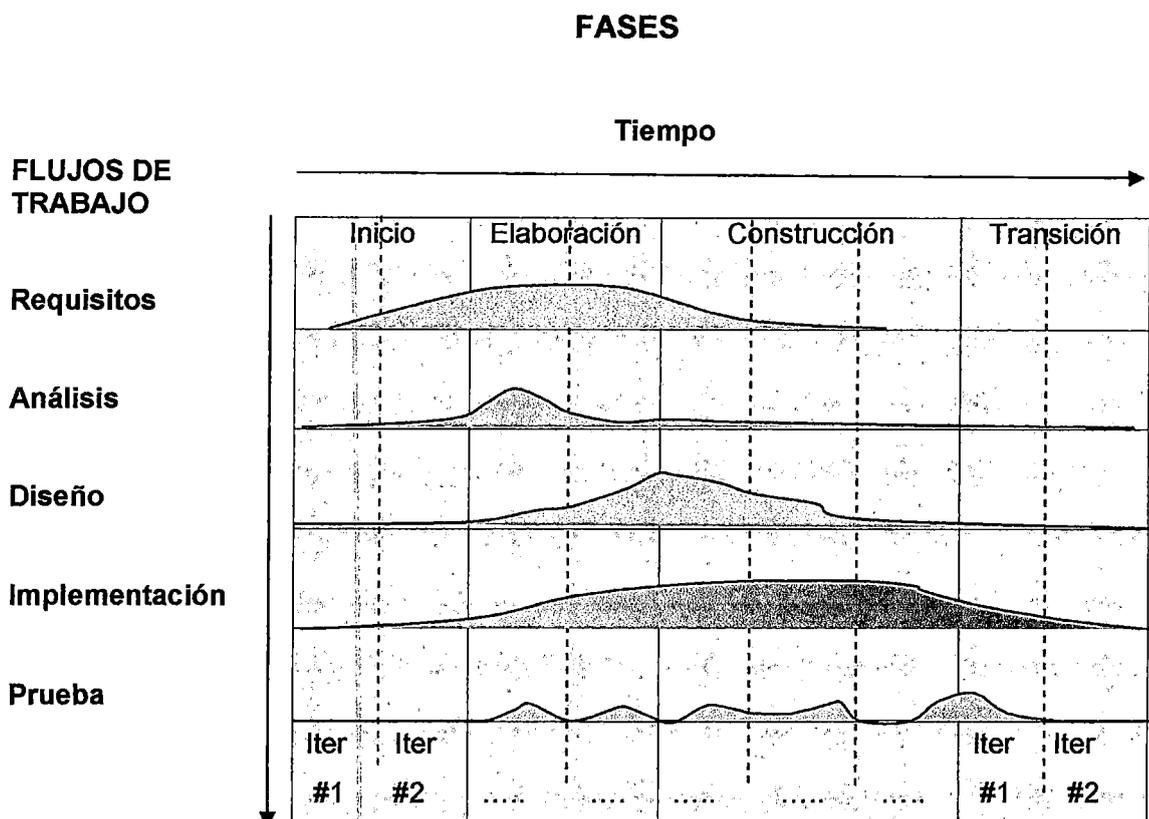
#### **2.4.4 Estructura del Proceso Unificado.-**

A continuación se va a describir la estructura de la metodología RUP basándonos en la Figura II.3. En la parte superior se muestra el ciclo de vida de un proyecto compuesta por cuatro fases: inicio, elaboración, construcción y transición, en la columna izquierda se encuentran las cinco actividades conocidas como actividades: requisitos, análisis, diseño, implementación y prueba. Cada fase puede tener una o más iteraciones, y en cada iteración se ejecutan casi siempre cinco actividades. Al completar las iteraciones de una fase se alcanza un hito. Los hitos para cada fase se dan a continuación.

- Fase inicio: el hito de los objetivos.
- Fase elaboración: el hito de la arquitectura.
- Fase de construcción: el hito de la funcionalidad.
- Fase de transición: el hito de una versión del producto.

En la Figura II.3 las curvas aproximadamente describen la cantidad de trabajo de cada actividad (workflows) en cada fase. Cada fase se divide en iteraciones o miniproyectos. Una iteración típica debe pasar por los cinco flujos de trabajo.

Estas fases son descritas en la sección 2.4.4.2.



**Figura II.3 Los cinco flujos de trabajo y las cuatro fases del RUP**

**Fuente: Schach, S. R. (2005). Análisis y Diseño Orientado con UML y el Proceso Unificado. Pág. 29.**

En el Proceso Unificado el trabajo es dividido en fases e iteraciones. Las iteraciones permiten a los desarrolladores identificar las necesidades reales de los clientes y de reducir los riesgos. La construcción por incrementos permite observar el nivel de progreso, lo que a su vez reducirá los problemas de última hora y el tiempo requerido para terminar el proyecto o el producto.

**2.4.4.1 Actividades ( Workflows).**- En cada fase del proceso se realizan cinco actividades: requisitos, análisis, diseño, implementación y prueba. A continuación describiremos estas cinco actividades conocidas como workflows.

**i. Actividad de requisitos.-**

En el desarrollo de un sistema de software se requiere de los requisitos. Un requisito especifica algo que el sistema debe realizar o cumplir. Se requiere averiguar, descubrir lo que se debe construir para el sistema, (Booch, G.,1999). Los clasifica principalmente en dos categorías: funcionales y no funcionales:

- **Requisito funcional.-** Especifica una acción que el sistema de información debe ser capaz de realizar. Se expresan con frecuencia en términos de entradas y salidas ; dada una entrada específica, el requisito funcional estipula cual debe ser la salida. Es representado como caso de uso.
- **Requisito no funcional.-** Especifica las propiedades del sistema de información mismo, como las restricciones de plataforma (“el sistema de información debe correr bajo Linux”), los tiempos de respuesta (“en promedio, la respuesta a las consultas del tipo 3B debe ser de 99.5 por ciento del tiempo”).

Los analistas de sistemas (especialistas en análisis y diseño de sistemas) son los encargados de obtener los requisitos, para ello deberán trabajar principalmente con el cliente (persona que requiere el sistema de software o el sistema de información) y los usuarios que serán quienes van a hacer uso del sistema de información.

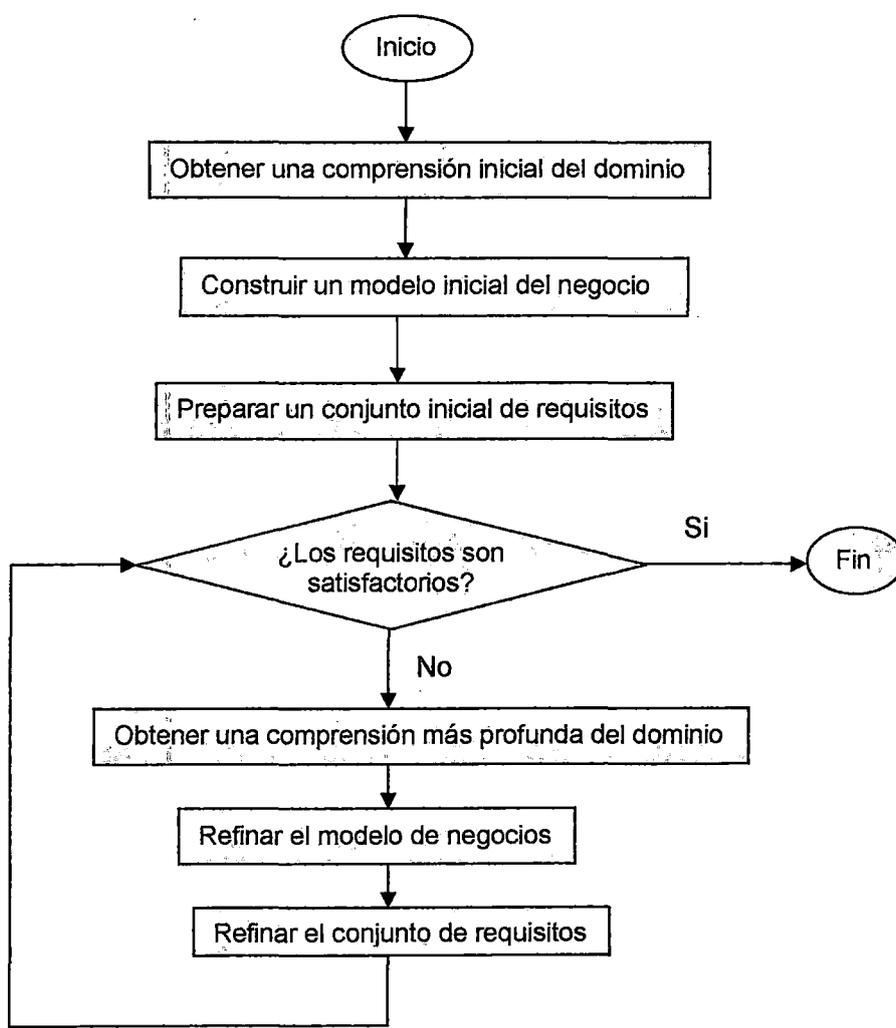
Para determinar cuáles son las necesidades del cliente, una forma directa sería preguntarle qué es lo que necesita, cuáles son sus necesidades para el sistema, sin embargo muchas veces de esta manera no se obtiene la información necesaria para que forme parte de los requerimientos que sirvan para la construcción del sistema de información. Muchas veces el cliente no es experto en tecnología de información, y con frecuencia no es fuente buena de información para indicar que se necesita desarrollar, por lo que es necesario que esta tarea lo realice un experimentado analista de sistemas. El analista de sistemas deberá seguir los pasos del proceso unificado, lo que le permitirá determinar los requerimientos del cliente, es decir lo que necesita realmente para su sistema de información.

El Proceso Unificado es iterativo según (Booch, 1999), en la Figura II.7 se muestra un diagrama de los pasos que se debe seguir con el proceso unificado, (Schach, 2005).

Observamos que el primer paso es obtener una comprensión del dominio de la aplicación o simplemente dominio, quiere decir el entorno de negocios específico en el cual va operar el sistema de información, podemos decir establecer límites al sistema en estudio. Por ejemplo un dominio puede ser Finanzas, Ventas, Logística, etc.. Cuando ya se comprende el dominio, posteriormente se construye el modelo de negocios, es decir, se usa los diagramas UML para describir los procesos de negocios del cliente. Se utiliza el modelo de negocios para determinar cuáles son los requisitos del cliente. A continuación se comienza a iterar, que quiere decir repetir para ganar mayor comprensión.

Podemos decir que se inicia una comprensión inicial del dominio y se utiliza la información para construir el modelo de negocios inicial. Luego el modelo construido nos servirá para preparar un conjunto inicial de los requisitos del cliente.

Como se observa en la Figura II.4: La pregunta ¿Los requisitos son satisfactorios? De no ser así, entonces el siguiente paso es obtener una comprensión más profunda del dominio, utilizando este conocimiento para refinar el modelo de negocios, y obtener posteriormente el conjunto de requisitos el cliente, la iteración continuara hasta que se está satisfecho con el conjunto de los requisitos, y la iteración se detendrá.



**Figura II.4 Diagrama de la actividad de requisitos**

**Figura II.3 Los cinco flujos de trabajo y las cuatro fases del RUP**

**Fuente: Schach, S. R. (2005). Análisis y Diseño Orientado con UML y el Proceso Unificado. Pág. 65.**

**Modelo de negocios.-** Un modelo es un conjunto de diagramas UML que representan uno o más aspectos del sistema de información que se quiere desarrollar. El modelo de negocios es una descripción de procesos de una empresa en términos de casos de uso del negocio y actores del negocio que corresponden con los procesos del negocio y los clientes. Por ejemplo en un banco, se tienen los procesos: aceptar los depósitos de los clientes del banco, otorgar préstamos a sus clientes, hacer inversiones, dar créditos, etc.

Un modelo de negocios proporciona una comprensión de los negocios del cliente como un todo, que le servirá al analista de sistemas tener un conocimiento del sistema de información, y le permitirá aconsejar, sugerir al cliente sobre que partes del sistema tendrá que computarizar o automatizar. En el caso de sistemas de información computarizados ya existentes, también es necesario entender el sistema como un todo para poder ampliar, modificar el sistema de información. El analista de sistemas debe de comprender bien los procesos de negocios para que pueda construir un modelo de negocios.

A continuación explicaremos las técnicas para obtener la información necesaria para construir el modelo de negocios.

**Las entrevistas.-** Es una de las principales técnicas para obtener la información necesaria. Los encargados de realizar las entrevistas se deben reunir con los miembros de la empresa y con los futuros usuarios del sistema de información.

Existen otras técnicas para obtener información para el modelo de negocios es mediante los cuestionarios, que generalmente van dirigidos a los miembros importantes de la empresa. También para obtener los requisitos se debe examinar los formularios, la documentación, utilizada por el negocio que conducirá a ob-

tener las necesidades para el sistema de información. Otra forma de obtener información, es mediante la observación directa a los usuarios, es decir que acciones realizan en sus tareas. Actualmente, se utilizan cámaras de video dentro del área de trabajo para registrar que tareas se están realizando por el personal de la empresa, habiendo solicitado permiso con anticipación.

**Casos de uso.-** Representan los requisitos funcionales. Todos los casos de usos juntos constituyen el modelo de casos de uso, el cual describe la funcionalidad total del sistema, y contesta a la pregunta ¿Qué debe hacer el sistema? (Booch, 1999). El primer diagrama UML utilizado en la elaboración de modelos de negocios es el caso de uso. Un caso de uso hace a un modelo de una interacción entre el sistema de información y los usuarios de éste (actores). Por ejemplo en la Figura II.8 se muestra un caso de uso de un sistema de información bancario, se tienen dos actores el cliente y el cajero, el texto dentro del ovalo describe la actividad de negocios representada por el casos de uso, Retiro de dinero. También se debe considerar, un caso de uso muestra la interacción entre el sistema de información y el entorno en el cual opera. Un actor es un miembro del mundo exterior al sistema, y el rectángulo en el caso de uso representa el sistema de información. Un actor muchas veces es un usuario del sistema de información. En el ejemplo mencionado del sistema de información bancario los usuarios del sistema son los clientes y el personal del banco, los cajeros, administradores, etc. Es decir que un actor desempeña un papel con respecto al sistema. Se debe adjuntar un glosario al modelo de casos de uso.

En el análisis orientado a objetos se pueden encontrar errores. Es posible que se requieran otros requisitos, como también los que se tengan ya no se apliquen. El proceso unificado es iterativo y por incrementos y se debe tener presente que los cambios y extensiones a la versión del sistema de información

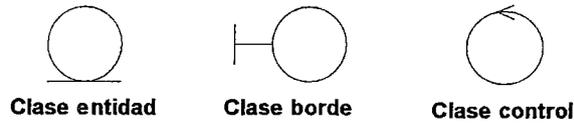
se pueden realizar en cualquier momento, el siguiente tema a tratar será la actividad de análisis.

Luego de haber obtenido los requisitos funcionales, requisitos no funcionales y el modelo de negocios, el equipo de trabajo deberá prestar atención a todos los casos de uso con la finalidad de decidir qué casos de uso deben de tomarse en cuenta en esta fase, y detallar cada caso de uso. También se debe presentar la relación de los riesgos, y como mitigarlos.

## **ii. Actividad de análisis.-**

En el análisis orientado a objetos el análisis consiste en obtener una comprensión más profunda de los requisitos, describirlos de tal manera que proporcione apoyos para la comprensión de la estructura del sistema de información.

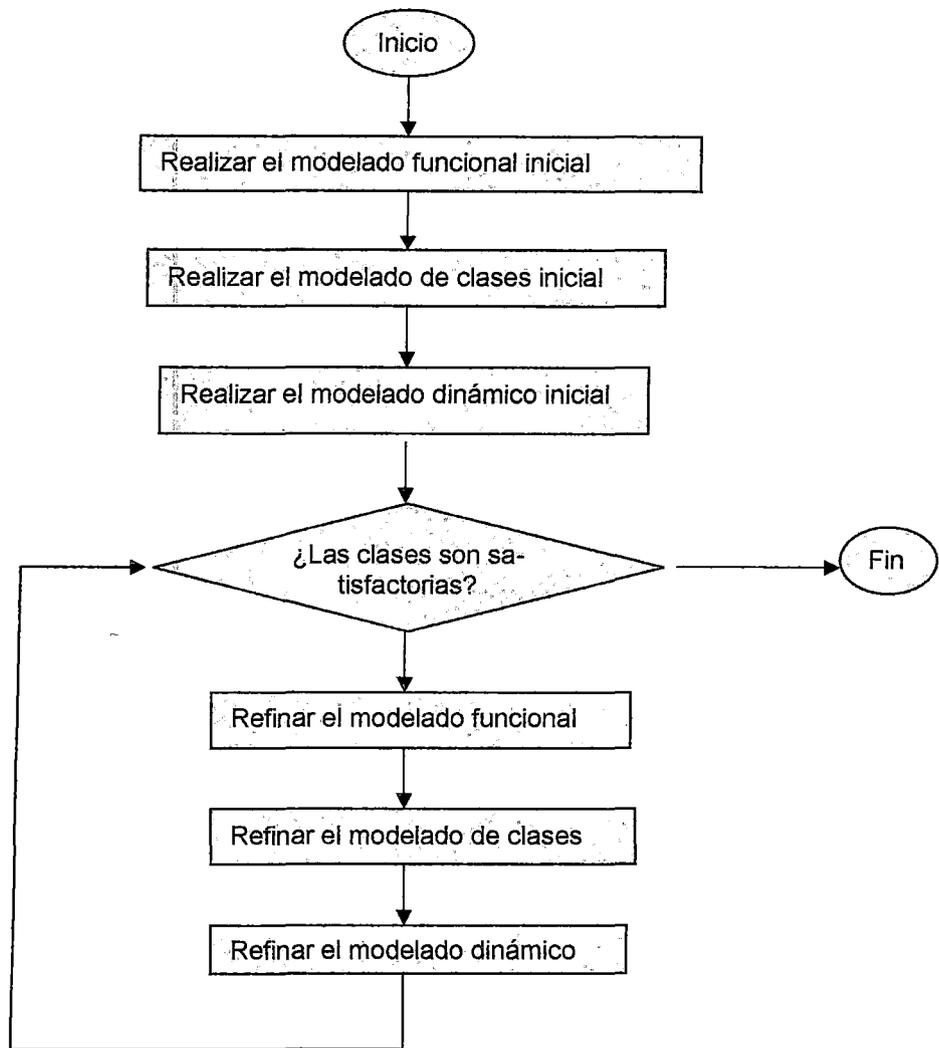
En el proceso unificado se utilizan los casos de uso, y en el análisis los casos de uso se describen en términos de las clases. Existen tres clases: entidad, borde y control, como se muestra en la Figura II.5. La clase entidad es un modelo de la información perdurable. Por ejemplo en el sistema bancario, la clase cuenta es una entidad porque la información tiene que permanecer en el sistema de información. La clase borde, permite modelar la interacción entre el sistema de información y los actores, generalmente se asocian con la entrada y salida. Por ejemplo en un sistema de Ventas, una clase borde sería Informe de los artículos vendidos en periodo dado. Es decir la clase borde Informe de los artículos vendidos es necesario en el sistema. La clase control es un modelo de los cálculos y algoritmos complejo. En el sistema de Ventas una clase control sería Clase Calcular el Monto Total de la Factura.



**Figura II.5 Estereotipos del UML(extensiones del UML), para representar una clase entidad, una clase borde y una clase control**

1. **Extracción de las clases entidad.-** Consiste en tres pasos que se llevan a cabo de manera iterativa y por incrementos, según (Schach, 2005). tenemos:
  - **Modelado funcional**, consiste en presentar los escenarios de todos los casos de uso (un escenario es una instancia de un caso de uso, del mismo modo que un objeto es una instancia de una clase).
  - **Modelado de clases**, consiste en determinar las clases entidad y sus atributos. Posteriormente definir las interrelaciones entre las clases entidad. Presentar esta información en forma de un **diagrama de clases**.
  - **Modelado dinámico**, Determinar las operaciones realizadas por o con cada clase o subclase. Presentar esta información en forma de **diagrama de estado**, que refleje todas las operaciones realizadas por o para el sistema de información.

En la Figura II.6 se muestran estos pasos, que no necesariamente deben de estar en este orden.



**Figura II.6 Diagrama de flujo para la extracción de clases entidad**

**Fuente: Schach, S. R. (2005). Análisis y Diseño Orientado con UML y el Proceso Unificado. Pág. 118.**

2. **Extracción de la clase borde.-** Las clases bordes no son difíciles de extraer. Están considerados como clases borde: la pantalla de entrada, de salida, informe impreso. En la extracción de clases inicial se tiene solo una clase de pantalla conocida como Interfaz de usuario, es decir el menú principal de la pantalla interfaz de usuario.
  
3. **Extracción de la clase control.-** Para la obtención de la clase control es muy similar a la clase borde, es fácil de obtener. En el sistema de Ventas: una clase control sería: Clase Calcular el Monto Total de una Factura.

**Refinamiento de los casos de uso.-** En la actividad de requerimientos se creó el modelo de negocios, el proceso unificado consiste en iteraciones e incrementos, si ha realizado iteraciones al diagrama de clase, probablemente se ha incrementado la cantidad de casos de uso en el modelo de negocios obtenido en la actividad de los requisitos, como consecuencia el modelo es mejorado en este paso, por lo que vendría hacer una siguiente iteración para este modelo, de ahí parte el nombre de refinamiento de los casos de uso del sistema de información.

**Realización de los casos de uso.-** Es el proceso de ampliar y refinar los casos de uso. Se utilizaran el diagrama de interacción que representa la realización de un escenario específico del caso de uso. Presenta dos tipos: **el diagrama de colaboración**, muestra los objetos, así como los mensajes, numerados en el orden en el cual envían en un escenario específico, **y el diagrama de secuencia**, es equivalente al diagrama de colaboración, muestra de mensajes sin ambigüedades. Los diagramas de interacción se desarrollan para todos los casos de uso del sistema de información.

**Incremento del diagrama de clase.-** En esta parte se une todas las diagramas de clases creados anteriormente (clase borde, clase control y clase entidad) de los casos de uso del sistema de información junto con el diagrama de clases (clases entidades).

Es importante mencionar que el proceso unificado no necesita de “**un documento de especificaciones**”, como en el paradigma tradicional, en el proceso unificado los casos de uso y los artefactos derivados de ellos contienen toda la información que aparecen en el documento de especificaciones en forma de texto.

### **iii. Actividad de diseño.-**

En esta etapa del Sistema de información, la entrada la actividad del diseño son los artefactos del análisis, estos artefactos se iteran hasta obtener un formato que se pueda programar. Es importante resaltar, la identificación de operaciones y sus asignaciones de las clases apropiadas. En el diagrama de clases se debe especificar de manera exacta los atributos y los formatos de los atributos; también se debe asignar ciertas operaciones a las clases que se deducen a partir de los mensajes en los diagramas de interacción de las realizaciones de los casos de uso.

También debe tomarse en cuenta la selección del lenguaje de programación que se utilizará para la implementación. Generalmente los sistemas de información son grandes, pero es recomendable descomponerlos en partes, es decir en subsistemas, de tal manera que la programación se hace más fácil de desarrollar. En el proceso unificado los subsistemas se obtienen a partir de los paquetes del análisis. Es más fácil programar subsistemas pequeños y relativamente independientes entre sí. Este proceso de descomposición los sistemas forma parte de la arquitectura del sistema.

Un sistema debe satisfacer los requisitos funcionales, es decir los casos de uso. También debe satisfacer a los requisitos no funcionales, incluyendo la portabilidad, la fiabilidad, la solidez, la capacidad de mantenimiento y la seguridad, teniendo en cuenta que se debe hacer dentro el presupuesto y tiempo programado.

La arquitectura de un sistema de información es un factor vital en la que se refiere a si la entrega del sistema de información será un éxito o un fracaso, ya que durante la ejecución de la actividad de diseño es que deben tomarse las decisiones críticas con respecto a la arquitectura, (Schach, 2005).

El modelo de despliegue también debe ser importante desarrollar dado este flujo de trabajo, pues será una entrada para el flujo de implementación. Este modelo, describe todas las configuraciones de redes sobre las cuales debería implantarse el sistema, (Booch, 1999).

#### **iv. Actividad de implementación.-**

El objetivo es implementar el sistema de información en Lenguaje de programación seleccionado. Es necesario que el sistema de información se divida en subsistemas más pequeños, que posteriormente se implementaran paralelamente con equipos de codificación. Los subsistemas consisten de componentes o artefactos de código, los cuales serán codificados por el programador, cuando el componente se encuentre correctamente, o no tenga errores, pasará a control de calidad, es decir a las pruebas, esta parte corresponde a la actividad de prueba, que será explicado en la siguiente sección.

Los propósitos de la implementación según (Booch, 1999), son:

- Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue.
- Implementar las clases y subsistemas encontrados durante el diseño.
- Probar los componentes individualmente, y luego integrándolos compilándolos y enlazándolos en uno o más ejecutables antes de ser enviados para ser integrados y llevar a cabo las comprobaciones del sistema.

**Artefacto Componente.-** Es el empaquetamiento físico de los elementos de un modelo como son las clases en el modelo de diseño. A continuación se indican algunos estereotipos estándar de componentes:

**Executable .-** Es un programa que puede ser ejecutado en un nodo.

**File.-** Es un fichero que contiene código fuente o datos.

**Library.-** Es una librería estática o dinámica

**Table.-** Es una tabla de base de datos

**Document.-** Es un documento.

**Artefacto Subsistema de implementación.-** Proporcionan una forma de organizar del modelo de implementación en partes más manejables. Un subsistema puede estar formado por componentes y otros subsistemas (recursivamente).

Un subsistema de implementación se manifiesta a través de un mecanismo de empaquetamiento concreto en un entorno de implementación determinado, como por ejemplo:

- Un paquete en Java.
- Un proyecto en Visual Basic.

- Un directorio de ficheros en un proyecto de C++.
- Un subsistema en un entorno de desarrollo integrado como Rational Apex.
- Un paquete en una herramienta de modelado visual como Rational Rose.

**Artefacto Interfaz.-** Las interfaces que han sido desarrolladas en el workflow del diseño pueden ser utilizadas en el modelo de implementación para especificar las operaciones implementadas por componentes y subsistemas de implementación.

#### **v. Actividad de prueba.-**

En esta actividad corresponde la verificación de la implementación, es decir lo que se ha construido. Pasaremos a describir sus tareas principales:

- Planificar las pruebas necesarias en cada iteración, incluyendo las pruebas de integración y las pruebas del sistema.
- Diseñar e implementar las pruebas creando los casos de prueba, que permiten especificar que se va a probar.
- Realizar las diferentes pruebas y manejar los resultados de cada prueba sistemáticamente.

A continuación pasaremos describir los artefactos necesarios en esta actividad.

**Artefacto Modelo de pruebas.-** Describe como se prueban los componentes ejecutable (construcciones) en el modelo de implementación con pruebas de integración y de sistema. El modelo de pruebas contiene casos de prueba, procedimiento de pruebas y componentes de prueba.

**Artefacto Caso de prueba.-** Especifica la forma de probar el sistema, se incluye la entrada o resultado con la que se ha de probar y las condiciones bajo las que ha de probarse. Existen dos casos de prueba más conocidos:

- El caso de prueba donde se especifica como probar un caso de uso o un escenario específico de un caso de uso. Es decir es la verificación del resultado de la interacción entre los actores y el sistema, es la verificación de la descripción detallada del caso de uso.
- El caso de prueba que especifica como probar una realización de caso de uso diseño o un escenario específico de la realización. Puede incluir la verificación de la interacción entre los componentes que implementan dicho caso de uso.

**Artefacto Procedimiento de prueba.-** Especifica cómo realizar los casos de prueba. Un procedimiento de prueba puede ser una instrucción al encargado de realizar las pruebas, indicarle cómo debe hacer las pruebas manualmente o interaccionar manualmente con una herramienta de automatización de pruebas para crear componentes ejecutables de prueba.

**Artefacto: Componente de prueba.-** Permite automatizar los procedimientos de prueba. Los componentes de pruebas usualmente son desarrollados con un lenguaje de programación. Los componentes de prueba son a veces llamados Test drivers, test harnesses y test scripts.

**Artefacto: Plan de prueba.-** Se describen las estrategias, recursos y planificación de la prueba.

**Artefacto Defecto.-** Documento que contiene la lista de defectos. El defecto es una anomalía del sistema, por ejemplo el error de software. Los defectos encontrados por los desarrolladores deben ser controlados y resueltos.

**Artefacto Evaluación de prueba.-** Documento que contiene los resultados de los esfuerzos es una evaluación de los resultados de los esfuerzos de prueba, como la cobertura del caso de prueba, la cobertura de código y el estado de los defectos.

Luego de haber descrito las diferentes actividades que se realizan en las fases a continuación, explicaremos las fases del Proceso Unificado, así como también los productos o entregables en cada una de ellas.

#### **2.4.4.2 Fases de la metodología RUP.-**

El proceso completo de RUP comprende cuatro fases secuenciales: inicio, elaboración, construcción y transición.

##### **2.4.4.2.1 Fase de inicio.-**

El objetivo principal de la fase de inicio es desarrollar el análisis de negocio, que permita justificar la puesta en marcha del proyecto. Los flujos de trabajo más desarrollados en esta fase son de requisitos y análisis. Para realizar el análisis de negocio se debe delimitar el ámbito del negocio, que permitirá saber que debe cubrir la arquitectura. Este análisis de negocio nos permitirá seguir adelante con el negocio. Habíamos mencionado que para lograr este objetivo se necesita conocer el ámbito del sistema propuesto, esbozar una arquitectura conocida como "arquitectura candidata", identificar los riesgos críticos para el éxito del proyecto y perfilar un plan para mitigarlos, también es necesario un prototipo del sistema, que en esta fase se conoce como "prototipo desechable".

En la fase de inicio se debe producir lo siguiente: un plan de proyecto, los cinco flujos fundamentales (requisitos, análisis, diseño, implementación y prueba), una relación de riesgos acompañada por la forma de mitigar los riesgos, y el análisis de costo beneficio del negocio.

#### **i. Plan del proyecto.-**

En el Proceso Unificado es necesario hacer un plan del proyecto para las cuatro fases: inicio, elaboración, construcción y transición, y consiste en:

- Designar a cada fase el tiempo y fechas necesarios para completar con las actividades. Los tiempos establecidos pueden ser inestables, el principio de la fase de inicio, sin embargo en la fase de elaboración ya se tendría una propuesta firme.
- Indicar cuantas iteraciones se deben realizar en cada fase. El número de iteraciones depende de la complejidad del sistema propuesto, en el caso de proyectos simples solo requerirá de una iteración en cada fase, mientras que los proyectos más complicados o grandes será necesario varias iteraciones.

#### **ii. Flujos de trabajo.-**

**Requisitos.-** En esta actividad el objetivo es de identificar, detallar y representar lo requisitos del sistema de información.

Mencionaremos los aspectos que se debe tener en cuenta:

- Enumerar los requisitos candidatos.
- Comprender el contexto del sistema.

- Representar los requisitos funcionales como casos de uso: Modelos de Negocios Inicial.
- Presentar una lista de los requisitos no funcionales.

**Análisis.-** El objetivo de este flujo es analizar los requisitos, refinarlos y estructurarlos y obtener un modelo inicial de análisis.

Se considera los siguientes análisis:

- **Análisis de la arquitectura,** consiste en clasificar los casos de uso o escenarios que se necesitan examinar cuidadosamente para el propósito de esta fase, se trata de comprenderlos y refinarlos. El arquitecto construirá una primera versión del modelo de análisis. No es necesario que sea perfecto.
- **Análisis de un caso de uso,** algunos casos de uso comparten recursos en el sistema, como base de datos, recursos computacionales, con el modelo de análisis permite mostrar esos recursos compartidos, por lo que es necesario realizar el análisis con la finalidad de resolverlos. Se pueden analizar y refinar, algunos casos de uso, es decir el 10% del total de los casos de uso, y se pueden detallar alrededor del 5% por ciento del conjunto de casos de uso.

**Diseño.-** El objetivo en este flujo es esbozar un modelo de diseño de la arquitectura candidata. También se puede desarrollar un prototipo de interfaces que solo muestre una idea, en algunos casos se puede presentar el desarrollo de un algoritmo, aunque no es necesario. Se consideran los siguientes diseños:

- **Diseño de la arquitectura**, se desarrolla un esbozo inicial del modelo de diseño es decir una arquitectura candidata. En el modelo de diseño los requisitos funcionales son representados por casos de uso, también los requisitos no funcionales representan el rendimiento, que puedan ser un riesgo.
- **Diseñar un caso de uso**, el diseño del caso de uso es mínimo.
- **Diseñar una clase y diseñar un subsistema**, si se realiza en esta fase es mínimo.

**Implementación.-** En algunos casos se construye un prototipo inicial de demostración (desechable).

**Prueba.-** Se van considerando que pruebas serán necesarias en y se van desarrollando algunos planes provisionales de prueba.

### **iii. Lista de riesgos y como mitigarlos.-**

Se presenta una lista inicial de riesgos y la forma de cómo mitigarlos.

### **iv. Análisis de costo beneficio.-**

Se realiza en esta fase para saber si continuar o no con el proyecto, es un análisis inicial. El análisis de costo beneficio es una técnica fundamental para decidir si un cliente debe computarizar su negocio y, de ser así, como lo haría. Se realizan estimaciones de costos y beneficios con la finalidad de saber si el sistema de información que se va computarizar va hacer rentable, es decir si va a tener beneficios y ahorros. Con respecto a los costos se hacen estimaciones de gastos en equipo de hardware, software, salarios del equipo de trabajo, etc. Los beneficios que se obtendrían serian tangibles e intangibles. Los beneficios tan-

gibles son más fáciles de medir, por ejemplo se puede estimar el ahorro en dólares del salario de 20 empleados, pudiendo ser reemplazados por cuatro digitadores encargados del ingreso de datos, mientras que los beneficios intangibles son más difíciles de cuantificar, una forma práctica de asignar un valor en dólares haciendo suposiciones. Cuando ya se han considerados todos los costos totales estimados y los beneficios totales, se calcula la diferencia entre los beneficios y los costos, si es mayor, entonces se continúa con el proyecto.

#### **Resumen de entregables de la fase de inicio:**

- Plan de proyecto.
- Requisitos funcionales, representado por el modelo de negocios inicial.
- Relación de requisitos no funcionales.
- Glosario del proyecto, que se adjunta al modelo de casos de uso.
- Relación inicial de riesgos, y descripción de cómo mitigarlos.
- Esquema inicial de la descripción de una arquitectura candidata, que profile las vistas de los modelos de análisis y diseño e implementación.
- Un prototipo desechable.
- Análisis de negocio: análisis de costo beneficio inicial.

#### **2.4.4.2.2 Fase de elaboración.-**

Los objetivos principales en esta fase es la recopilación de la mayor parte de los requisitos que aún están pendientes representándolos como casos de uso, establecer una base de la arquitectura sólida que permita guiar el trabajo de la fase de construcción y fase de transición, y continuar con la observación y control de los riesgos.

Las actividades que se desarrollaran en paralelo en esta fase son cuatro: los cinco flujos de trabajo; planificación de las iteraciones, evaluación de iteraciones y fases y el análisis de negocios costo beneficio.

#### **i. Flujos de trabajo.-**

**Requisitos:** En esta actividad los analistas de sistemas revisaran los casos de uso presentados anteriormente en la fase de inicio, y posteriormente identificarán los casos de uso y actores adicionales que no se han considerado. Se recomienda comprender un promedio del 80% de los casos de uso para alcanzar el objetivo de esta fase, no es necesario detallar todos los casos de uso.

También es importante determinar la prioridad de los casos de uso. El modelo de negocios inicial desarrollado en la fase de inicio, es revisado, mejorado, reestructurado, puede que se adicionen nuevos casos de uso, o se eliminen algunos casos de uso, dependiendo de los requerimientos del usuario.

**Análisis.-** En esta parte se trabajaran con los casos de uso más significativos y con los casos de uso complejo que se necesitan refinar.

Pasaremos a describir en esta parte los análisis requeridos:

- **Análisis de la arquitectura.-** Se debe extender el análisis de la arquitectura hasta que pueda servir de base a una línea base de la arquitectura ejecutable.
- **Analizar un caso de uso.-** Es probable que algunos casos de uso sean complejos y no sean comprensibles, por lo que es necesario un refinamiento. Es necesario analizar detenidamente los casos de uso principal-

mente aquellos que dependen unos de otros, es decir un caso de uso puede requerir información de entrada de otro caso de uso.

El análisis de casos de uso consiste en analizar cada caso de uso del modelo de negocios y especificar detalladamente en función de las clases y sus responsabilidades.

- **Analizar una clase.-** Consiste en refinar las clases identificadas en los pasos anteriores, uniendo responsabilidades que han sido asignado a estas clases desde diferentes casos de uso.
- **Analizar una paquete.-** El equipo de trabajo responsable de los componentes, tendrán como tarea de refinar y mantener los paquetes.

**Diseño.-** A continuación se presentan los diseños que deben ser considerados.

- **Diseño de la arquitectura** En la vista de la arquitectura del diseño se incluyen subsistemas, clases, interfaces y realizaciones de casos de uso
- **Diseñar un caso de uso.-** Los casos de uso significativos son diseñados en término de subsistemas del diseño, subsistemas de servicio y clases de diseño.
- **Diseñar una clase.-** Se diseñaran las clase que participaron en las realizaciones de caso uso en el paso anterior.
- **Diseñar un subsistema.-** Se diseñaran los subsistemas resultantes del diseño de la arquitectura.

### **Implementación.-**

- **Implementación de la arquitectura.-** El modelo de despliegue se identifican los componentes necesarios para implementar los subsistemas de servicio. Los componentes ejecutables se asignan a los nodos de la red de computadores en la que van a ejecutarse.
- **Implementación de una clase e implementación de un subsistema.-** El flujo de trabajo de diseño se diseñaron una serie de clase, muy importantes para la creación de la línea base de la arquitectura, que será una versión ejecutable inicial del sistema.

**Integrar el sistema.-** En base al pequeño grupo de casos de uso que se han implementado, se procederá a integrar los subsistemas y los componentes que corresponderán a una línea base de la arquitectura.

**Prueba.-** El objetivo es verificar que todos los subsistemas del paso anterior funcionen, lo que significa que si funciona para una parte del sistema funcionaran para el resto de sistema de información. Se consideran las siguientes pruebas: planificar las pruebas, diseñar las pruebas, realizar pruebas de integración y realizar pruebas del sistema.

### **Resumen de Productos o entregables de la fase de elaboración:**

- Modelo de negocios de los casos de uso.
- Diagramas de interacción Colaboración y estados.
- Diagrama de clases.
- Una lista de riesgos actualizada.
- Diagrama despliegue.
- Un plan de proyecto para las fases de construcción y transición.

**2.4.4.2.3 Fase de construcción.-** En esta fase se debe completar todos los requisitos, análisis y diseño y evolucionar la línea base de arquitectura obtenida en la fase de elaboración hasta obtener el producto o sistema de información alrededor del 100%. La fase de construcción puede ser considerada como el desarrollo del proyecto. Se desarrollaran las siguientes actividades en paralelo: los cinco flujos de trabajo, planificación de las actividades, el análisis de negocio: análisis de costo beneficio y la evaluación de: iteraciones y fases y planificación de la fase de transición. En la fase de elaboración, es probable que se haya implementado menos del 10% de los casos de uso, lo necesario para producir la línea base de la arquitectura. A continuación describiremos los cinco flujos de trabajo.

**i. Flujo de trabajo**

**Requisitos.-** En la fase de elaboración, se identificaron en promedio del 80% por ciento de los casos de uso y se describieron en detalle un aproximado del 20% por ciento, y se implemento menos del 10% por ciento de los casos de uso, lo necesario para producir la line base de la arquitectura. La finalidad en este flujo es identificar y detallar los requisitos en un 100%.

Se considerara las actividades siguientes:

**Encontrar actores y casos de uso,** se completara el 20% de los casos de uso.

**Estructurar el modelo de casos de uso,** es posible que pueda ser mejorado el modelo de casos de uso, los cambios que tengan que realizarse se recomiendan aquellos casos de uso que no se hayan desarrollado en la fase anterior.

**Determinar prioridades de los casos de uso.-** Para el desarrollo base de la arquitectura se clasifico algunos casos de uso, en esta parte adicionaremos a la clasificación los casos de uso, a fin de asignarles prioridad.

**Desarrollar un prototipo de la interfaz de usuario,** en las fases de inicio y elaboración no se desarrollan prototipos de la interfaces de usuario, solo se desarrollaron un nuevo tipo de interfaz o un prototipo con la finalidad de realizar una demostración. En esta parte se deben diseñar las interfaces de usuario del sistema.

**Análisis.-** Se considerarán las actividades análisis de la arquitectura, analizar un casos de uso, analizar una clase y analizar un paquete, que fueron analizadas en la fase de elaboración. Podemos decir que en la fase de elaboración se han analizado en promedio del 40% por ciento de los casos de uso, y el 80% por ciento necesario para realizar el análisis económico. Los porcentajes indicados son aproximaciones, que pueden variar según las características del proyecto, es decir en tamaño o complejidad.

A continuación indicaremos los análisis a realizar:

- **Analizar la arquitectura,** se harán algunas actualizaciones necesarias debidas a cambios que hubieran afectado a la arquitectura.
- **Analizar un caso de uso,** en las iteraciones que se realicen es probable que se adicionen algunos casos de uso, debiendo ser actualizado el modelo de casos de uso.
- **Analizar una clase,** se continua el trabajo que se inicio en la fase de elaboración.

- **Analizar un paquete,** se refinan los paquetes identificados en la fase de elaboración.

**Diseño.-** Corresponde en esta fase de construcción de concluir con el 90% por ciento de los casos de uso con el diseño y la implementación. No se debe añadir subsistemas, se podrán adicionar algunos subsistemas siempre que sea una alternativa adicional, es decir subsistemas similares que no necesiten nuevas interfaces. Se puede decir que el diseño y la implementación son los principales objetivos de la fase de construcción, como resultado es el modelo de diseño y el modelo de despliegue.

**Flujo de trabajo de implementación.-** Es la actividad que se desarrolla en mayor cantidad en la fase de construcción, se construyen los componentes con código, realizando las iteraciones necesarias. Al finalizar la fase de construcción todos los componentes serán completados con código.

Se indican las implementaciones necesarias para este flujo:

- **Implementación de un arquitectura.-** Si es necesario se actualizara la arquitectura.
- **Implementar una clase e implementar un subsistema.-** Se implementaran las clases y subsistemas del modelo de implementación.
- **Realizar pruebas de unidad.-** Por cada componente se realizan las pruebas de unidad.
- **Integrar el sistema.-** Se crea un plan de integración de construcciones, que indique la secuencia de las construcciones. Luego se integraran los componentes según el orden indicado en el plan de integración.

**Prueba.-** Es una actividad fundamental en la fase de construcción, se realizan las pruebas de unidad, se obtendrán los incrementos después de las iteraciones, hasta que se obtenga la versión completa del sistema.

Se consideran las siguientes pruebas:

- **Planificar las pruebas.-** Se seleccionaran los objetivos que comprueben las sucesivas construcciones y el sistema.
- **Diseñar las pruebas.-** Se determina como probar los requisitos en el grupo de construcciones con la finalidad de verificar los requisitos. Se preparan casos y procedimientos de prueba con este propósito.
- **Realizar pruebas de integración.-** Se ejecutaran los casos de prueba, siguiendo los procedimientos de prueba, añadirá construcciones al sistema según vayan estando disponibles.
- **Realizar pruebas del sistema.-** Se ejecutaran los casos de prueba de sistema, siguiendo los procedimientos de prueba del sistema. En la última iteración de la fase de construcción se comprobará la versión operativa inicial.
- **Evaluar las pruebas.-** Se revisan los resultados de las pruebas al final de cada construcción según los objetivos indicados en el plan de pruebas.

**Resumen de productos o entregables de la fase de construcción:**

- El plan de proyecto para la fase de transición.
- El sistema software ejecutable.

- Todos los artefactos, incluyendo los modelos del sistema.
- La descripción de la arquitectura actualizada.
- La versión preliminar del manual del usuario.
- El análisis de negocio: costo beneficio.

**2.4.4.2.4 Fase de transición.-** El objetivo principal es cumplir con los requisitos del usuario, entre otros objetivos es corregir errores y preparar los sitios y adaptar el software para que funcione en los sitios del usuario. Los manuales son completados. También es importante tratar de descubrir cualquier riesgo no identificados en las fases anteriores. El producto en esta fase se convierte en una versión beta, la cual es revisada por un pequeño grupo de usuarios con mucha experiencia, realizan las pruebas, para posteriormente emitir su informe donde indiquen las deficiencias y defectos. Luego los desarrolladores corrigen los problemas, para dirigir la versión general a la comunidad de usuarios.

Las actividades que se realizan en paralelo son cuatro: los flujos de trabajo.

#### **i. Flujos de trabajo.-**

**Requisitos.-** No se aplica en la fase de transición.

**Análisis.-** No se aplica en la fase de transición.

**Diseño.-** Las actividades del diseño es mínima, en la fase de transición, en el caso de presentarse algún problema de última hora se tendrá que modificar el diseño o rediseñar.

**Implementación.-** Preparación de la versión beta( o de pruebas de aceptación) a partir de la versión con capacidad operativa inicial producida durante la fase de transición. Instalar la versión en los lugares indicados de prueba.

**Prueba.-** Se realizan dos tipos de pruebas: beta y aceptación. Para las pruebas de beta, el personal de transición no estará presente, por lo que a los usuarios y al cliente se les dará las indicaciones necesarias para que pueda instalar el software de prueba. En las pruebas de aceptación, los miembros del personal del proyecto generalmente estarán presentes, de surgir algún error en el software, de ser posible deberá ser corregido en ese momento.

**Resumen de productos o entregables de la fase de transición:**

- El software ejecutable.
- Los modelos del sistema.
- Descripción completa y actualizada de la arquitectura.
- Manual del usuario y manual del sistema

**2.4.4.3 Comparación crítica del método análisis y diseño estructurado con el método orientado a objetos con la metodología RUP.-**

- i. En el método de análisis y diseño estructurado, las actividades que se desarrollan son: requerimientos, análisis, diseño, implementación y prueba. Estas actividades se realizan secuencialmente, es decir se termina con una actividad y se continúa con la siguiente. No está planeado revisar o regresar a las actividades anteriores, lo que origina el riesgo de no incluir requerimientos detectados en fases posteriores. Esto se debe a que en la práctica sucede que ciertos requerimientos son detectados como resultado de la ejecución de fases posteriores. En cambio, en la metodología orientado a objetos RUP, el proyecto es dividido en las fases de inicio, elaboración, construcción y transición, y dentro de cada una de ellas (con diferente intensidad) se realizan las actividades de requerimientos, análisis, diseño, implementación y prueba. En cada fase se pueden completar omisiones y corregir errores cometidos en la fase an-

terior. De esta manera, la metodología RUP ofrece una mayor garantía de que el software producido satisfaga todos los requerimientos del usuario.

- ii. En general, los métodos de análisis y diseño orientado a objetos, inclusive la metodología RUP, no incluyen un Modelo Entidad Relación como artefacto o producto. En cambio, para la metodología estructurada el Modelo Entidad Relación es un producto importante en la parte de diseño. Se supone que el diagrama de clases del análisis y diseño orientado a objetos estaría reemplazando al Modelo Entidad Relación; sin embargo, debido a que actualmente la mayoría de los manejadores de base de datos son relacionales y no orientados a objetos, el Modelo Entidad Relación resulta conveniente, ya que pasar de este modelo al modelo relacional (normalizado) del manejador es una tarea sencilla. Esto se confirma por el hecho de que en la actualidad los analistas y diseñadores de sistemas que aplican el método orientado a objetos (metodología RUP) lo incluyen como parte de los productos o entregables en el desarrollo de software.
- iii. Las metodologías orientadas a objetos, a pesar de usar el UML como instrumento de expresión en los artefactos o productos, no tienen un estándar en las notaciones utilizadas. Por ejemplo, “hay más de diez notaciones diferentes de los elementos de análisis y diseño orientados a objetos; ello dificulta una comunicación eficaz y uniforme”. (Larman, 1999).
- vi. En la actualidad, en nuestro país hay organizaciones que todavía usan el método estructurado y organizaciones que usan la metodología RUP. Se ha observado también que ambos métodos son utilizados como una

mezcla en el análisis y diseño orientado a objetos: "Este es bastante antiguo y sencillo, y se ha constituido en el método clásico de análisis que muchas instituciones siguen usando, ya sea en su última versión o combinándolo con los métodos orientados a objetos", (Carranza, 2006, p. 13).

#### **2.4.4.4 Evaluación crítica y mejoras de la Metodología RUP.-**

Como parte de la investigación se ha realizado una evaluación crítica a la metodología RUP, y como resultado de ello se presentan sugerencias de modificaciones que facilitan su aplicación en el desarrollo de sistemas de información medianos y pequeños.

- a) La metodología RUP es aplicable a instituciones (organizaciones) grandes y complejas, porque permite dividir en partes o procesos, identificando casos de uso y actores, presentando un modelo de negocios inicial, para luego llegar a la realización de los casos de uso. Se requiere de un equipo profesional capacitado para las diferentes actividades, lo que implica un costo alto en recursos humanos.

**En el caso de las pequeñas y medianas empresas PYMES, la metodología RUP resulta de difícil aplicación, pero es posible hacer variaciones y simplificaciones, y con respecto a la cantidad de iteraciones en las cuatro fases (inicio, elaboración, construcción y transición), se sugiere realizar una iteración en la primera fase, y dos iteraciones en la tres fases siguientes. En el caso de empresas pequeñas, debido a que sus procesos no son complejos, sería suficiente una iteración, pesar de que ciertos autores recomiendan como mínimo dos.**

b) La mayoría de los autores toma como referencia al autor (Booch, 1999). Sin embargo, la metodología no está totalmente estandarizada, pues diferentes autores dan definiciones similares pero no idénticas; además, los artefactos que se deben entregar en cada fase son distintos de acuerdo a diferentes autores.

**Por esta razón, en el marco teórico, en las secciones 2.4.4.2.1, 2.4.4.2.2, 2.4.4.2.3 y 2.4.4.2.4 de las cuatro fases (inicio, elaboración, construcción y transición), se presentan los artefactos o productos que en opinión de la autora de la presente tesis son los más utilizados y fáciles de entender para los proyectos de software; esta opinión también es compartida en los diversos textos investigados.**

c) En la Metodología RUP supuestamente los artefactos son representados utilizando el UML. Sin embargo, durante la investigación realizada se ha observado que, en su mayoría, los textos o libros investigados, los representan usando diversas notaciones. Por ejemplo, algunos textos usan estereotipos (una extensión de UML), mientras que otros utilizan la notación tradicional.

**En opinión de la autora de la tesis, en todo proyecto de desarrollo de software se debería usar como notación el UML actualizado, con lo que se evitaría confusiones.**

**Existen diversas herramientas para desarrollar software con la metodología RUP, entre ellas Visio, Artist, etc. Sin embargo, la herramienta Rational Rose es la que se acerca más a la notación estándar, ofrece más ventajas.**

d) En la metodología RUP se realiza el análisis costo beneficio en las tres primeras fases, como lo indica la mayoría de los autores, principalmente

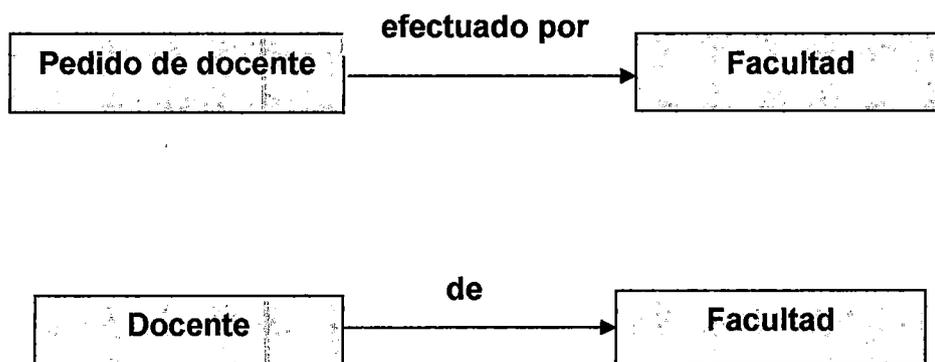
(Booch, 1999). Este análisis costo beneficio se justifica porque a medida que se avanza con el proyecto hay cambios, aparecen más requisitos representados como casos de uso, lo que hace que el sistema incremente su costo; también las iteraciones en cada fase implican un incremento en el costo de recursos humanos. Sin embargo, la metodología RUP no considera un estudio de factibilidad antes de iniciar el proyecto, estudio que si es considerado en la metodología tradicional.

**De acuerdo a la experiencia de la autora de la tesis resultaría conveniente que antes de iniciar un proyecto de software con la metodología RUP se realice un estudio de factibilidad para ser presentado al cliente. ( Schach, 2005, Larman, 1999 y Arlow, 2006) .**

- e) **Propuesta de mejora para los diagramas de clase.-** Para hacer que los diagramas de clases sean más descriptivos y más fáciles de entender se propone la siguiente convención para denotar las asociaciones entre clases:
- i. El nombre de la asociación debe ser una frase calificativa ó una preposición en vez de un verbo. Por ejemplo, en la Figura II. la asociación entre la clase **Facultad** y la clase **Pedido de docente** debe ser nombrada con **“efectuado por”** en vez del verbo **“realiza”**; la asociación entre la clase **Facultad** y la clase **Docente** debe ser nombrada con la preposición **“de”** en vez del verbo **“tiene”** o del verbo **“pertenece”**. Entonces, la primera asociación se leerá como **“pedido efectuado por la facultad”**; la segunda asociación se leerá como **“docente de la facultad”**, con lo que el diagrama resulta más descriptivo y claro.
  - ii. Se debe indicar la dirección de la asociación, de la clase que es origen de la asociación hacia la clase que el final de la asociación. En el

primer ejemplo, la dirección es de **Pedido de docente** hacia **Facultad**; en el segundo ejemplo la dirección es de **Docente** hacia **Facultad**. La dirección se indicará con una flechita que apunta hacia el final de la asociación.

Con estas sencillas recomendaciones los diagramas de clases y los entidad relación ganan poder descriptivo y claridad.



**Figura II.7 Mejora del diagrama de clase**

## CAPÍTULO III

### **ANÁLISIS Y DISEÑO DE UN SISTEMA DE GESTIÓN DE INFORMACIÓN ACADÉMICA DE UNA UNIVERSIDAD UTILIZANDO LA METODOLOGÍA RUP.**

En esta parte del trabajo se aplica la metodología RUP descrita en el capítulo 2, para el análisis y diseño de un sistema para la gestión académica de una universidad cuya estructura organizacional esté de acuerdo a lo establecido por la Ley Universitaria. En la formulación del problema se indicó que actualmente en nuestro país un apreciable número de universidades de 52% no cuentan con un sistema que les permita automatizar las actividades académicas. La Universidad Nacional Mayor de San Marcos se ciñe a la estructura dada por la Ley Universitaria, por lo que para el desarrollo de la parte aplicativa de la tesis se ha tomado a esta universidad como referencia o modelo.

El análisis y diseño del sistema que se efectúa en la presente tesis es lo que en la metodología RUP corresponde a las fases de **inicio** y **elaboración**. Las fases **construcción** y **transición** de RUP están fuera de los objetivos y alcances de la tesis. Sin embargo, se presenta un prototipo que muestra las principales funciones.

### **3.1 Fase de inicio.-**

**3.1.1 Plan de proyecto.-** Presentaremos el plan de proyecto, de acuerdo a lo descrito en las secciones anteriores. Se debe indicar que los tiempos requeridos y las iteraciones son aproximaciones que pueden variar durante el desarrollo del proyecto. Se debe señalar que los tiempos y las iteraciones indicadas es lo necesario, ajustando los tiempos, a fin de minimizar los costos de recurso humanos. No se han considerado las fechas de inicio y fechas de término en las fases, porque el trabajo realizado de investigación, ha sido desarrollado según la disponibilidad de tiempo y recursos económicos propios del investigador. Como se observará el trabajo será desarrollado hasta la fase de elaboración, un aproximado de seis meses.

<b>Fases</b>	<b>Tiempo requerido</b>	<b>Numero de iteraciones</b>
Fase de inicio	2 meses	2
Fase de elaboración	4 meses	3
Fase de construcción	5 meses	3 o 4
Fase de transición	1 mes	2
<b>Tiempo requerido</b>	<b>12 meses</b>	

**Figura III.1 Plan de Proyecto**

**3.1.2 Flujo de trabajo de requisitos.-** A continuación pasaremos a desarrollar las actividades necesarias para el proyecto.

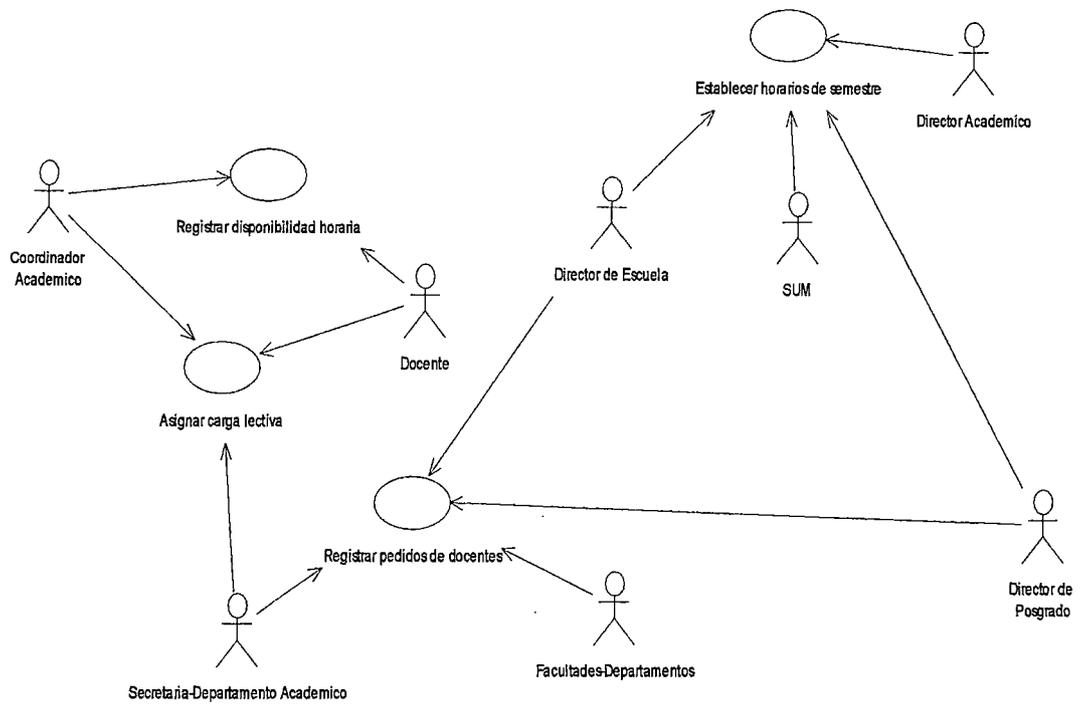
Las técnicas de la metodología RUP, nos ha permitido obtener y conocer los requisitos del los usuarios que a continuación pasaremos a describir.

### **3.1.2.1 Relación de requisitos candidatos.-**

1. Almacenamiento de datos e información de los docentes de las Facultades como los datos personales, código docente, categoría, clase, etc.
2. Almacenamiento y actualización de las actividades académicas de los docentes como tutorías, consejería, coordinaciones, etc., conocida como la Ficha de Actividades.
3. Almacenamiento de información de los docentes durante cada semestre, como asistencia a los cursos, participación en Talleres, Seminarios, congresos, estudios de capacitación, se conoce como el Historial del Docente.
4. Almacenamiento y actualización de la disponibilidad de los docentes de sus horarios y preferencias de curso por cada Semestre Académico.
5. Elaboración de cargas lectivas de manera automatizada.
6. Almacenamiento y actualización de los planes de estudio de las Escuelas Académicas Profesionales.
7. Almacenamiento y actualización de los planes de estudio post grado.
8. Registro de pedidos o solicitudes de requerimiento de docentes.
9. Elaboración y emisión del Cuadro de Horarios del Semestre Académico.
10. Emisión de cargas lectivas aprobadas por el Consejo de Facultad.
11. Generación de reportes de las actividades académicas de los docentes.
12. Emisión de reportes del Historial de los Docentes.

### **3.1.2.2 Requisitos funcionales representados en el Modelo de Negocios Inicial.-**

En base a los requisitos candidatos, presentaremos los requisitos funcionales utilizando casos de uso y actores representados en el modelo de negocios inicial.



**Figura III.2 Modelo de Negocios Inicial**

A continuación anotaremos las descripciones de los casos de uso, por lo tanto se añaden las descripciones de los casos de uso iniciales:

## **Caso de uso: Establecer horario de semestre**

### **1. Caso de uso: Establecer horario de semestre**

**Breve descripción:** Al inicio de un semestre académico se definen los cursos que se van a dictar, con su respectivo horario, que incluye hora y aula.

**Descripción:**

No se aplica en esta etapa inicial.

### **2. Caso de uso: Registrar disponibilidad horaria**

**Breve descripción:** Se llena la Hoja de Prioridades de Docente para el semestre, que contiene su disponibilidad y preferencias de curso.

**Descripción:**

No se aplica en esta etapa inicial.

### **3. Casos de uso: Registrar pedidos de docentes**

**Breve descripción:** Se registra un pedido de docentes de las Escuelas de la Facultad y de otras Facultades.

**Descripción:**

No se aplica en esta etapa inicial.

### **4. Casos de uso: Asignar carga lectiva**

**Breve descripción:** El Coordinador del Departamento Académico asigna al docente los cursos que dictaran en el Semestre Académico.

**Descripción:**

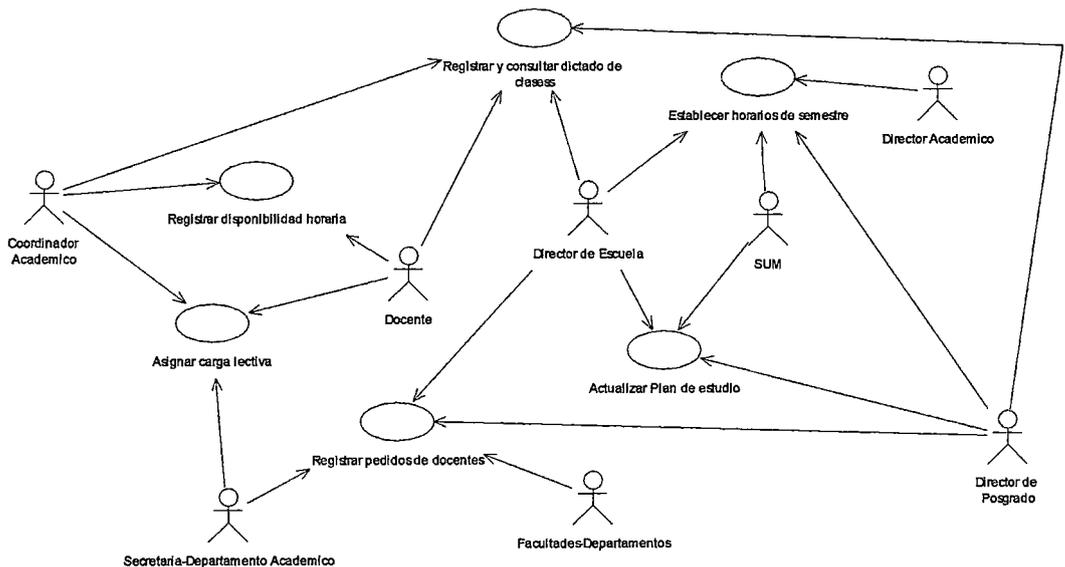
No se aplica en esta etapa inicial.

### 3.1.2.3 Relación de requisitos no funcionales.-

El diseño del sistema es independiente de la plataforma de la institución, con el objetivo de que cualquier universidad que tenga la estructura según la Ley Universitaria lo pueda implementar. A continuación se presentan los requisitos no funcionales básicos para esta fase.

- El sistema de información se implementara en la plataforma Cliente/Servidor.
- El tiempo de respuesta será el promedio de una intranet.
- Las interfaces de usuario deberán ser amigables y fáciles de aprender.

**3.1.3 Flujo de trabajo de análisis.-** En esta parte se refinaron los requisitos funcionales, y se obtuvo dos casos de uso adicionales: el caso de uso “Registrar y consultar el dictado de cursos” y el caso de uso “Actualizar plan de estudio”, tal como se muestra en el modelo de negocios mejorado.



**Figura III.3 Modelo de Negocios Inicial**

A continuación se realiza una descripción detallada de los casos de uso más importantes o por prioridad del modelo de negocios:

### **1. Caso de uso: Establecer horario de semestre**

**Breve descripción:** Al inicio de un semestre académico se define los cursos que se van a dictar, con su respectivo horario, que incluye hora y aula.

**Descripción:**

En proceso de desarrollo.

### **2. Caso de uso: Registrar disponibilidad horaria**

**Breve descripción:** Se llena la Hoja de Prioridades de Docente para el semestre, que contiene su disponibilidad y preferencias de curso.

**Descripción:**

1. El sistema emite formato para ser llenado por el docente con su disponibilidad horaria (Hoja de prioridades).
2. El docente ingresa su disponibilidad horaria dentro del plazo permitido.
  - 2.1 Nombre de curso.
  - 2.2 Facultad y Escuela Académica donde desea dictar.
  - 2.3 Horario (mañana, tarde, noche).
3. El sistema actualiza datos del formato del docente, si hubiera ingresado de manera incorrecta.
  - 1.1 Adición de cursos.
  - 1.2 Modificación de cursos.
  - 1.3 Eliminación de cursos.
2. El sistema emite reporte del formato con los datos ingresados.

### 3. Casos de uso: Registrar pedidos de docentes

**Breve descripción** Se registra un pedido de docente de las Escuela Académica de la Facultad o de otras Facultades.

#### **Descripción:**

1. El sistema registra el pedido de la Dirección de Escuela o de la Dirección de Postgrado de la misma Facultad, y de otros Departamentos Académicos de la Facultad.
2. El sistema modifica datos en caso de error de digitación o algún cambio imprevisto.
3. El sistema emite reportes de los pedidos:
  - 3.1 De la misma Facultad por Direcciones de Escuelas.
  - 3.2 De otras Facultades a través de los Departamentos Académicos.

### 4. Casos de uso: Asignar carga lectiva

**Breve descripción:** El Coordinador del Departamento Académico asigna al docente los cursos que dictarán en el semestre académico.

#### **Descripción:**

1. El sistema crea las cargas lectivas en base a la disponibilidad horaria del docente y el horario del curso.
2. El sistema consistencia o valida cada curso designado por el Coordinador Académico en la carga lectiva.
3. El sistema consistencia el turno del uso con la disponibilidad horaria del docente.
4. El sistema contabiliza la cantidad de horas mientras se va seleccionando el curso, sumando la cantidad de horas total del docente, hasta completar las horas que le corresponde según su carga lectiva, sea tiempo parcial, tiempo completo, dedicación exclusiva entre otros.

5. El sistema emite reporte de la carga lectiva del docente.

#### **5. Caso de uso: Actualizar plan de estudios**

**Breve descripción:** Se actualiza el plan de estudios de las Escuelas Académicas Profesionales.

**Descripción:**

En proceso de desarrollo.

#### **6. Caso de uso: Registrar y consultar dictado de clases.**

**Breve descripción:** Se lleva un control del dictado de cursos a través de supervisión del curso, encuestas durante el semestre, entre otros.

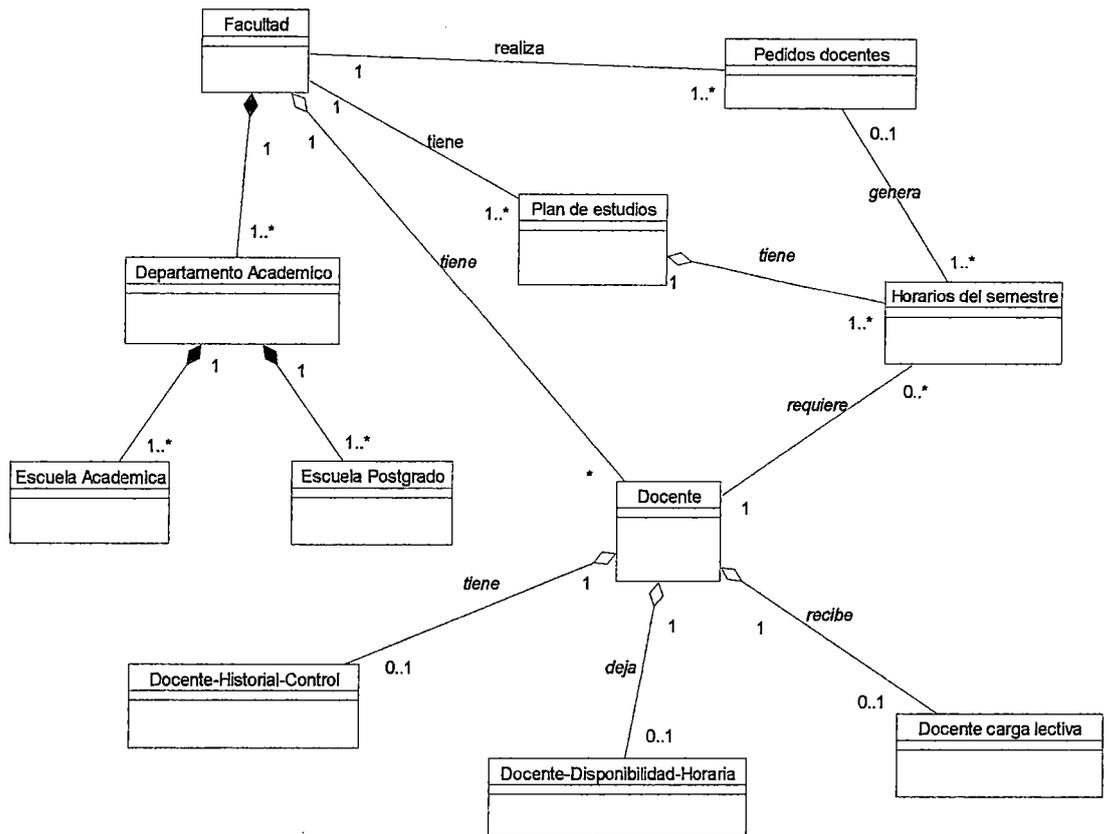
**Descripción:**

En proceso de desarrollo.

**3.1.4 Flujo de trabajo de diseño.-** En esta actividad corresponde presentar un diagrama de clases inicial y un prototipo que muestre la idea del sistema de información.

**3.1.4.1 Diagrama de clases.-** Se presenta un diagrama de clases inicial, que comprende once clases, como se muestra en la Figura III. Se presenta cada clase con su respectivo nombre, no es necesario presentar los nombres de los atributos de las clases, en caso conocer los atributos de algunas clases, se pueden ir colocando respectivamente. Las once clases se mencionan a continuación:

- i. Clase Facultad
- ii. Clase Departamento Académico
- iii. Clase Escuela Académica
- iv. Clase Escuela Postgrado
- v. Clase Plan de estudios
- vi. Clase Pedidos docentes
- vii. Clase horarios del semestre
- viii. Clase docente
- ix. Clase Docente-Historial-Control
- x. Clase Docente-Disponibilidad-Horaria
- xi. Clase Docente carga lectiva



**Figura III.4 Diagrama de clases**

### **3.1.5 Relación inicial de riesgos, y descripción de cómo mitigarlos.-**

#### **3.1.5.1 Relación inicial de riesgos.-**

- i. Los requerimientos no han sido bien definidos.
- ii. El tiempo estimado para culminar con el proyecto sea insuficiente.
- iii. Insatisfacción de los usuarios con algunas interfaces.
- iv. Los programas no se ejecuten correctamente.
- v. Los usuarios no consideraron algunos requisitos.

#### **3.1.5.2 Plan de contingencia.-**

- i. Replantear nuevamente los requerimientos con los usuarios y el equipo de trabajo encargado de los requisitos.
- ii. Ampliar el tiempo con horas o días adicionales para culminar con el proyecto.
- iii. Modificar las interfaces con la ayuda de un prototipo de interfaces.
- iv. Adicionar más horas para realizar las correcciones con el grupo de analistas y programadores más experimentados.
- v. Tratar en lo posible que los requisitos adicionales sean considerados en los casos de uso existentes; de lo contrario, adicionar los casos de uso necesarios.

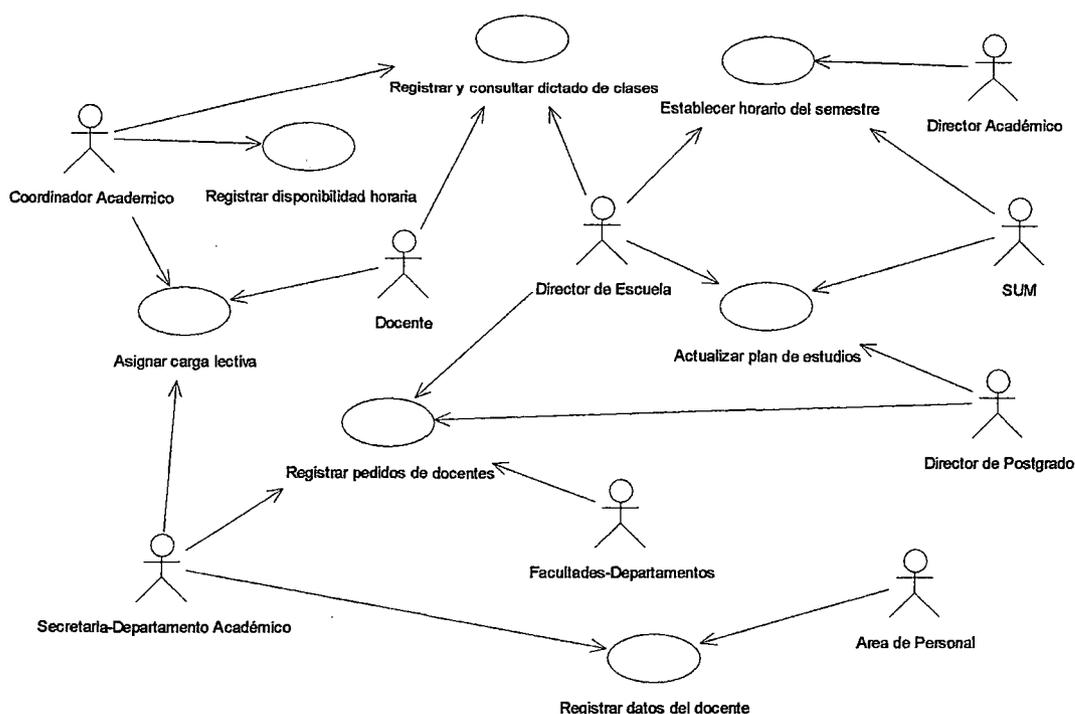
## **3.2 Fase de elaboración.-**

En el capítulo II, sección 2.4.4.2.2 se explicó la fase de elaboración, que consistía en cuatro actividades, que a continuación pasaremos a desarrollar.

### **3.2.1 Flujo de trabajo de requisitos.-**

**3.2.1.1 9Requisitos iniciales.-** Con la finalidad de completar los requisitos de los usuarios, revisamos el Modelo de Negocios diseñado en la fase de inicio. Analizamos e iterando más de una vez, se observa que se requiere adicionar el

caso de uso "Registrar datos del docente", con el que finalmente obtenemos el Modelo casos del Sistema de Información como se muestra en la Figura III.5 .



**Figura II.5 Modelo de Caso de usos del Sistema de Información**

**1. Caso de uso: Establecer horario de semestre**

<p><b>Breve descripción:</b> Al inicio de un semestre académico se define los cursos que se van a dictar, con su respectivo horario, que incluye hora y aula.</p>
<p><b>Descripción:</b></p> <ol style="list-style-type: none"> <li>1. El sistema mantiene actualizado el registro de horarios en cada semestre académico por Escuelas Académicas Profesionales:</li> </ol>

- 1.1 Ingreso de cursos, horarios y aulas.
- 1.2 Modificación de cursos, horarios y aulas.
- 1.3 Eliminación cursos, horarios y aulas.
- 2. El sistema emite reportes por Escuelas Académicas Profesionales de los horarios para el semestre académico.
- 3. El sistema interactúa con el SUM, a fin de mantener actualizada la base de datos para la matrícula.

**2. Caso de uso: Registrar disponibilidad horaria**

<p><b>Breve descripción:</b> Se llena la Hoja de Prioridades de Docente para el semestre, que contiene su disponibilidad y preferencias de curso.</p>
<p><b>Descripción:</b></p> <ul style="list-style-type: none"><li>1. El sistema emite formato para ser llenado por el docente con su disponibilidad horaria (Hoja de prioridades).</li><li>2. El docente ingresa su disponibilidad horaria dentro del plazo permitido.<ul style="list-style-type: none"><li>2.1 Nombre de curso.</li><li>2.2 Facultad y Escuela Académica donde desea dictar.</li><li>2.3 Horario hora de inicio y hora de fin.</li></ul></li><li>3. El sistema actualiza datos del formato del docente, si hubiera ingresado de manera incorrecta.<ul style="list-style-type: none"><li>3.1 Adición de cursos.</li><li>3.2 Modificación de cursos.</li><li>3.3 Eliminación de cursos.</li></ul></li><li>4. El sistema emite reporte del formato con los datos ingresados.</li></ul>

### 3. Casos de uso: Registrar pedidos de docentes

**Breve descripción:** Se registra un pedido de docente de las Escuela Académica de la Facultad o de otras Facultades.

**Descripción:**

1. El sistema registra el pedido de la Dirección de Escuela o de la Dirección de Postgrado de la misma Facultad, y de otros Departamentos Académicos de la Facultad.
2. El sistema modifica datos en caso de error de digitación o algún cambio imprevisto.
3. El sistema emite reportes de los pedidos:
  - 3.1 De la misma Facultad por Direcciones de Escuelas.
  - 3.2 De otras Facultades a través de los Departamentos Académicos.

### 4. Casos de uso: Asignar carga lectiva

**Breve descripción:** El Coordinador del Departamento Académico asigna al docente los cursos que dictarán en el semestre académico.

**Descripción:**

1. El sistema crea las cargas lectivas en base a la disponibilidad horaria del docente y el horario del curso.
2. El sistema consistencia o valida cada curso designado por el Coordinador Académico en la carga lectiva.
3. El sistema consistencia el turno del uso con la disponibilidad horaria del docente.
4. El sistema contabiliza la cantidad de horas mientras se va seleccionando el curso, sumando la cantidad de horas total del docente, hasta completar las horas que le corresponde según su carga lectiva, sea tiempo parcial, tiempo completo, dedicación exclusiva entre otros.

5. El sistema emite reporte de la carga lectiva del docente.

### 5. Caso de uso: Actualizar plan de estudios

**Breve descripción:** Se actualiza el plan de estudios de las Escuelas Académicas Profesionales.

**Descripción:**

1. El sistema ingresa los planes de estudios para Escuela.
  - 1.1 Código de Escuela Académica Profesional.
  - 1.2 Código del curso.
  - 1.3 Nombre del curso.
  - 1.4 Cantidad de créditos.
2. El sistema actualiza los planes de estudio por Escuela.
  - 2.1 Adición cursos.
  - 2.2 Modificación cursos.
  - 2.3 Eliminación de cursos.
3. El sistema emite Reportes por Escuelas Académicas Profesionales.
4. El sistema interactúa con el SUM para informar de las actualizaciones realizadas, a fin de que mantengan actualizadas su base de datos para la pre matricula y matrícula del semestre académico.

## **6. Caso de uso: Registrar y consultar dictado de clases.**

**Breve descripción:** Se lleva un control del dictado de cursos a través de supervisión del curso, encuestas durante el semestre académico, entre otros.

### **Descripción:**

1. El sistema permite ingresar:
  - 1.1 El puntaje obtenido en la encuesta en el semestre académico.
  - 1.2 La asistencia y puntualidad en el dictado de clases.
  - 1.3 Información referente al Historial del docente:
    - 1.3.1 Cantidad de horas no lectivas por semestre.
    - 1.3.2 Jurado de tesis y asesoría de tesis.
    - 1.3.3 Participación en Talleres, Seminarios, Conferencias, entre otros.

## **7. Caso de uso: Registrar y actualizar datos del docente.**

**Breve descripción:** Se registra y actualiza la base de datos del sistema con el Sistema de Personal de la Universidad.

### **Descripción:**

1. El sistema mantendrá los datos actualizados del docente con enlace del Sistema de Personal por Facultades.
  - 1.1 Código docente.
  - 1.2 Nombre docente.
  - 1.3 Dni.
  - 1.4 Dirección.
  - 1.5 Teléfono.
  - 1.6 Dirección electrónica.
  - 1.7 Categoría.
  - 1.8 Clase.
  - 1.9 Tiempo de servicios.
  - 1.10 Grados obtenidos.

#### 1.11 Títulos obtenidos.

2. Cada Departamento Académico de la Facultad podrá acceder a la base de datos del Sistema para actualizar información de:

2.1 Adición de datos.

2.2 Modificación de datos.

2.3 Eliminación de datos.

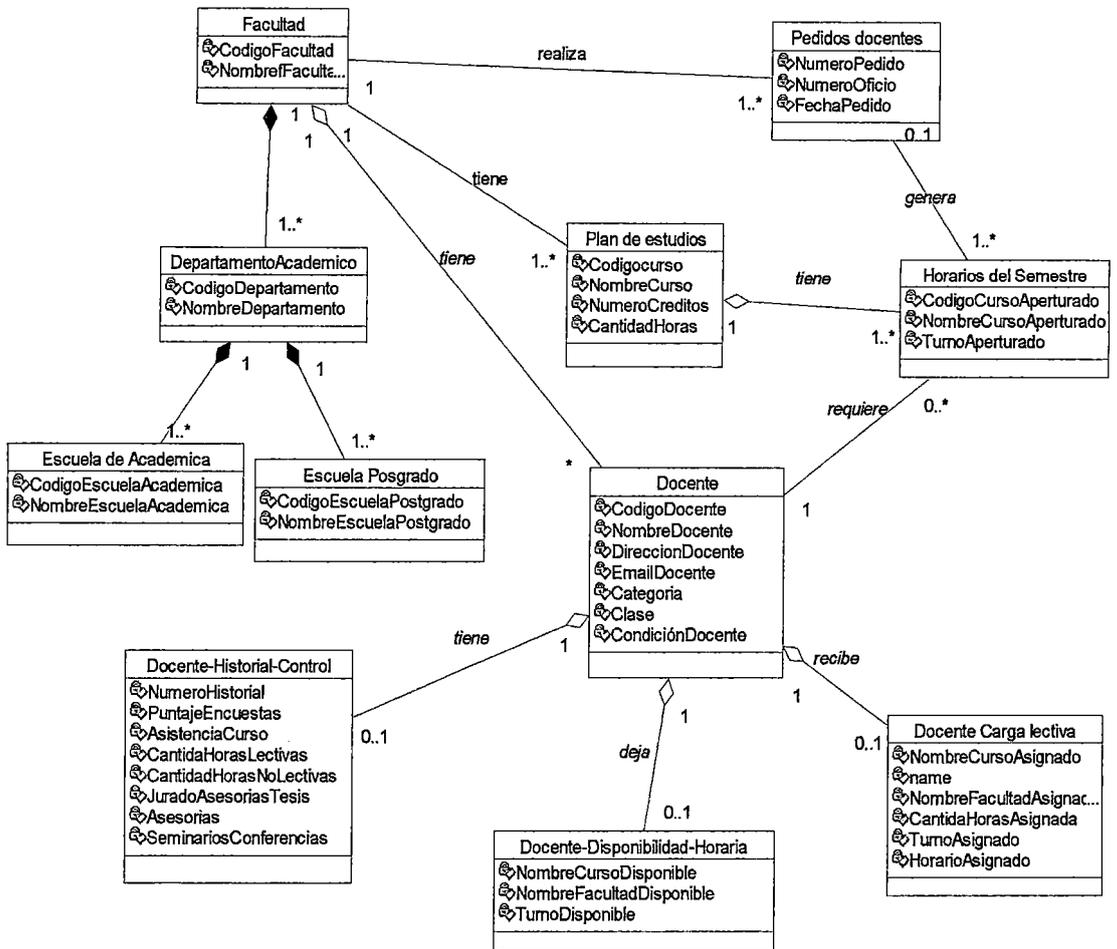
3. El Sistema informará al Sistema de Personal de la Universidad de algún cambio o modificación que hubiere realizado, así como también Personal hará lo mismo.

4. El sistema emitirá reporte de los docentes nombrados por categoría, clase.

5. El sistema emitirá reportes de los docentes contratados por categoría, clase.

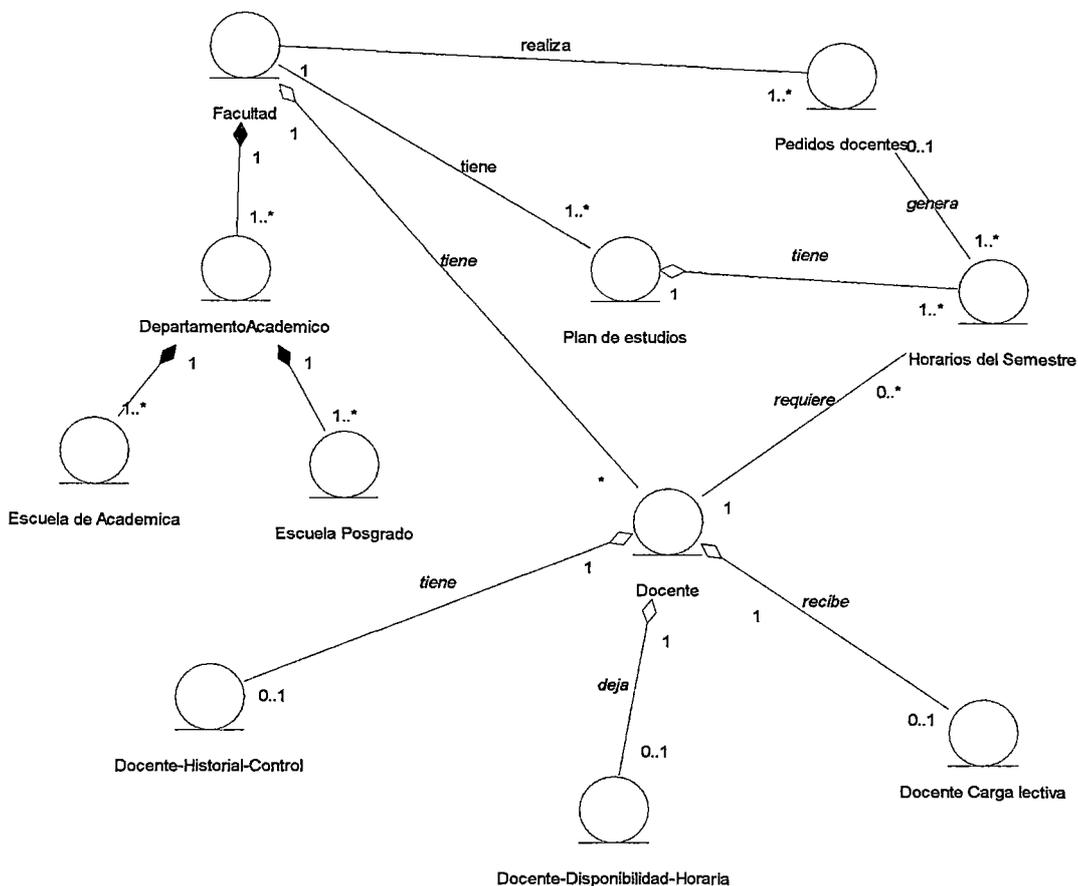
### **3.2.2 Flujo de trabajo de análisis.-**

**3.2.2.1 Diagrama de clases.-** Corresponde en esta parte mejorar el diagrama de clases, conocidas también como clase entidad, con el modelado de clases, revisaremos las relaciones entre clases y adicionaremos sus y atributos respectivamente. En la Figura III.6 se muestra el diagrama respectivo.



**Figura III.6 Diagrama de clases inicial con atributos y relaciones**

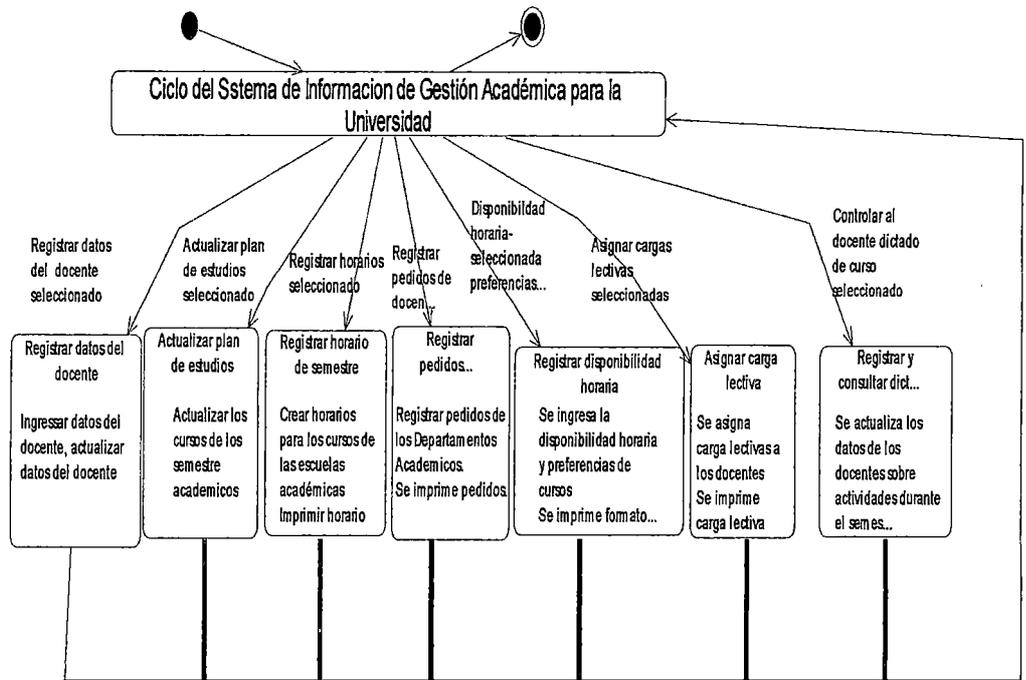
También se pueden representar las clases entidades como estereotipos, como se muestra en la Figura III.7.



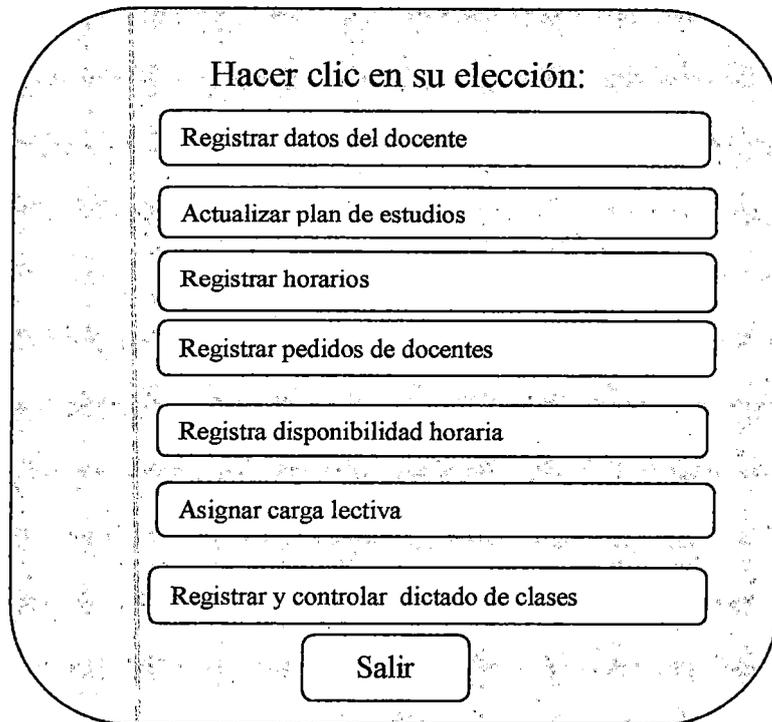
**Figura III.7 Diagrama de clases inicial que muestra los estereotipos.**

**3.2.2.2.2 Modelo dinámico inicial.-** El tercer paso es el modelado dinámico, se dibujara el diagrama de estado, mostrara las operaciones realizadas por el sistema de información, se tomara como fuente principal a los escenarios.

En la siguiente Figura III.8 se presenta el diagrama de estados inicial del Sistema de Información de Gestión para la Universidad.



**Figura III.8 Diagrama de estados inicial del Sistema de Gestión Académica para la Universidad**



**Figura III.9 Menú principal inicial en el Sistema de Gestión Académica De la Universidad**

**3.2.2.3 Extracción de clases borde.-** Las clases borde corresponden a las pantallas de entrada, salida e informes impresos, a continuación pasaremos a indicar las clases borde iniciales en el sistema de información por casos de uso:

**Clase interfaz de usuario**

Clase borde del caso de uso: Registrar datos del docente.

Inicialmente no presenta en esta etapa.

Clase borde del caso de uso: Actualizar plan de estudios.

Clase Reporte de plan de estudios por Escuela Académica y semestre académico.

Clase Reporte de plan de estudios por Escuela Académica de todos los semestres.

Clase Reporte de plan de estudios por Escuela de Posgrado y semestre académico.

Clase Reporte de plan de estudios por Escuela de Posgrado de todos los semestres.

Clase bordes del caso de uso: Registrar disponibilidad horaria.

Clase Informe de horarios por Escuela Académica.

Clase Informe de horarios por Escuela Académica y por semestre.

Clase Informe de horarios por Escuela de Postgrado.

Clase Informe de horarios por Escuela de Postgrado y por semestre.

Clase bordes del caso de uso: Registrar pedidos de docentes.

Clase informe de pedidos por Facultad de sus Departamentos Académicos.

Clase Informe de pedidos por Direcciones de Escuela Académica.

Clase Informe de pedidos por Direcciones de Escuela Posgrado.

Clase borde del caso de uso: Registrar disponibilidad horaria.

Clase Formato con prioridades del docente para el semestre académico.

Clase borde del caso de uso: Asignar carga lectiva.

Clase Reporte de carga lectiva del docente.

Clase borde del caso de uso: Registrar y controlar dictado de cursos.

Clase Historial del Docente durante el semestre académico.

#### **3.2.2.4 Extracción de clase control.-**

Clase control del caso de uso: Registrar datos del docente.

Inicialmente no presenta

Actualizar plan de estudios Actualizar plan de estudios.

Inicialmente no presenta.

Clase control del caso de uso: Actualizar plan de estudios.

Inicialmente no presenta.

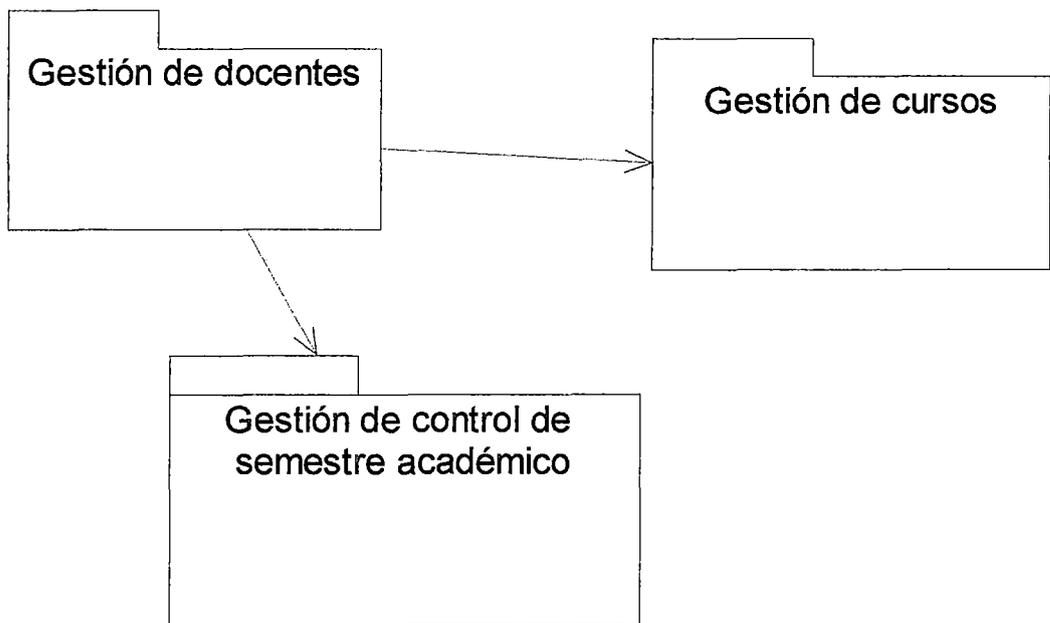
Clase control del caso de uso: Registrar disponibilidad horaria.

Inicialmente no presenta.

Clase control del caso de uso: Registrar disponibilidad horaria.

Inicialmente no presenta.

**3.2.2.5 Diagrama de paquetes.-** En el siguiente Figura III.10 se muestra el Diagrama de paquetes del análisis, con la finalidad de organizar el modelo del análisis en partes más pequeñas y que puedan ser más manejables.



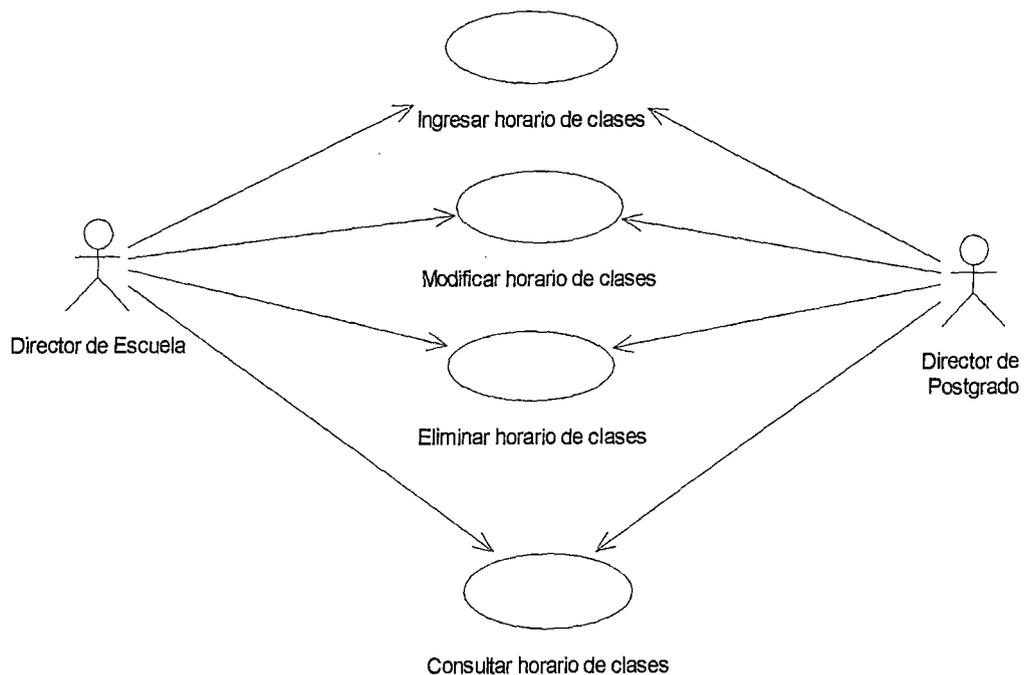
**Figura III.10 Diagrama de paquetes**

### 3.2.3 Flujo de trabajo de diseño.-

#### 3.2.3.1 Refinamiento de los casos de uso.-

En esta parte del análisis ya se ha obtenido las clase entidad, la clase borde, y todavía no se ha encontrado clase control. Ahora revisaremos nuevamente el diagrama de casos de uso para determinar si requiere de una siguiente iteración.

Al realizar el análisis se observa que hay modificaciones con respecto al Modelo de Caso de Usos del Sistema de Información, entonces se realizará una siguiente iteración.



**Figura III.11 Diagrama de casos de uso de horario de clases.**

<b>Breve descripción:</b> El sistema permite ingresar el horario de clases de un semestre académico de la Escuela Académica Profesional y de la Escuela de Posgrado.
<b>Actores :</b> Director de Escuela y Director de Posgrado.
<b>Precondición:</b> El horario del semestre académico ha sido definido por el Director de Escuela.
<b>Descripción :</b> <ol style="list-style-type: none"> <li>1. Director de Escuela y el Director Posgrado definen los cursos del semestre.</li> <li>2. Se ingresa el curso con el aula y hora en que se dicta.</li> <li>3. Se ingresa el nombre del Docente que dictara el curso. Es opcional.</li> <li>4. El sistema valida que no haya cruce de horario en la misma aula.</li> </ol>
<b>Postcondición :</b> El horario de clases se encuentra registrado en el sistema.
<b>Excepciones:</b> El curso fue eliminado del semestre académico.

**Tabla III. 1 Descripción del caso de uso Ingresar horario de clases**

<b>Breve descripción:</b> El sistema permite modificar el aula y hora de un curso del horario de clase de un semestre académico de la Escuela Académica Profesional y de la Escuela de Posgrado.
<b>Precondición:</b> El aula y hora de un curso se encuentra ingresado en el sistema.
<b>Actores:</b> Director de Escuela y Director de Posgrado.
<b>Descripción:</b> <ol style="list-style-type: none"> <li>1. El sistema muestra el curso ingresado con su turno del semestre académico.</li> <li>2. Se selecciona el curso a modificar el aula y/o hora en que se dicta.</li> <li>3. Se graba la modificación realizada.</li> </ol>

4. Si se desea realizar otra modificación se repite el paso 2 y el paso 3.
<b>Postcondición:</b> El aula y hora del curso ha sido modificado.
<b>Excepciones:</b> El curso fue eliminado del semestre académico.

**Tabla III. 2 Descripción del caso de uso Modificar horario de clases**

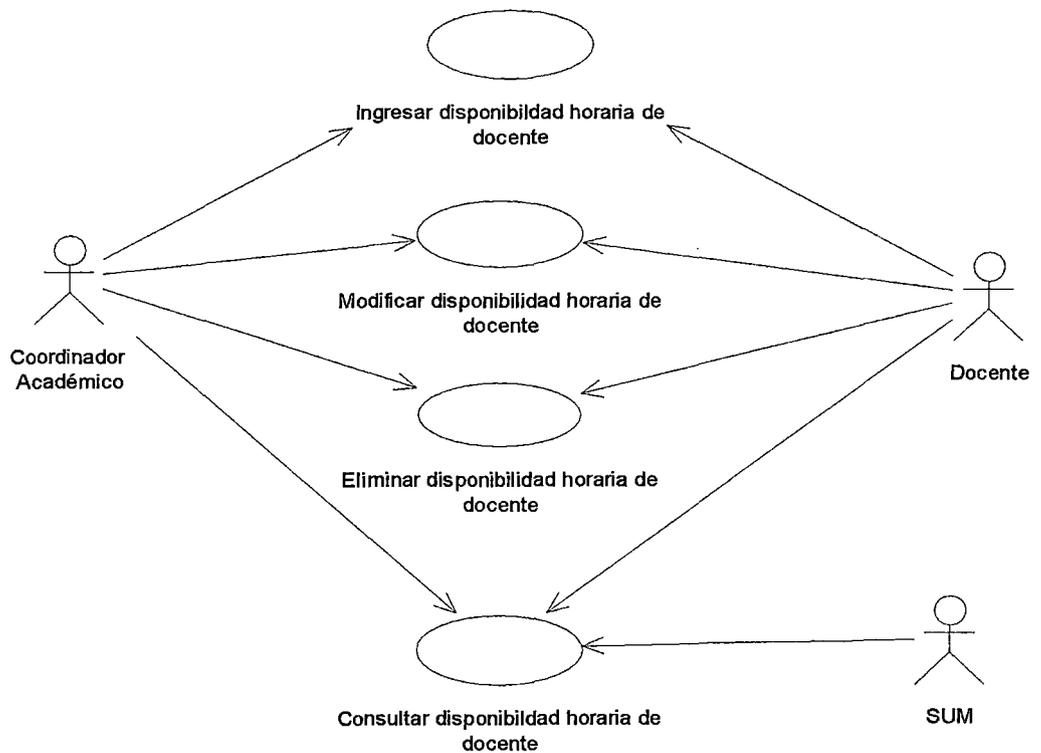
<b>Breve descripción:</b> El sistema permite eliminar un curso del horario de clases de un semestre académico de la Escuela Académica Profesional o de la Escuela de Posgrado.
<b>Actores:</b> Director de Escuela y Director de Posgrado
<b>Precondición:</b> El curso ha sido ingresado en el sistema.
<b>Descripción:</b> <ol style="list-style-type: none"> <li>1. Se selecciona el ciclo de la Escuela Académica.</li> <li>2. Se selecciona el curso a eliminar.</li> <li>3. Se graba la modificación realizada.</li> </ol>
<b>Postcondición:</b> El curso ha sido eliminado del semestre académico.
<b>Excepciones:</b>

**Tabla III.3 Descripción del caso de uso Eliminar horario de clases**

<b>Breve descripción:</b> El sistema permite visualizar el horario de clase ingresado.
<b>Actores:</b> Director de Escuela y Director de Posgrado.
<b>Precondición:</b> El curso ha sido ingresado en el semestre académico.
<b>Descripción:</b> <ol style="list-style-type: none"> <li>1. Se selecciona la escuela académica y el semestre académico.</li> <li>2. El sistema muestra los cursos con sus respectivas aulas y turnos designa-</li> </ol>

dos.
3. Si se desea realizar otra consulta, repetir el paso 1 y paso 2.
<b>Postcondición:</b>
<b>Exepciones:</b>

**Tabla III.4 Descripción del caso de uso Consultar horario de clases**



**Figura III.12 Diagrama de casos de uso de disponibilidad horaria de docente**

<b>Breve descripción:</b> El sistema permite ingresar datos de las prioridades del docente para el semestre académico, como la disponibilidad horaria y preferencias de cursos a dictar.
<b>Actores:</b> Coordinador Académico y Docente.
<b>Precondición:</b> Los cursos tienen aulas y horas en que se dictaran las clases.
<b>Descripción:</b> <ol style="list-style-type: none"> <li>1. El sistema permite ingresar los datos : <ol style="list-style-type: none"> <li>1.1 Los días y horas disponibles del docente.</li> <li>1.2 El nombre del curso de preferencia.</li> <li>1.3 La Escuela Académica Profesional a la que pertenece el curso.</li> <li>1.4 La Facultad a la que pertenece la Escuela Académica Profesional.</li> </ol> </li> <li>2. El sistema almacena la información ingresada.</li> <li>3. Si desea ingresar más cursos repetir paso 1 y paso 2.</li> </ol>
<b>Postcondición:</b> La disponibilidad horaria de docente se encuentra registrada en el sistema.
<b>Excepciones:</b>

**Tabla III.5 Descripción del caso de uso Ingresar disponibilidad horaria de docente**

<b>Breve descripción:</b> El sistema permite modificar la disponibilidad horaria del docente.
<b>Actores:</b> Coordinador Académico y Docente.
<b>Precondición:</b> El docente ha ingresado su disponibilidad horaria.

<p><b>Descripción:</b> El sistema permite modificar :</p> <ol style="list-style-type: none"> <li>1. Los días y horas disponibles del docente.</li> <li>2. El nombre del curso de preferencia.</li> <li>3. La Escuela Académica Profesional a la que pertenece el curso.</li> <li>4. La Facultad a la que pertenece la Escuela Académica Profesional.</li> </ol>
<p><b>Postcondición:</b> La disponibilidad horaria del docente ha sido modificado.</p>
<p><b>Excepciones:</b> Se ha vencido el plazo para la modificación de la disponibilidad horaria.</p>

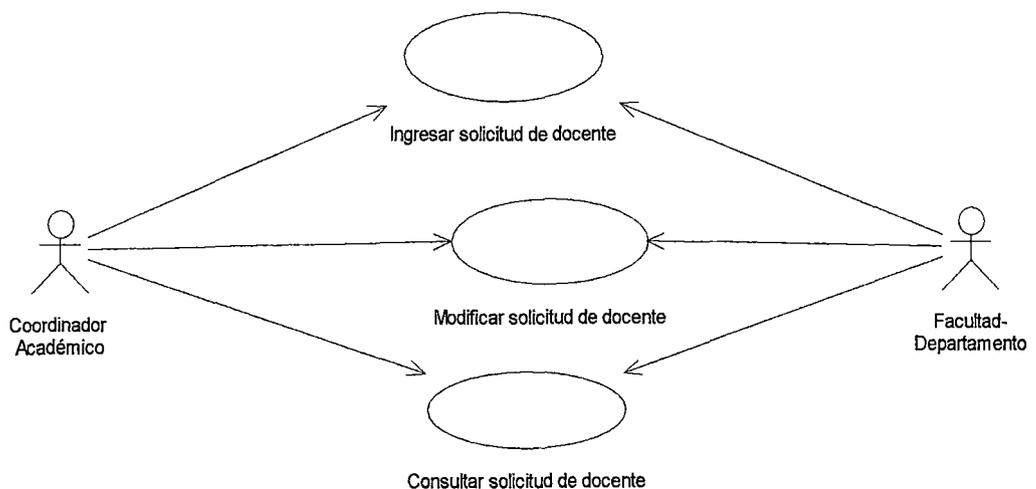
**Tabla III.6 Descripción del caso de uso Modificar disponibilidad horaria de docente**

<p><b>Breve descripción:</b> El sistema permite que el Docente pueda eliminar su disponibilidad horaria.</p>
<p><b>Actores:</b> Coordinador Académico y Docente.</p>
<p><b>Precondición:</b> La disponibilidad horaria ha sido ingresada.</p>
<p><b>Descripción:</b> El sistema elimina la disponibilidad horaria del docente dentro del plazo permitido.</p>
<p><b>Postcondición:</b> La disponibilidad horaria del docente ha sido eliminada.</p>
<p><b>Excepciones:</b> El plazo para realizar la eliminación de la disponibilidad horaria ha vencido.</p>

**Tabla III.7 Descripción del caso de uso Eliminar disponibilidad horaria de docente**

<b>Breve descripción:</b> El sistema permite que el docente y el Coordinador Académico puedan realizar la consulta de la disponibilidad horaria del docente.
<b>Actores:</b> Coordinador Académico, Docente y SUM.
<b>Precondición:</b> El docente ha ingresado su disponibilidad horaria.
<b>Descripción:</b> El sistema muestra la disponibilidad horaria del docente para el semestre académico a dictarse: 1. Los cursos de preferencia a dictarse con días y horas. 2. La Facultad y Escuela Académica que corresponden los cursos.
<b>Postcondición:</b>
<b>Excepciones:</b> El docente no ha ingresado su disponibilidad horaria dentro de la fecha permitida.

**Tabla III.8 Descripción del caso de uso Consultar disponibilidad horaria de docente.**



**Figura III.13 Diagrama de casos de uso solicitud de docente.**

**Breve descripción:** El sistema registra la solicitud docente de la Escuela

Académica de la Facultad y de otras Facultades.
<b>Actores:</b> El Coordinador Académico y Facultad-Departamento.
<b>Precondición:</b> Se ha registrado el horario de clases en el sistema.
<b>Descripción:</b> Se registra en el sistema: <ol style="list-style-type: none"> <li>1. El número de oficio de requerimiento de docente.</li> <li>2. La Facultad y Escuela Académica que corresponde.</li> <li>3. Los cursos con días y horas respectivamente.</li> <li>4. El nombre del docente sugerido que dictara el curso (opcional).</li> </ol>
<b>Postcondición:</b> La solicitud de docente se encuentra registrado en el sistema.
<b>Excepciones:</b>

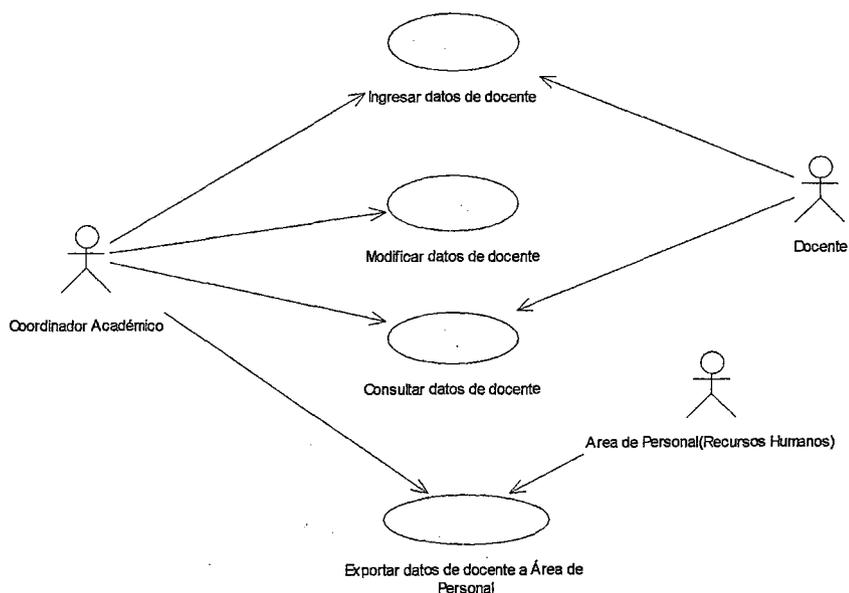
**Tabla III.9 Casos de uso: Ingresar solicitud de docente**

<b>Breve descripción:</b> El sistema modifica el requerimiento de docente.
<b>Actores:</b> Coordinador Académico y Facultad-Departamento.
<b>Precondición:</b> Se ha registrado la solicitud de docente.
<b>Descripción:</b> El sistema modifica: <ol style="list-style-type: none"> <li>1. La Facultad y Escuela Académica que corresponde.</li> <li>2. Los cursos con días y horas respectivamente.</li> <li>3. El nombre del docente sugerido que dictara el curso (opcional).</li> </ol>
<b>Postcondición:</b> La solicitud de docente ha sido modificada.
<b>Excepciones:</b>

**Tabla III.10 Descripción del caso de uso Modificar solicitud de docente.**

<b>Breve descripción:</b> El sistema permite consultar la solicitud de docente de la Facultad.
<b>Actores:</b> Coordinador Académico y Facultad-Departamento.
<b>Precondición:</b> La solicitud de docente ha sido ingresada.
<b>Descripción:</b> El sistema realiza la consulta de la solicitud de docente registrado en la Facultad. <ol style="list-style-type: none"> <li>1. El número de oficio de requerimiento de docente.</li> <li>2. La Facultad y Escuela Académica que corresponde.</li> <li>3. Los cursos con días y horas respectivamente.</li> <li>4. El nombre del docente sugerido que dictara el curso (opcional).</li> </ol>
<b>Postcondición:</b>
<b>Excepciones:</b>

**Tabla III.11 Descripción del caso de uso Consultar solicitud de docente.**



**Figura III.14 Diagrama de casos de uso de datos de docente.**

<b>Breve descripción:</b> El sistema registra la información de los docentes que pertenecen a la Facultad en coordinación con el Área de Personal de la Universidad.
<b>Actores:</b> Coordinador Académico y Docente.
<b>Precondición:</b> El Área de Personal registra información del docente por Resolución Rectoral o Servicios no Personales.
<p><b>Descripción:</b> El sistema registra información del docente: El sistema de información mantendrá los datos actualizados del docente en comunicación con el Sistema de Personal de la Universidad.</p> <ol style="list-style-type: none"> <li>1. Código docente.</li> <li>2. Nombre docente.</li> <li>3. Dni.</li> <li>4. Dirección.</li> <li>5. Teléfono.</li> <li>6. Dirección electrónica.</li> <li>7. Número de Resolución Rectoral.</li> <li>8. Categoría.</li> <li>9. Clase.</li> <li>10. Tiempo de servicios.</li> <li>11. Grados obtenidos.</li> <li>12. Títulos obtenidos.</li> </ol>
<b>Postcondición:</b> Los datos del docente se encuentran registrados en el sistema.
<b>Excepciones:</b> La Resolución Rectoral no ha sido emitida.

**Tabla III.12 Descripción del caso de uso Ingresar datos de docente.**

<b>Breve descripción :</b> El sistema permite modificar los datos del docente.
<b>Actores:</b> Coordinador Académico.
<b>Precondición:</b> Los datos del docente se encuentran registrados en el sistema.

<p><b>Descripción:</b> El sistema permite modificar los campos:</p> <ol style="list-style-type: none"> <li>1. Teléfono.</li> <li>2. Dirección.</li> <li>3. Dirección electrónica.</li> <li>4. Categoría.</li> <li>5. Clase.</li> <li>6. Tiempo de servicios.</li> <li>7. Grados obtenidos.</li> <li>8. Títulos obtenidos.</li> </ol>
<p><b>Postcondición:</b> Los datos del docente han sido modificados.</p>
<p><b>Excepciones:</b></p>

**Tabla III.13 Descripción del caso de uso Modificar datos de docente**

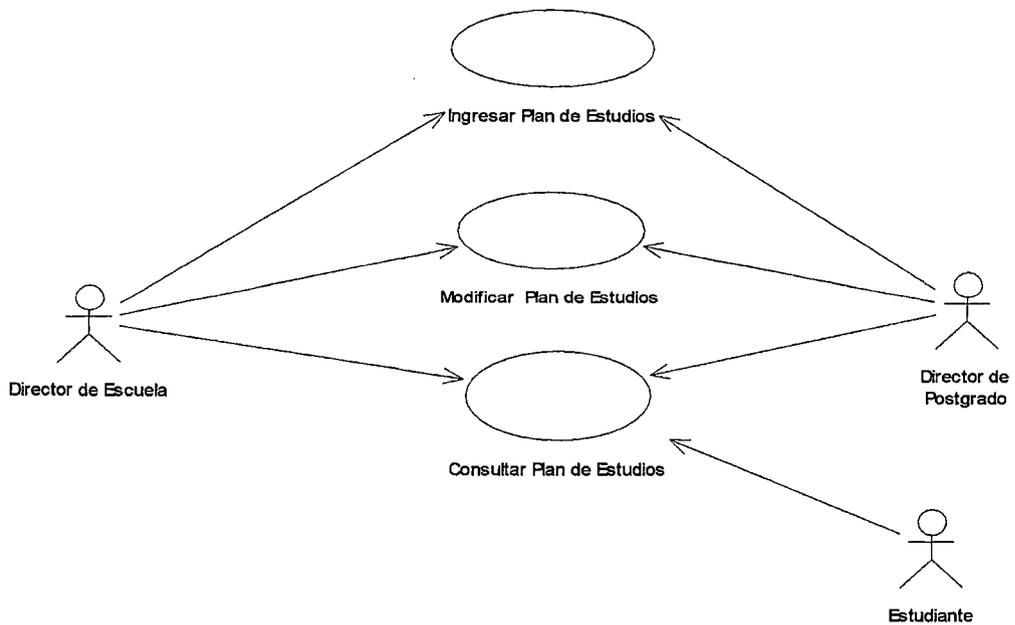
<p><b>Breve descripción:</b> El sistema permite visualizar los datos del docente que se han registrado.</p>
<p><b>Actores:</b> Coordinador Académico y Docente.</p>
<p><b>Precondición:</b> Los datos del docente se encuentran registrados en el sistema.</p>
<p><b>Descripción:</b> El sistema permite visualizar los datos del docente que se encuentran registrado en el sistema.</p> <ol style="list-style-type: none"> <li>1. Nombre docente.</li> <li>2. Dni.</li> <li>3. Dirección.</li> <li>4. Teléfono.</li> <li>5. Dirección electrónica.</li> <li>6. Número de Resolución Rectoral.</li> <li>7. Categoría.</li> </ol>

8. Clase.
9. Tiempo de servicios.
10. Grados obtenidos.
11. Títulos obtenidos.
<b>Postcondición:</b>
<b>Excepciones:</b>

**Tabla III.14 Descripción del caso de uso Consultar datos de docente**

<b>Breve descripción:</b> El sistema exportara los datos del docente de la Facultad al Área de Personal de la Universidad.
<b>Actores:</b> Coordinador Académico y Área de Personal (Recursos Humanos).
<b>Precondición:</b> Los datos del docente han sido actualizados.
<b>Descripción:</b> Mensualmente el sistema permitirá la exportación de datos del docente al Área de Personal de la Universidad.
<b>Postcondición:</b> Los datos han sido actualizados en el Área de Personal.
<b>Excepciones:</b> Durante el mes no se ha realizado actualización de datos del docente.

**Tabla III.15 Descripción del caso de uso Exportar datos de docente al Área de Personal.**



**Figura III.15 Diagrama de casos de uso de Plan de Estudios**

<b>Breve descripción:</b> El sistema permite ingresar el Plan de Estudios de la Escuela Académica Profesional.
<b>Actores:</b> Director de Escuela y Director de Postgrado.
<b>Precondición:</b>
<b>Descripción:</b> Se ingresa el plan de estudios para la Escuela Académica Profesional.
<ol style="list-style-type: none"> <li>1. Código de Escuela Académica Profesional.</li> <li>2. Código del curso.</li> <li>3. Nombre del curso.</li> <li>4. Cantidad de créditos.</li> </ol>

<b>Postcondición:</b> El Plan de Estudio se encuentra registrado en el Sistema.
<b>Excepciones:</b>

**Tabla III.16 Descripción del caso de uso Ingresar Plan de Estudios**

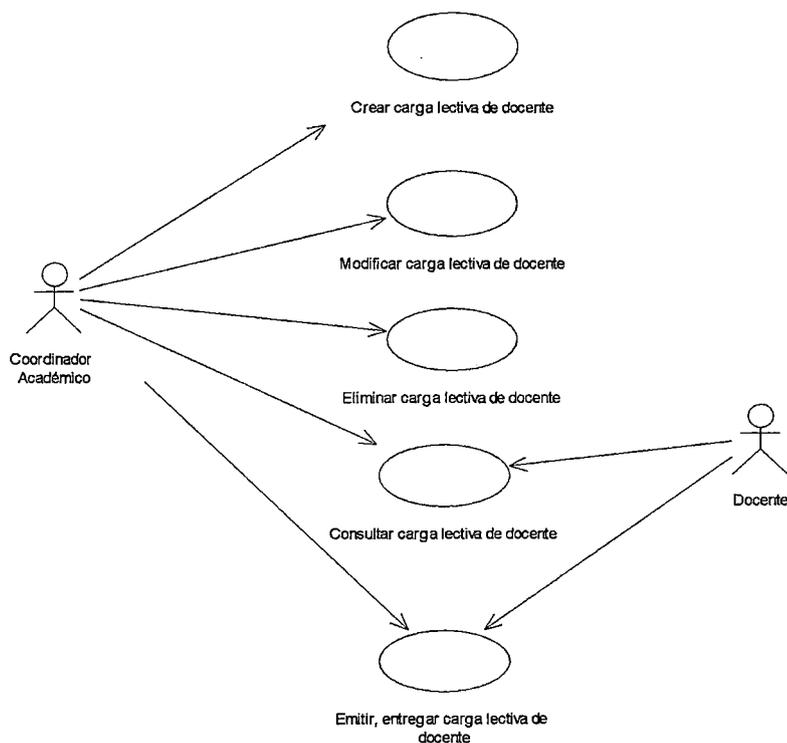
<b>Breve de Descripción:</b> El sistema permite modificar el Plan de Estudio de la Escuela Académica Profesional.
<b>Actores:</b> Director de Escuela y Director de Postgrado.
<b>Precondición:</b>
<b>Descripción:</b> Se modifica los datos los datos ingresados del plan de estudio, como: <ul style="list-style-type: none"> <li>1. Código del curso.</li> <li>2. Nombre del curso.</li> <li>3. Cantidad de créditos.</li> </ul> <p>El sistema podrá realizar las operaciones de:</p> <ul style="list-style-type: none"> <li>1. Adición de cursos.</li> <li>2. Eliminación de cursos.</li> </ul>
<b>Postcondición:</b> Los datos del Plan de Estudio han sido modificados.
<b>Excepciones:</b>

**Tabla III.17 Descripción del caso de uso Modificar Plan de Estudios**

<b>Breve descripción:</b> El sistema permite visualizar los datos ingresados del Plan de Estudio.
<b>Actores:</b> Director de Escuela, Director de Postgrado y Estudiante.

<b>Precondición:</b>
<b>Descripción:</b> El sistema muestra los datos ingresados del Plan de estudios de Escuela Académica Profesional.
<ol style="list-style-type: none"> <li>1. Código de Escuela Académica Profesional.</li> <li>2. Código del curso.</li> <li>3. Nombre del curso.</li> <li>4. Cantidad de créditos.</li> </ol>
<b>Postcondición:</b>
<b>Excepciones:</b>

**Tabla III.18 Descripción del caso de uso Consultar Plan de Estudios**



**Figura III.16 Diagrama de casos de uso de carga lectiva de docente**

<p><b>Breve descripción:</b> El sistema permite crear la carga lectiva para el docente con los cursos que dictara en el Semestre Académico.</p>
<p><b>Actores:</b> Coordinador Académico.</p>
<p><b>Precondición:</b> El horario del semestre, la disponibilidad horaria del docente y la solicitud de docente han sido ingresados al sistema.</p>
<p><b>Descripción:</b> El sistema crea la carga lectiva del docente en base al horario del semestre académico, la disponibilidad horaria del docente y la solicitud de docente de la Facultad.</p> <ol style="list-style-type: none"> <li>1. Se selecciona el nombre o se ingresa el código del docente.</li> <li>2. Se selecciona el curso y se ingresa en la carga lectiva del docente.</li> <li>3. Se ingresa el horario del curso según la disponibilidad horaria del docente y el horario del Semestre Académico.</li> <li>4. Se selecciona el nombre de la Escuela Académica que pertenece el curso o el nombre de la Facultad y el nombre de la Escuela Académica si el curso pertenece a otra Facultad.</li> <li>5. Suma la cantidad de horas por curso.</li> <li>6. Repetir el paso 2 hasta el paso 5 hasta completar la cantidad de horas según la clase del docente: tiempo parcial, tiempo completo o dedicación exclusiva.</li> </ol>
<p><b>Postcondición:</b> La carga lectiva ha sido creada en el sistema.</p>
<p><b>Excepciones:</b> La disponibilidad horaria del docente no ha sido ingresada al sistema. Los horarios de los cursos solicitados no coinciden con la disponibilidad horaria del docente. Los cursos solicitados ya tienen docentes.</p>

**Tabla III.19 Descripción del caso de uso Crear carga lectiva de docente**

<b>Breve descripción:</b> El sistema permite modificar los datos de la carga lectiva del docente.
<b>Actores:</b> Coordinador Académico.
<b>Precondición:</b> La carga lectiva ha sido ingresada en el sistema.
<p><b>Descripción:</b> El sistema permite realizar modificaciones de los dato de la carga lectiva.</p> <ol style="list-style-type: none"> <li>1. Se selecciona el nombre o se ingresa el código del docente.</li> <li>2. Se selecciona el curso a modificar y se ingresa el nuevo curso del horario del Semestre Académico o de la solicitud de docente a la carga lectiva del docente.</li> <li>3. Se modifica el horario del curso según la disponibilidad horaria del docente y el horario del Semestre Académico.</li> <li>4. Se selecciona el nombre de la Escuela Académica que pertenece el curso o el nombre de la Facultad y el nombre de la Escuela Académica si el curso pertenece a otra Facultad.</li> <li>5. Se suma la cantidad de horas del curso a la cantidad de horas que dictara el docente.</li> <li>6. Repetir el paso 2 hasta el paso 5 hasta completar la cantidad de horas según la clase del docente: tiempo parcial, tiempo completo o dedicación exclusiva.</li> </ol> <p>El sistema permite:</p> <ol style="list-style-type: none"> <li>1. Adicionar un curso.</li> <li>2. Eliminar un curso.</li> </ol>
<b>Postcondición:</b> La carga lectiva ha sido modificada.
<b>Excepciones:</b>

**Tabla III.20 Descripción del caso de uso:  
Modificar carga lectiva de docente**

<b>Breve descripción:</b> El sistema permite eliminar la carga lectiva del docente.
<b>Actores:</b> Coordinador Académico.
<b>Descripción:</b> El sistema elimina la carga lectiva del docente. <ol style="list-style-type: none"> <li>1. Se selecciona el código o el nombre del docente.</li> <li>2. Se elimina todos los datos de la carga lectiva.</li> </ol>
<b>Postcondición:</b> La carga lectiva del docente ha sido eliminada.
<b>Excepciones:</b>

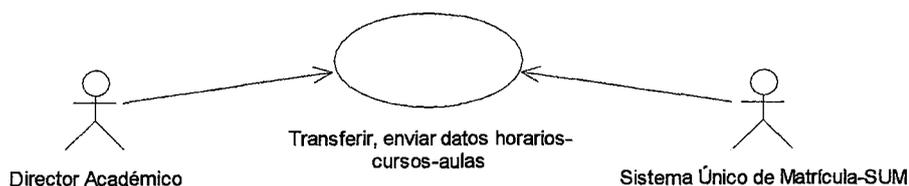
**Tabla III.21 Descripción del caso de uso: Eliminar carga lectiva de docente**

<b>Breve descripción:</b> El sistema permite realizar la consulta de la carga lectiva del docente.
<b>Actores:</b> Coordinador Académico y Docente.
<b>Precondición:</b>
<b>Descripción:</b> El sistema permite visualizar los datos de la carga lectiva: <ol style="list-style-type: none"> <li>1. Código del docente.</li> <li>2. Nombre del docente.</li> <li>3. Curso asignado.</li> <li>4. Nombre de la Facultad.</li> <li>5. Escuela Académica.</li> <li>6. Cantidad de horas del curso.</li> <li>7. Horario del curso.</li> <li>8. Cantidad total de horas asignadas.</li> </ol>
<b>Postcondición:</b>
<b>Excepciones:</b> La carga lectiva fue eliminada del sistema.

**Tabla III.22 Descripción del caso de uso Consultar carga lectiva de docente**

<b>Breve descripción:</b> El sistema emite la carga lectiva para ser entregada al docente.
<b>Actores:</b> Coordinador Académico y Docente.
<b>Precondición:</b> La carga lectiva ha sido creada en el sistema.
<b>Descripción:</b> El sistema emite la carga lectiva del docente.
<b>Postcondición:</b> La carga lectiva ha sido emitida por el sistema y entregado al docente.
<b>Excepciones:</b>

**Tabla III.23 Descripción del caso de uso Emitir, entregar carga lectiva de docente**

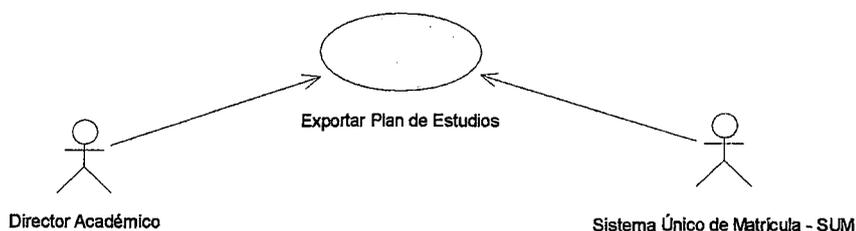


**Figura III.17 Diagrama de casos de uso de Transferir, enviar datos horarios-cursos-aulas.**

<b>Breve descripción:</b> El sistema transfiere o exporta los horarios de los cursos al Sistema Único de Matricula.
<b>Precondición:</b> Los horarios-cursos-aulas tienen asignado el docente que dictará el curso.
<b>Actores:</b> Director Académico y Sistema Único de Matricula.
<b>Descripción:</b> El sistema permitirá exportar los horarios de los cursos al Sistema

Único de Matricula antes del inicio de la matricula del Semestre Académico.
<b>Postcondición:</b> Los horarios de los cursos a dictar en el semestre académico han sido exportados por el sistema.
<b>Excepciones:</b>

**Tabla III.24 Descripción del caso de uso Transferir, enviar datos horarios-cursos-aulas**



**Figura III.18 Diagrama de casos de uso: Exportar Plan de Estudios.**

<b>Breve descripción:</b> El sistema exportará el plan de estudios al Sistema Único de Matrícula.
<b>Actores:</b> Director Académico y Sistema Único de Matricula.
<b>Precondición:</b> Los datos del Plan de Estudios han sido ingresados al sistema.
<b>Descripción:</b> El sistema exportara los datos del plan de estudios al Sistema Único de Matricula antes del inicio de la matricula del Semestre Académico.
<b>Postcondición:</b> Los datos del Plan de Estudios son almacenados por el Sistema Único de Matricula.
<b>Excepciones:</b>

**Tabla III.25 Descripción del caso de uso Exportar Plan de Estudios**

### **3.2.3.2 Refinamiento del diagrama de clases.-**

En la sección anterior el diagrama de casos de uso fue modificado, en consecuencia el diagrama el diagrama de clases debe ser modificado, pasaremos a realizar una siguiente iteración, obteniéndose el siguiente resultado como se observa en la Figura III.19.

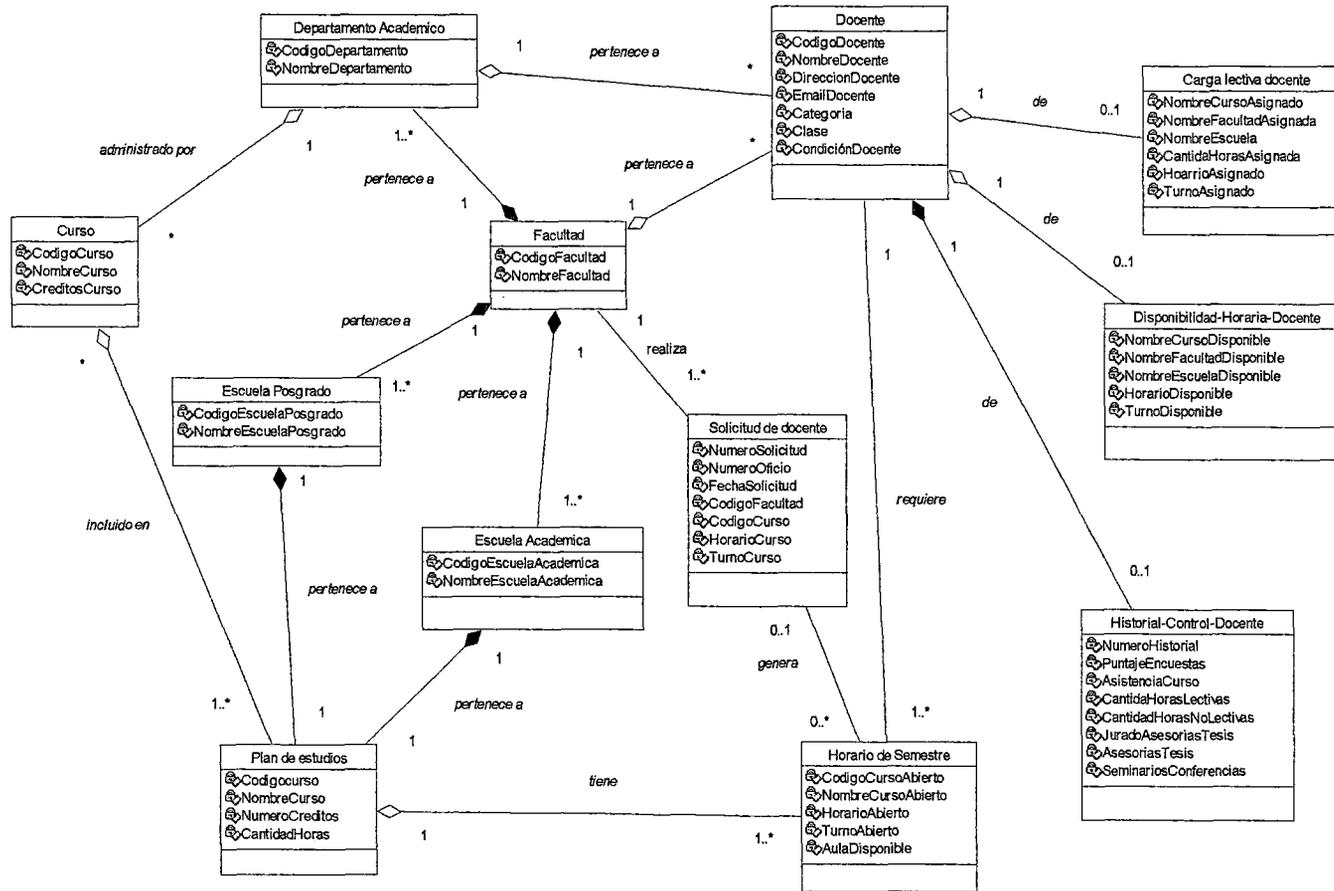
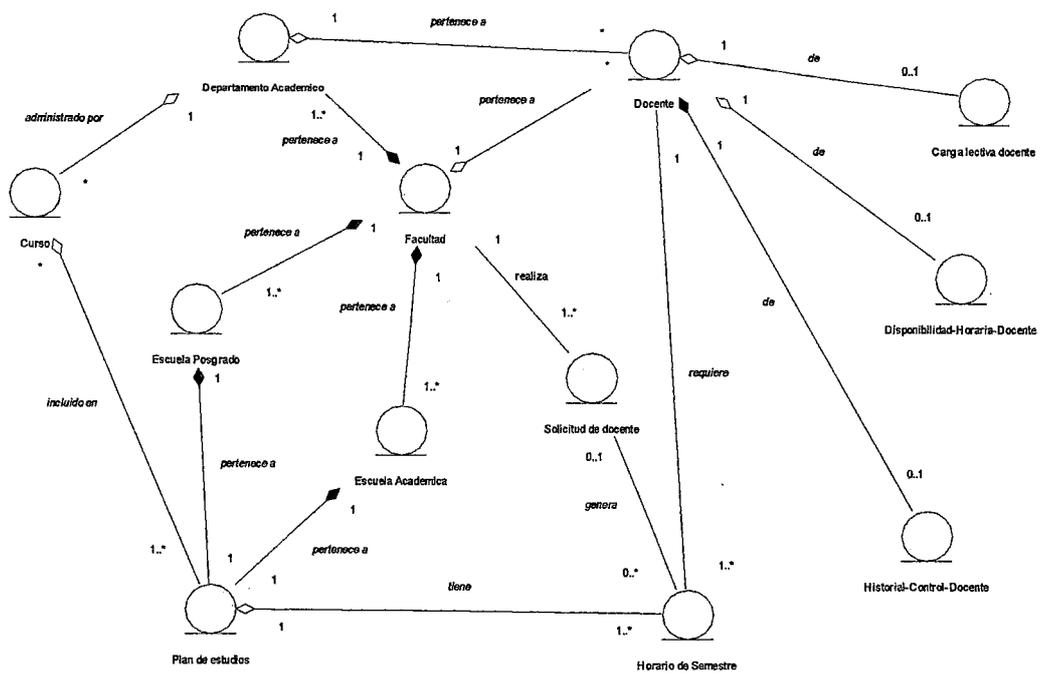


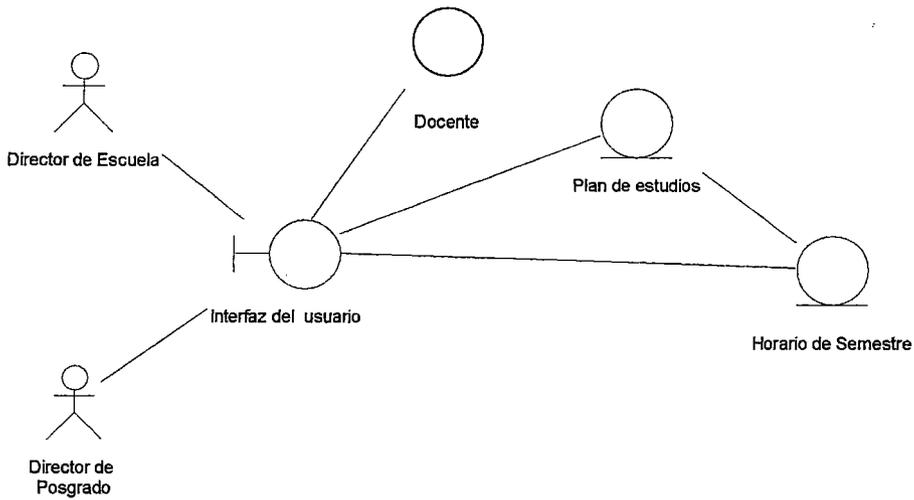
Figura III.19 Diagrama de clases refinado con atributos y relaciones

Anteriormente se represento las clases entidades como estereotipos, nuevamente en esta sección se presenta el diagrama modificado, como se muestra en la Figura III.20.

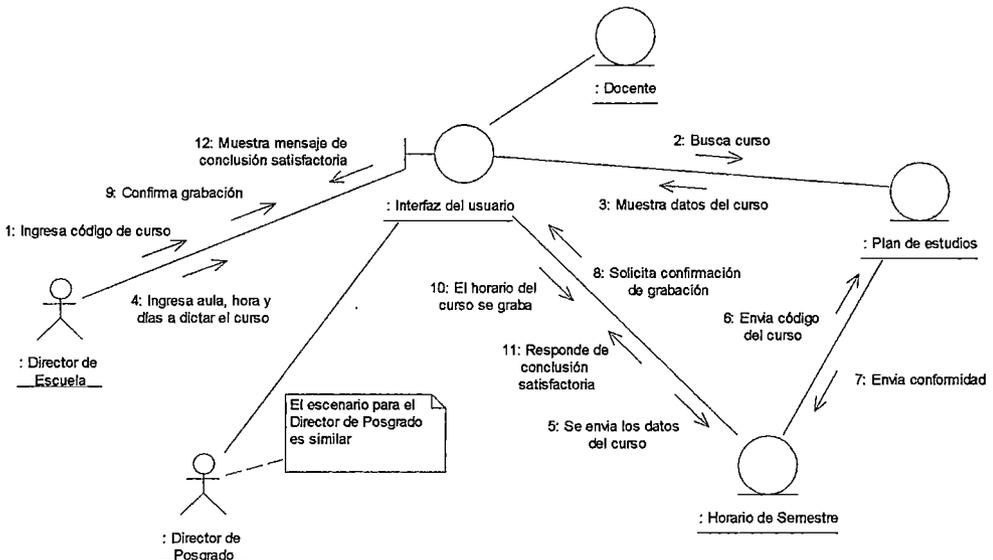


**Figura III.20 Diagrama de clases refinado que muestra los estereotipos.**

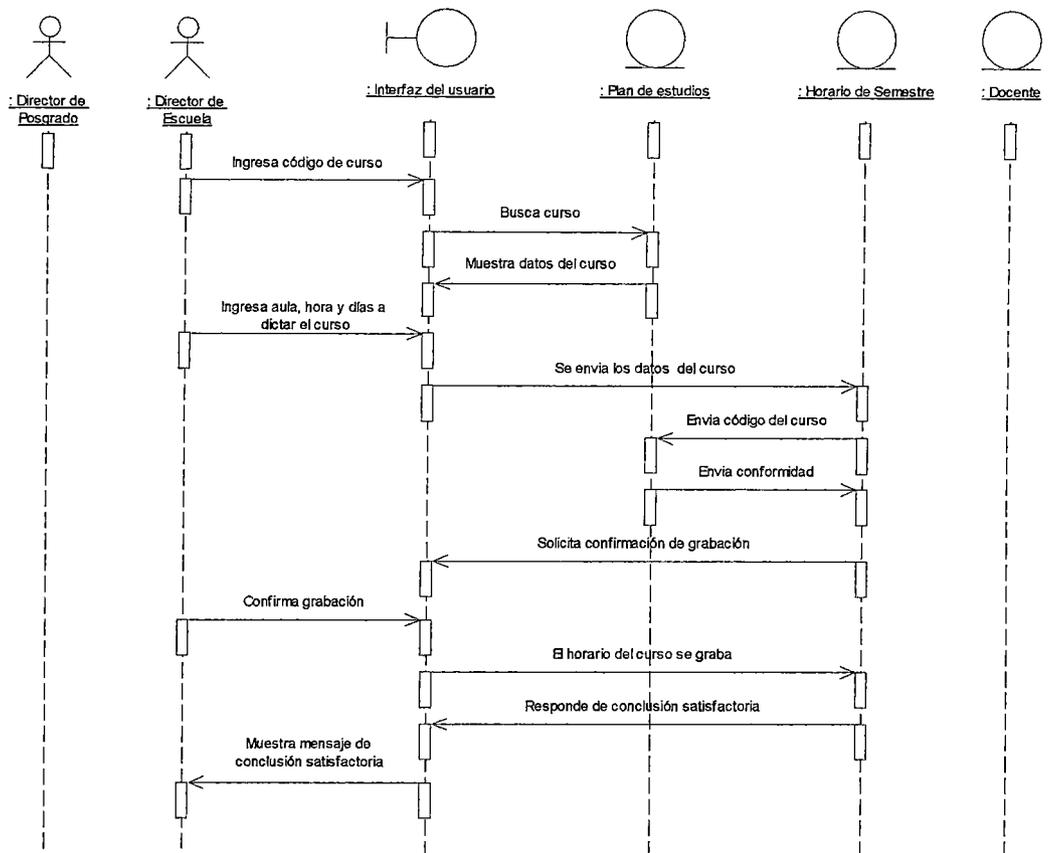
**3.2.3.3 Realización de los casos de usos.-** Tenemos los diagramas de clases (similar al diagrama de colaboración) y los diagramas de interacción: colaboración y secuencia de los casos de uso del Modelo del Sistema de Información. A continuación se muestra en las Figuras III.21, III.22, III.23, III.24, III.25, III.26, III.27, III.28, III.29, III.30, III.31, III.32, III.33, III.34, III.35, III.36, III.37, III.38, III.39, III.40, III.41, III.42, III.43, III.44, III.45, III.46, III.47, III.48, III.49, III.50, III.51, III.52 y III.53 respectivamente.



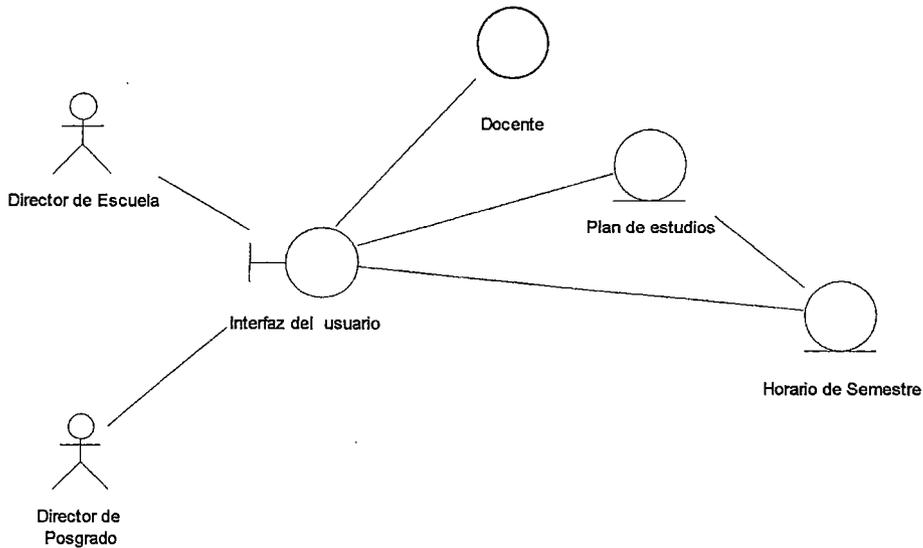
**Figura III.21 Diagrama de clase que muestra las clases que realizan el caso de uso Ingresar horario de clases**



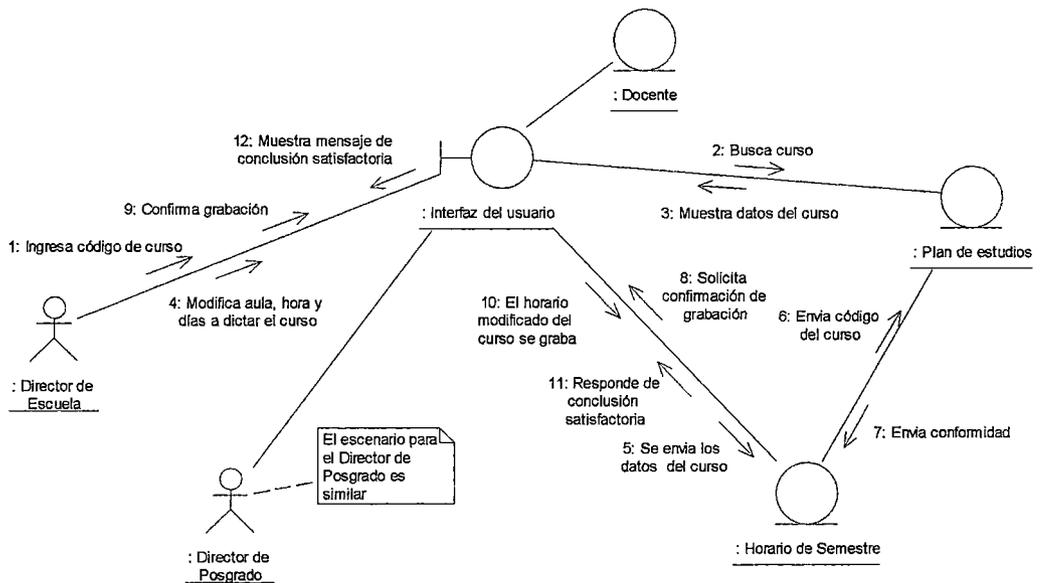
**Figura III.22 Diagrama de colaboración del escenario del caso de uso Ingresar horario de clases**



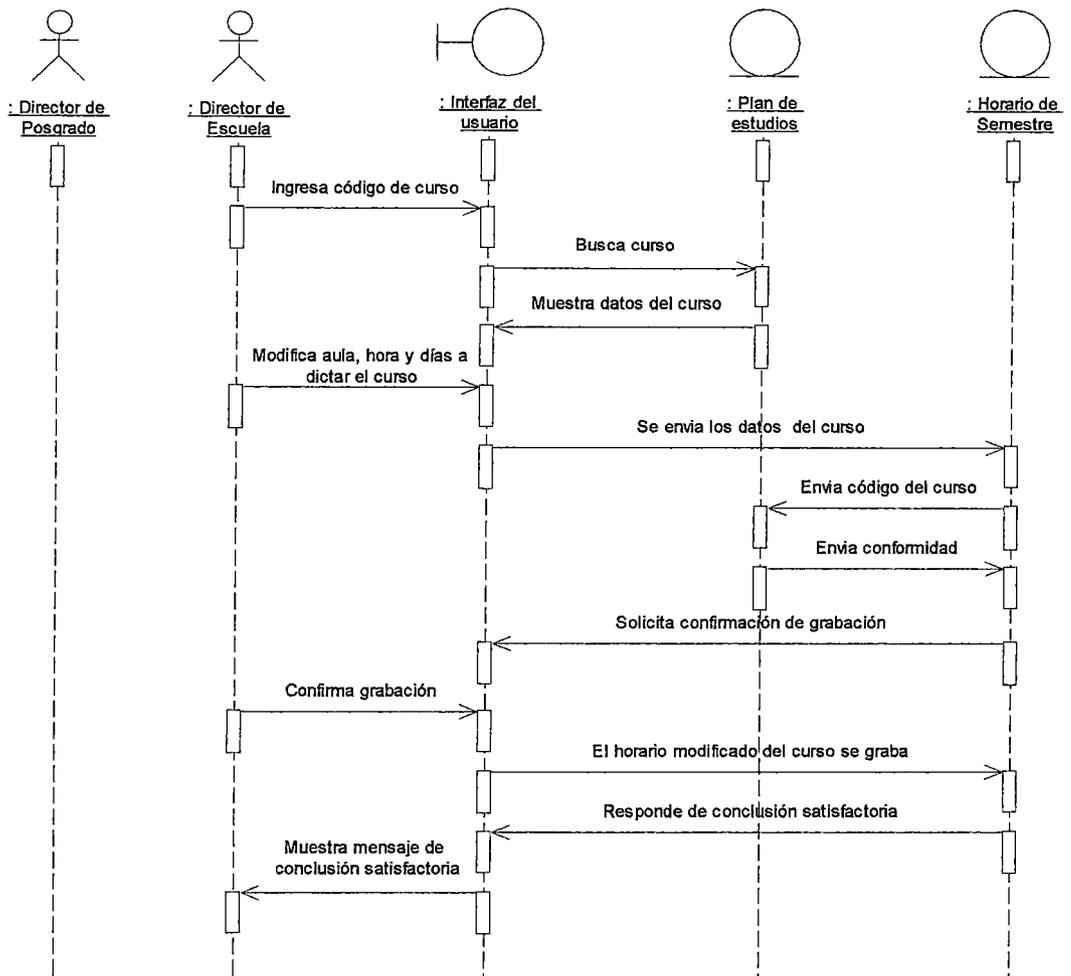
**Figura III.23 Diagrama de secuencia del escenario del caso de uso Ingresar horario de clases**



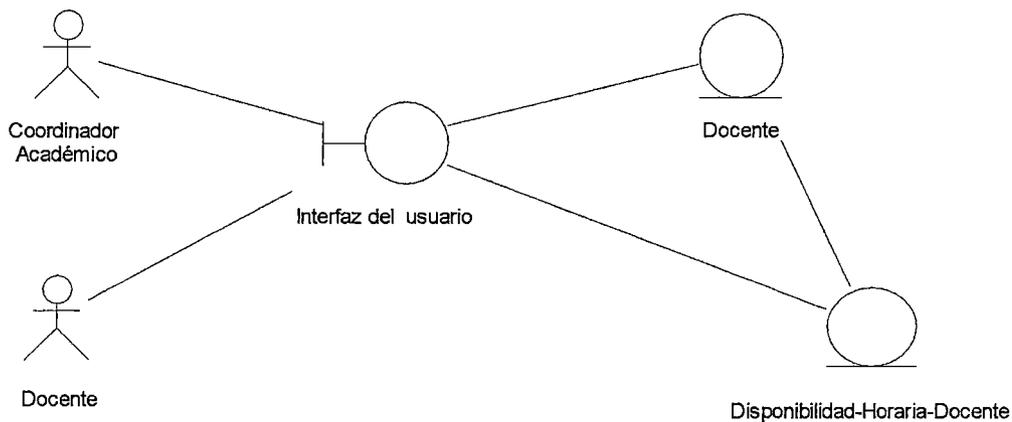
**Figura III.24 Diagrama de clase que muestra las clases que realizan el caso de uso Modificar horario de clases**



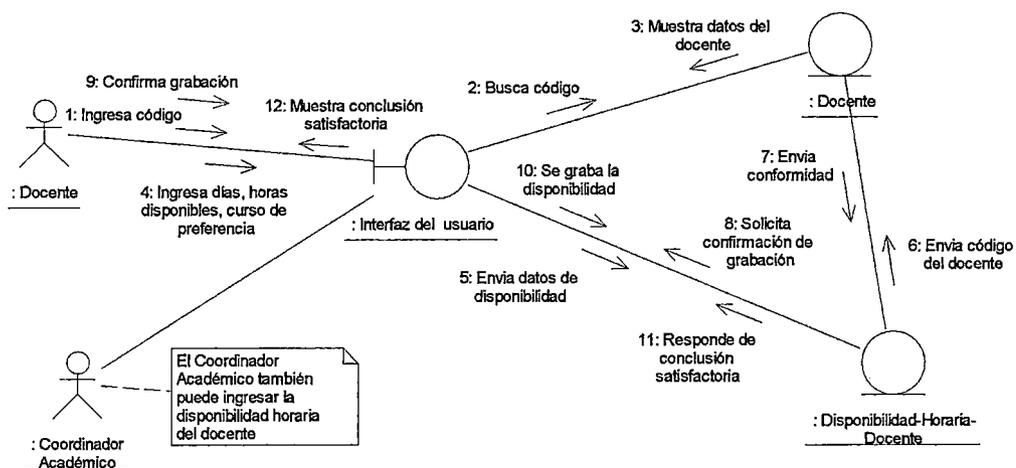
**Figura III.25 Diagrama de colaboración del escenario del caso de uso Modificar horario de clases**



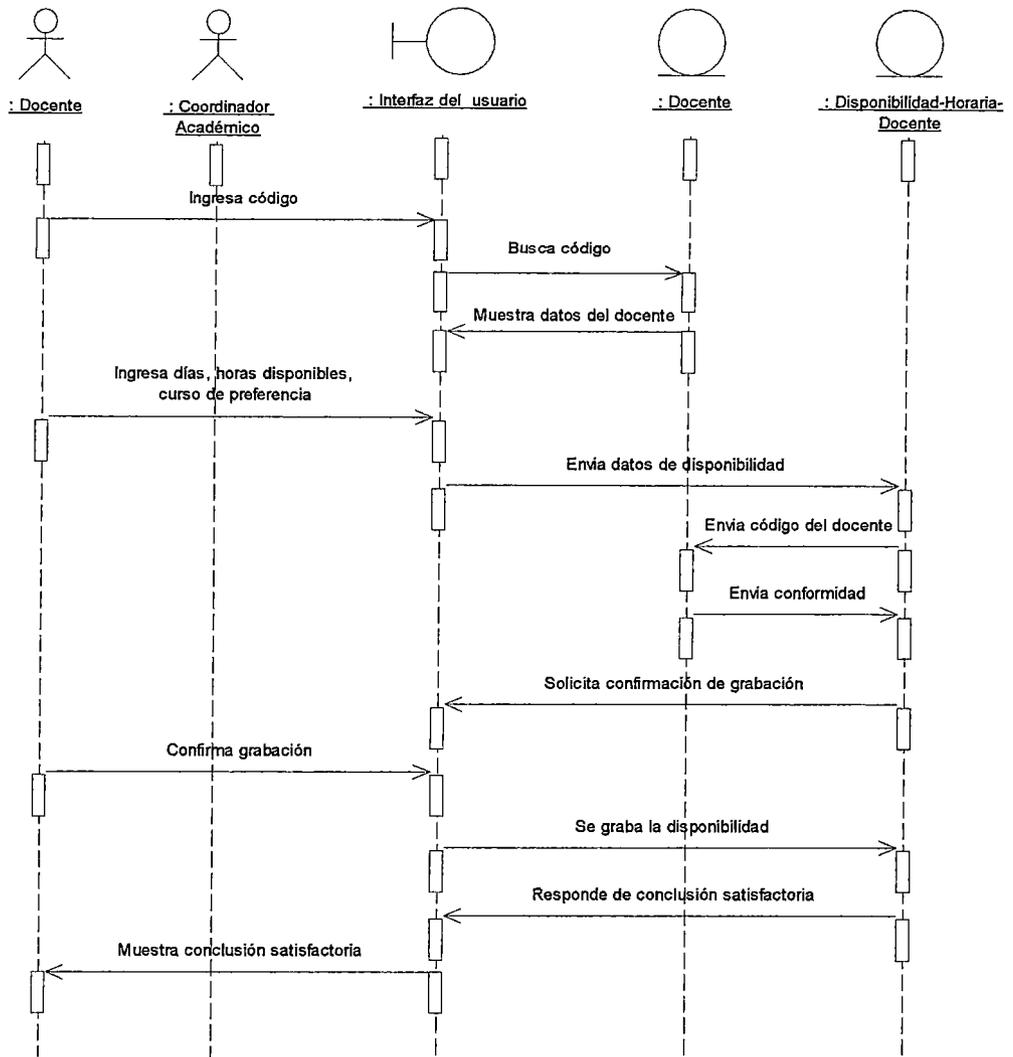
**Figura III.26 Diagrama de secuencia del escenario del caso de uso Modificar horario de clases**



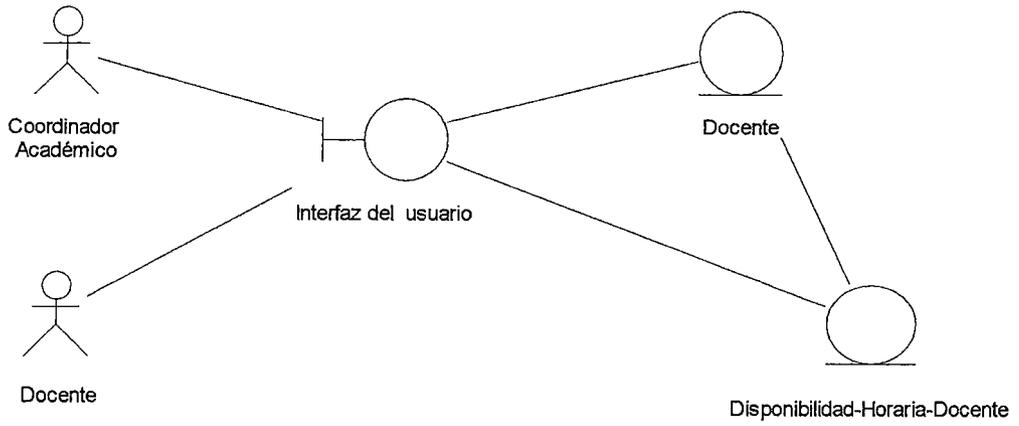
**Figura III.27 Diagrama de clase que muestra las clases que realizan el caso de uso Ingresar disponibilidad horaria docente**



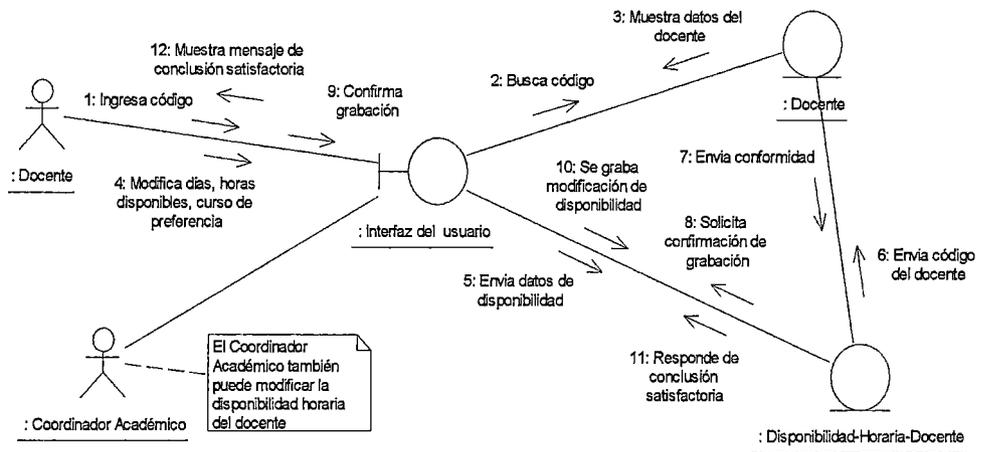
**Figura III.28 Diagrama de colaboración del escenario del caso de uso Ingresar disponibilidad horaria docente**



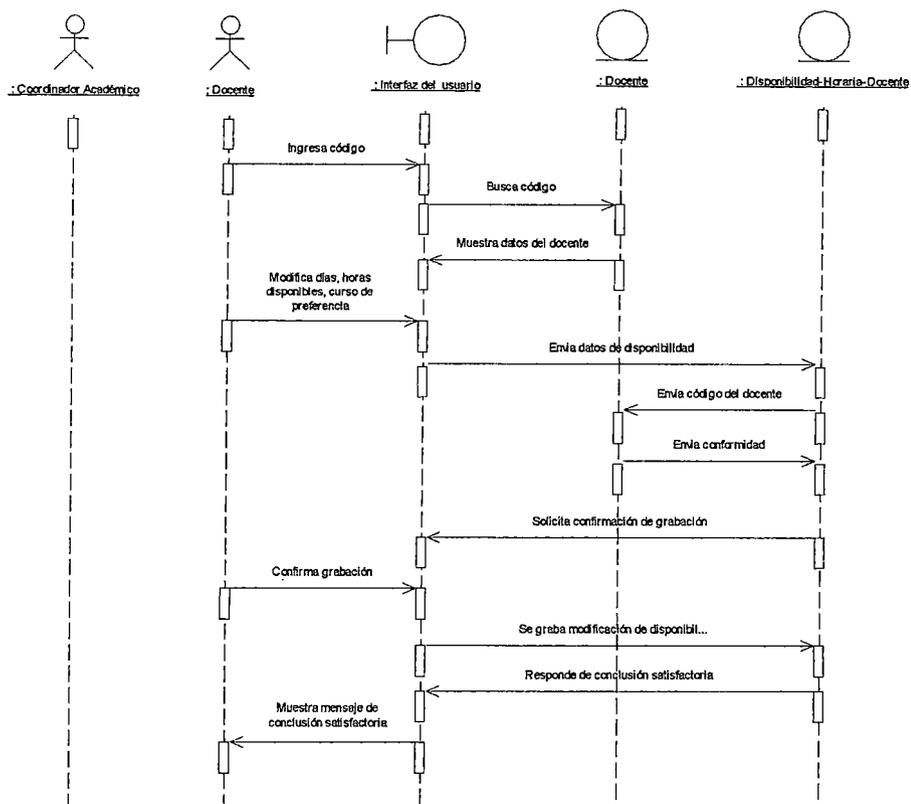
**Figura III.29 Diagrama de secuencia del escenario del caso de uso Ingresar disponibilidad horaria docente**



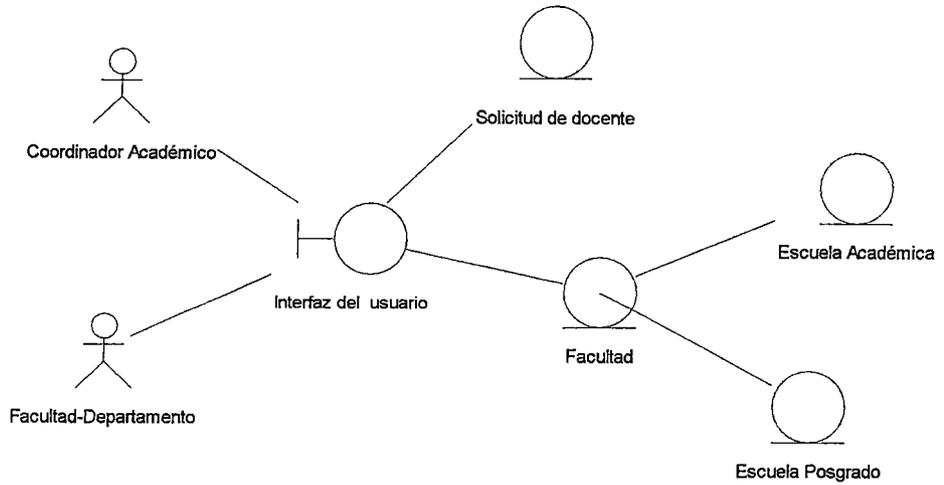
**Figura III.30 Diagrama de clase que muestra las clases que realizan el caso de uso Modificar disponibilidad horaria docente**



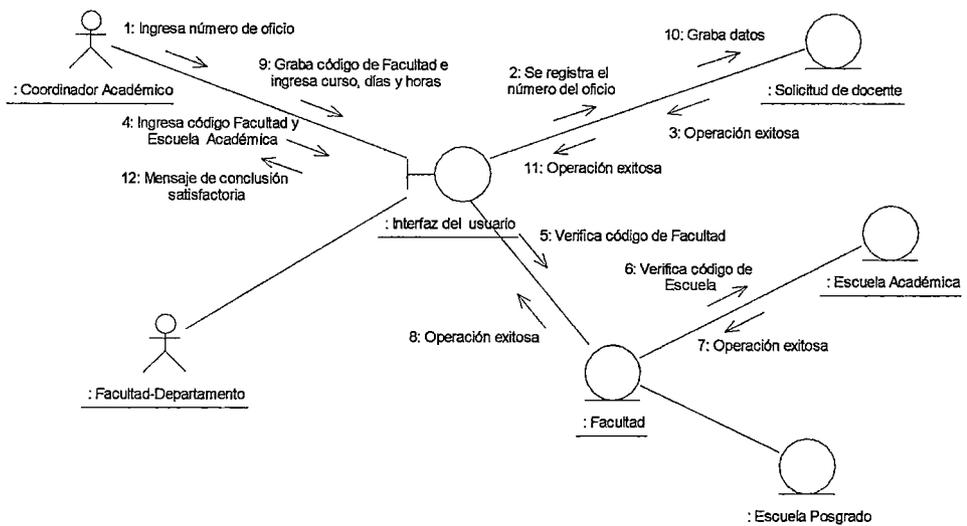
**Figura III.31 Diagrama de colaboración del escenario del caso de uso Modificar disponibilidad horaria docente**



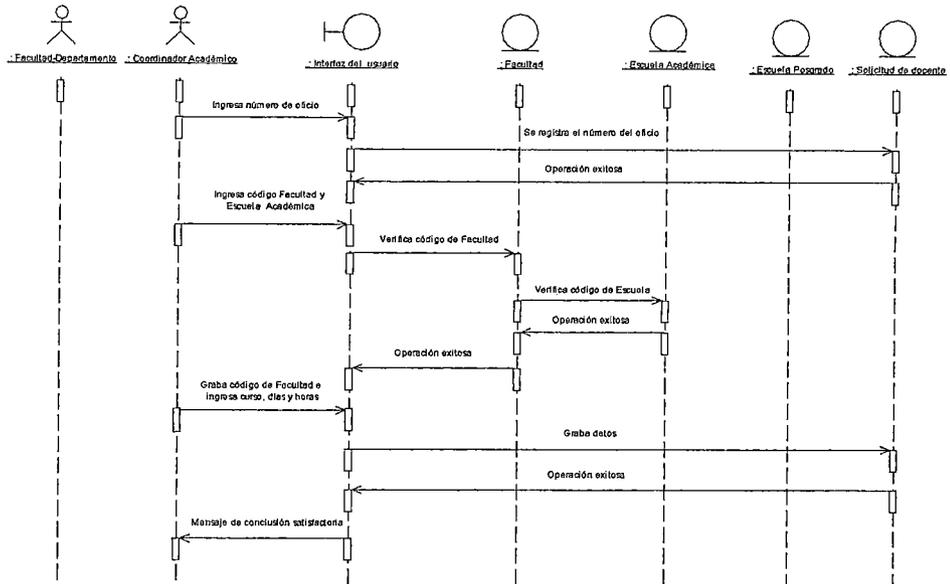
**Figura III.32 Diagrama de secuencia del escenario del caso de uso Modificar disponibilidad horaria docente**



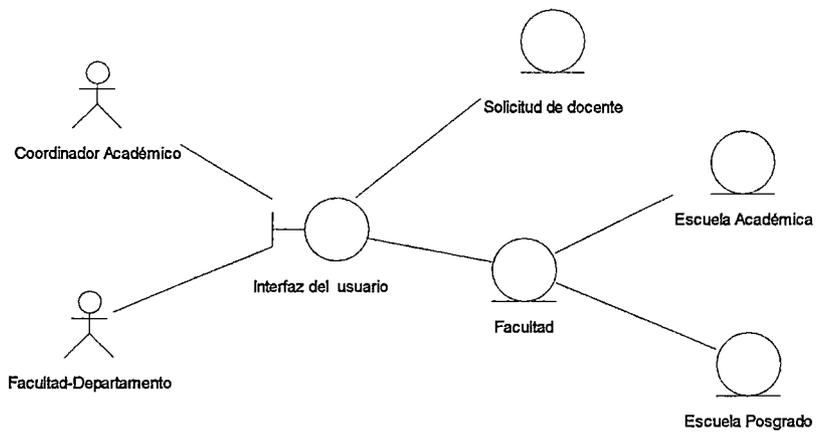
**Figura III.33 Diagrama de clase que muestra las clases que realizan el caso de uso Ingresar solicitud de docente**



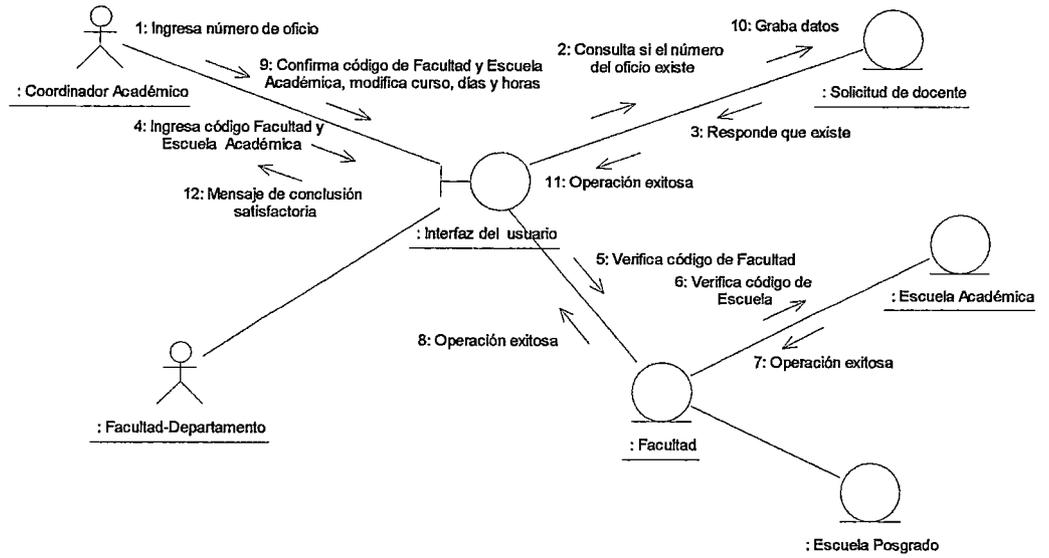
**Figura III.34 Diagrama de colaboración del escenario del caso de uso Ingresar solicitud de docente**



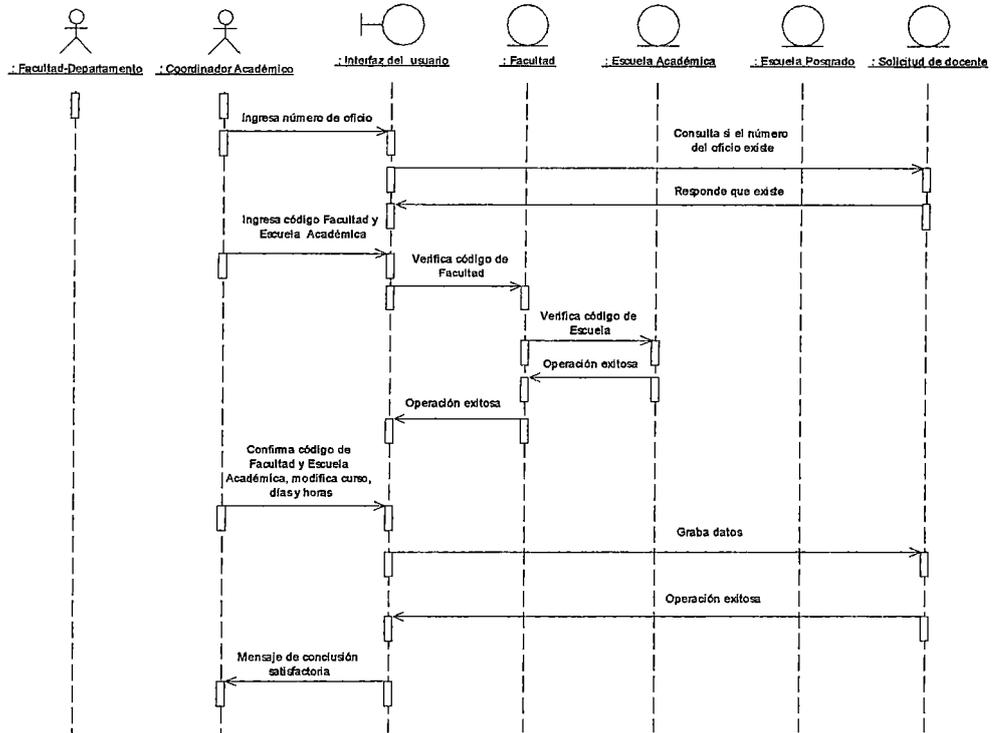
**Figura III.35 Diagrama de secuencia del escenario del caso de uso Ingresar solicitud de docente**



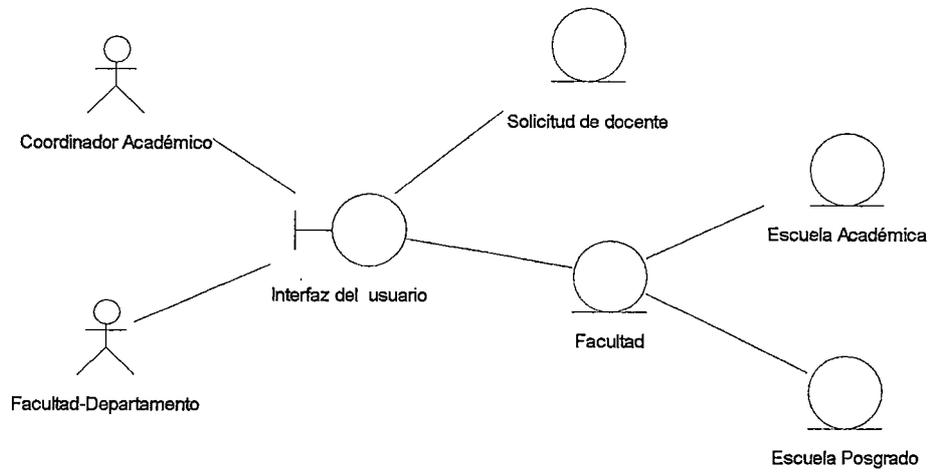
**Figura III.36 Diagrama de clase que muestra las clases que realizan el caso de uso Modificar solicitud de docente**



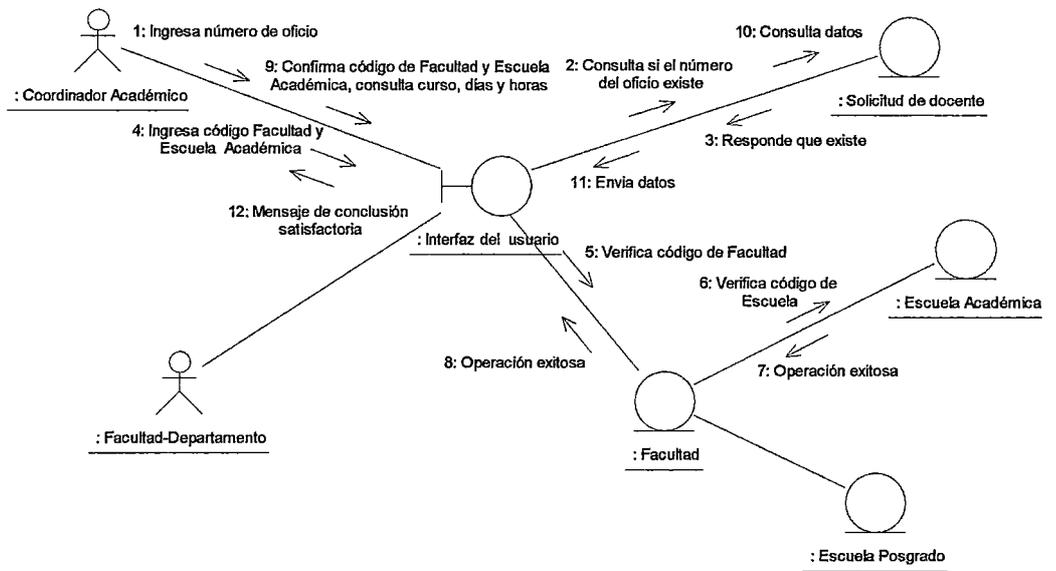
**Figura III.37 Diagrama de colaboración del escenario del caso de uso  
Modificar solicitud de docente**



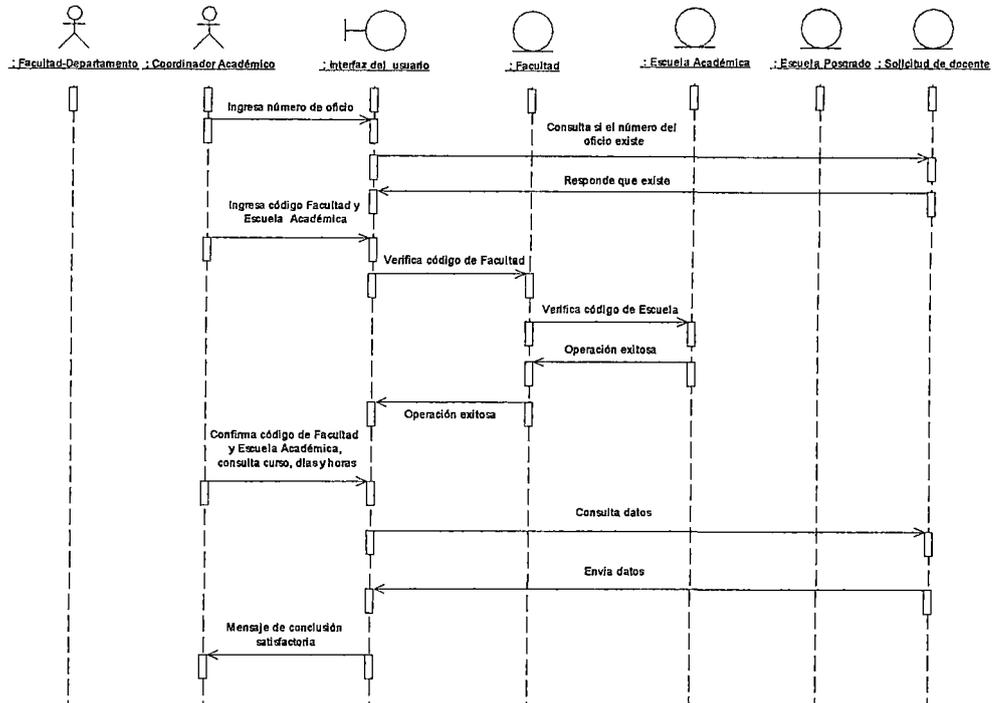
**Figura III.38 Diagrama de secuencia del escenario del caso de uso  
Modificar solicitud de docente**



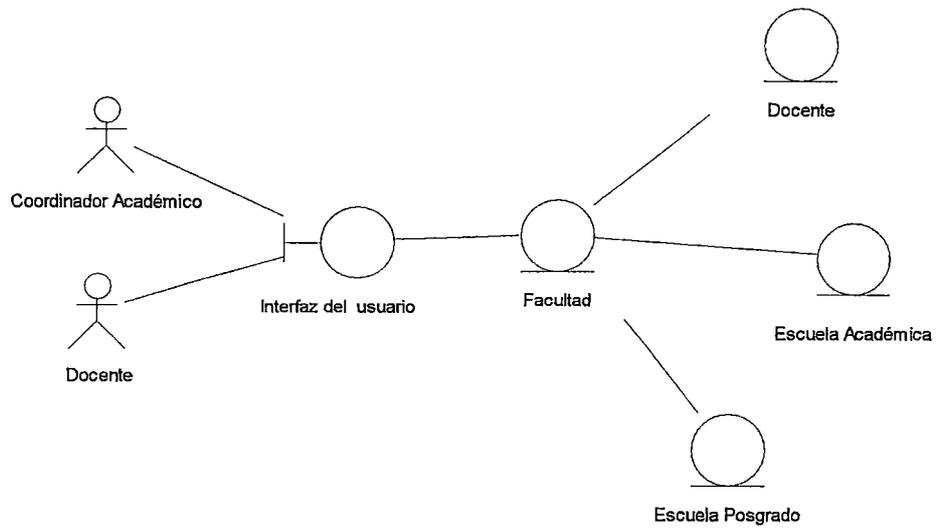
**Figura III.39 Diagrama de clase que muestra las clases que realizan el caso de uso Consultar solicitud de docente**



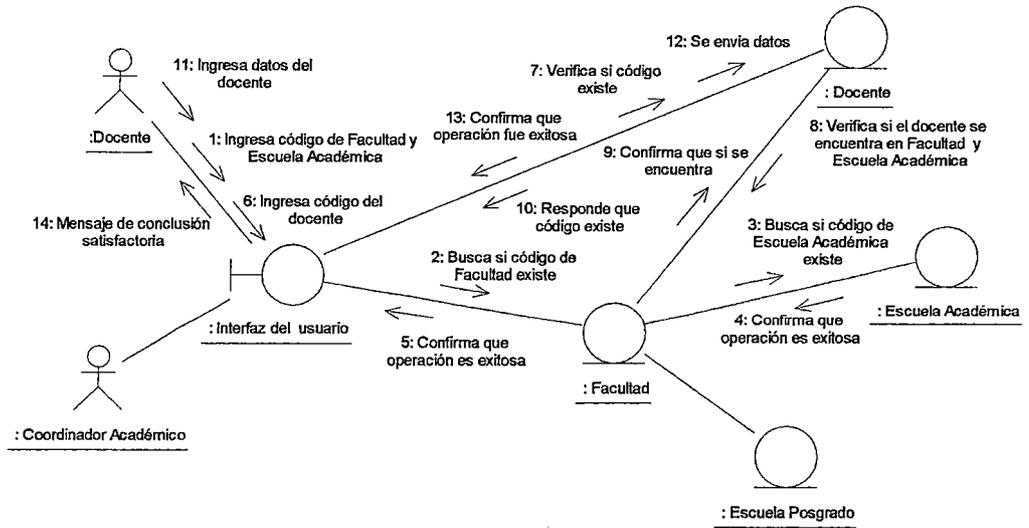
**Figura III.40 Diagrama de colaboración del escenario del caso de uso Consultar solicitud de docente**



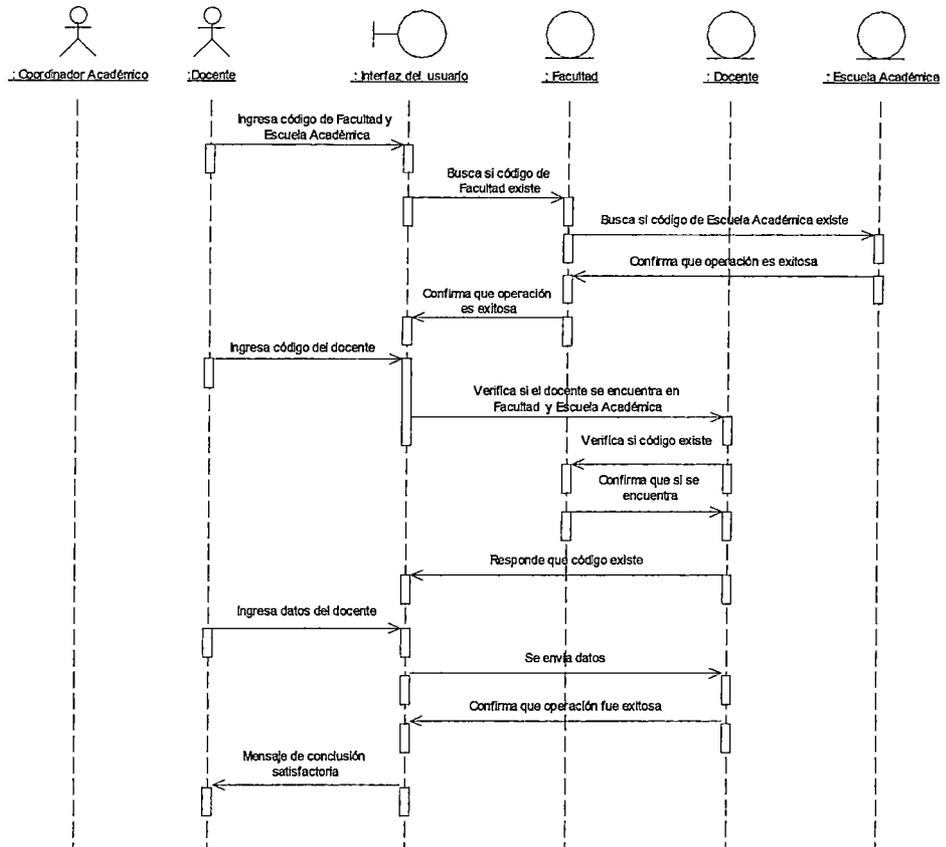
**Figura III.41 Diagrama de secuencia del escenario del caso de uso Consultar solicitud de docente**



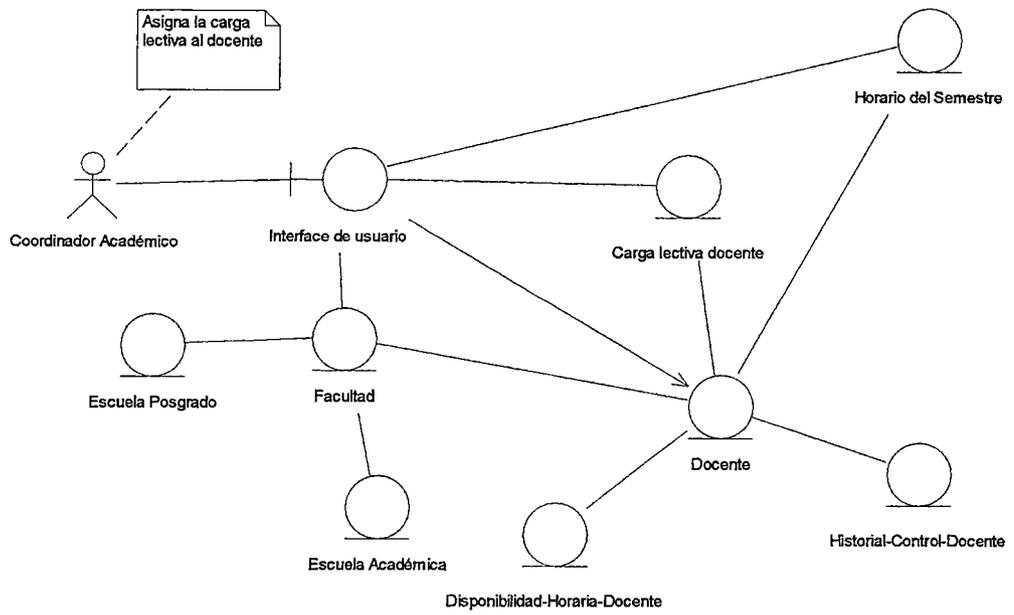
**Figura III.42 Diagrama de clase que muestra las clases que realizan el caso de uso Ingresar datos del docente**



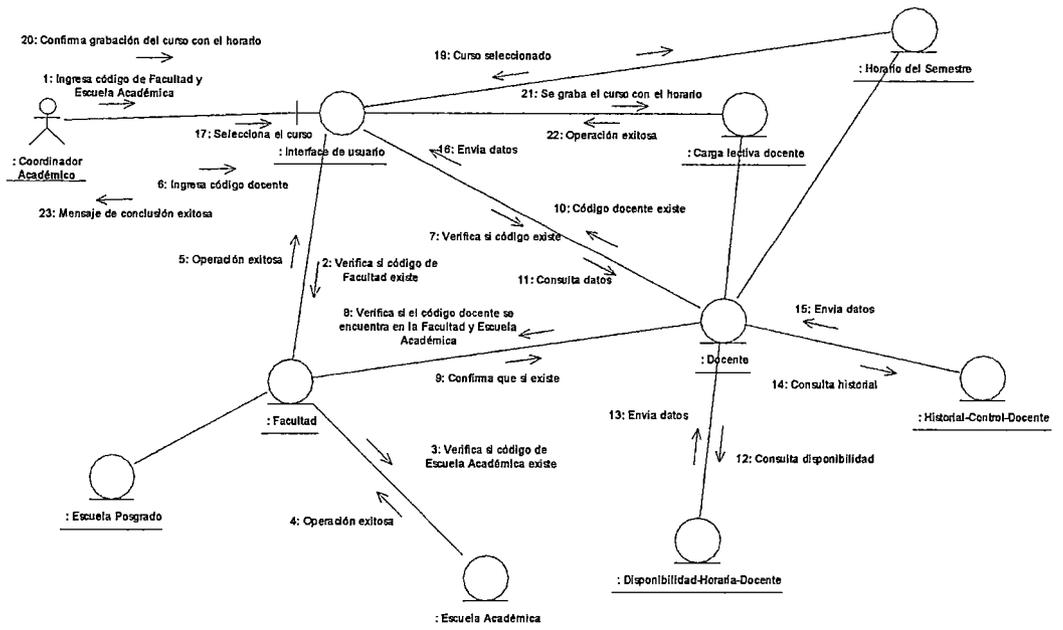
**Figura III.43 Diagrama de colaboración del escenario del caso de uso  
Ingresar datos del docente**



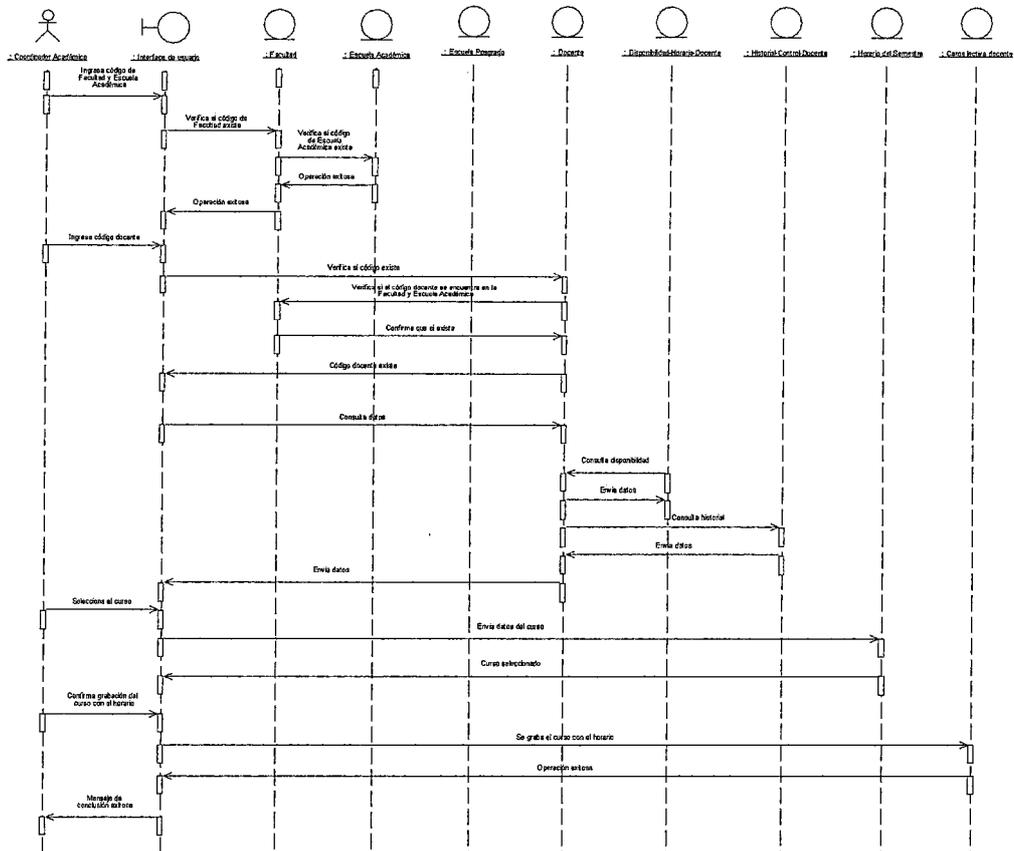
**Figura III.44 Diagrama de secuencia del escenario del caso de uso  
Ingresar datos del docente**



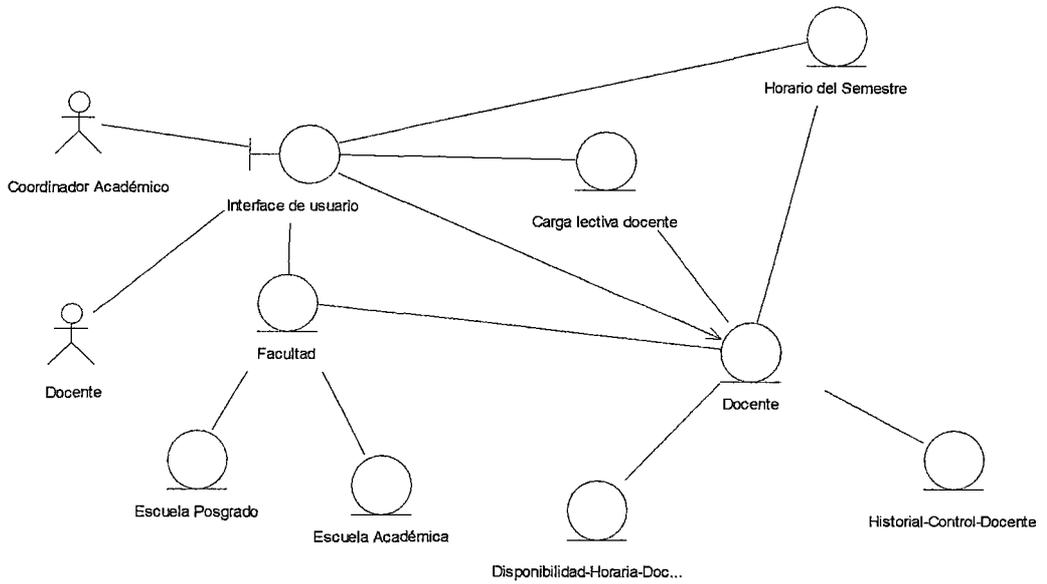
**Figura III.45 Diagrama de clase que muestra las clases que realizan el caso de uso crear carga lectiva del docente**



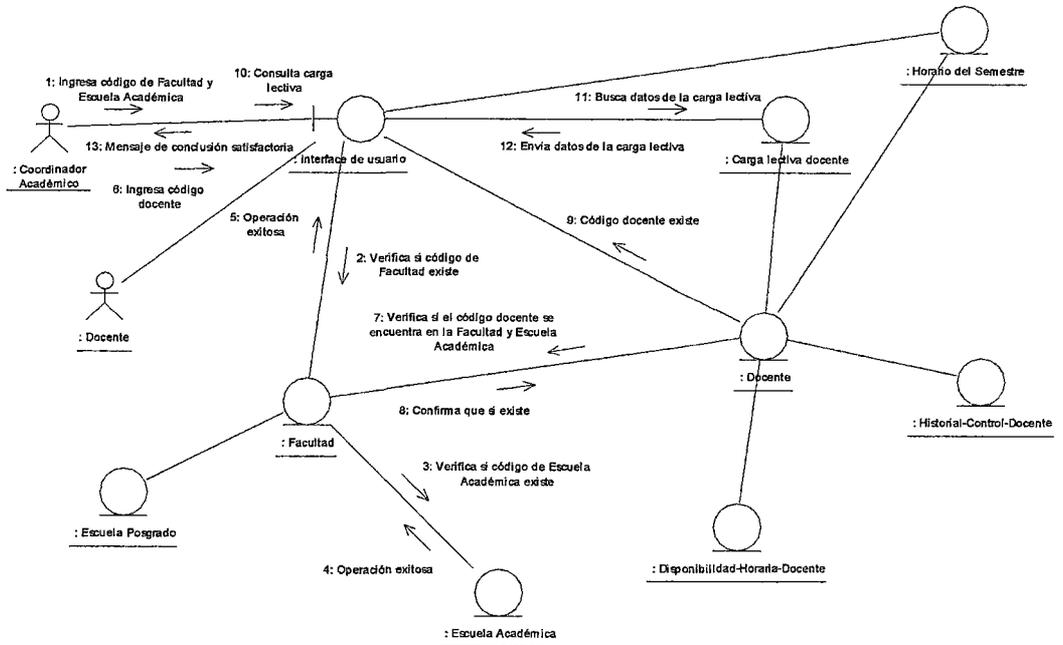
**Figura III.46 Diagrama de colaboración del escenario del caso de uso Crear carga lectiva del docente**



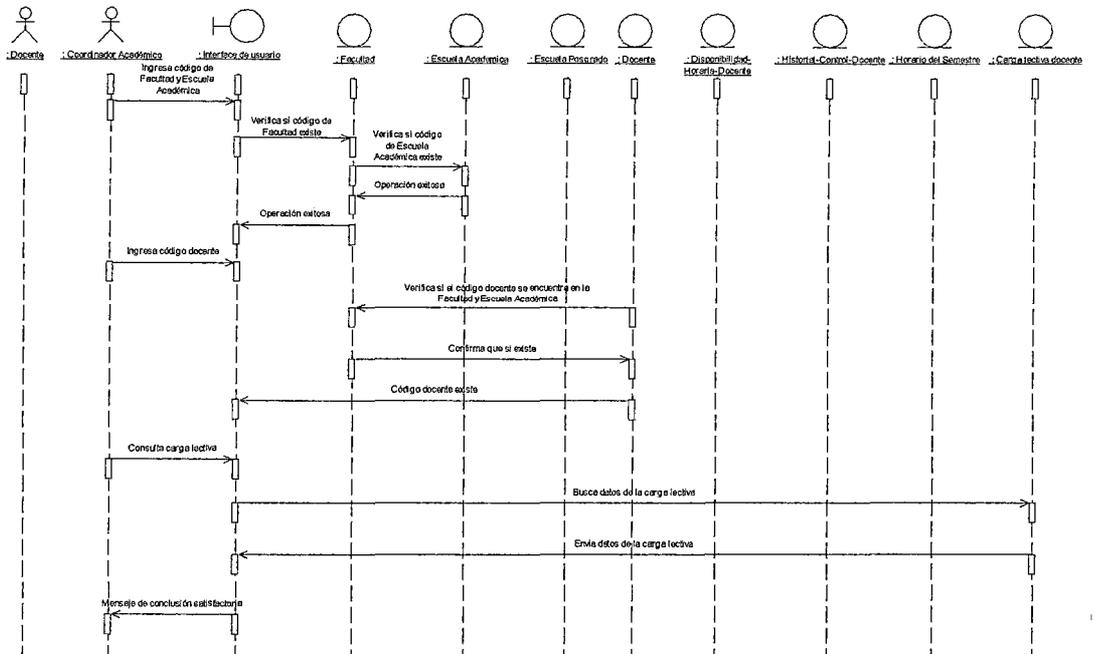
**Figura III.47 Diagrama de secuencia del escenario del caso de uso Crear carga lectiva del docente**



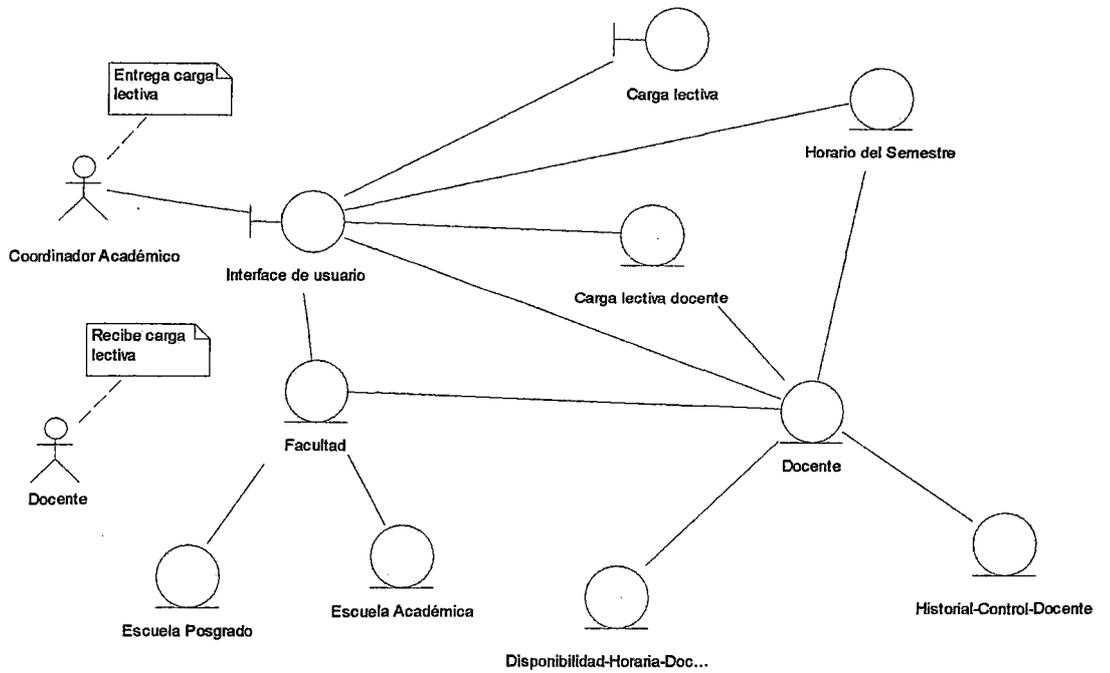
**Figura III.48 Diagrama de clase que muestra las clases que realizan el caso de uso Consultar carga lectiva de docente**



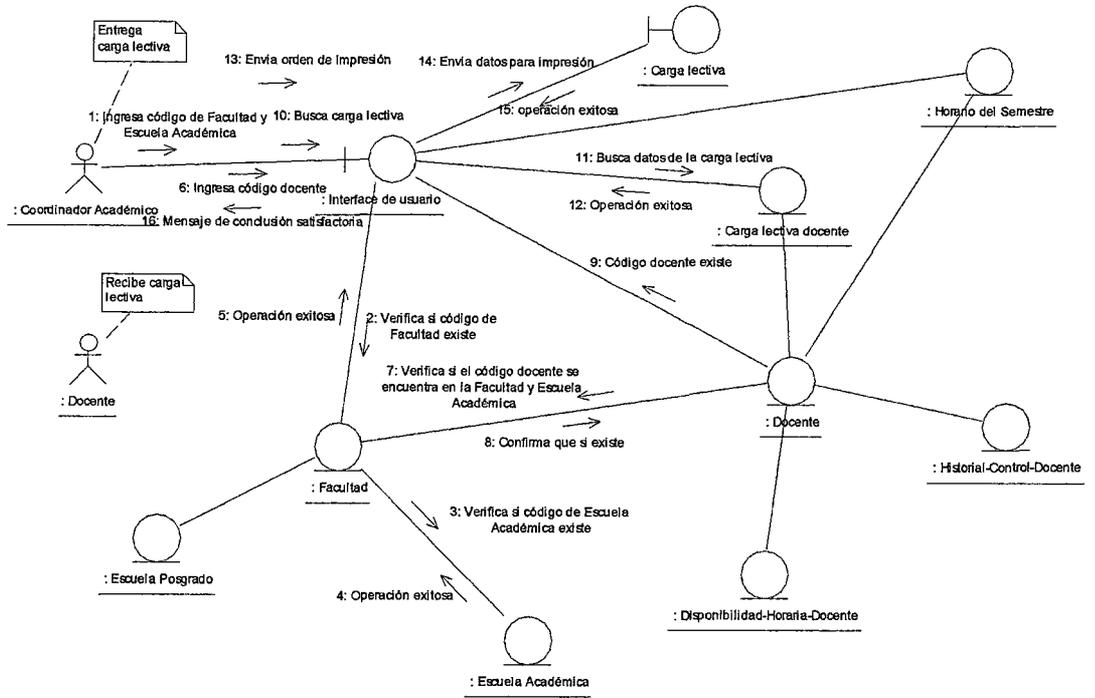
**Figura III.49 Diagrama de colaboración del escenario del caso de uso Consultar carga lectiva de docente**



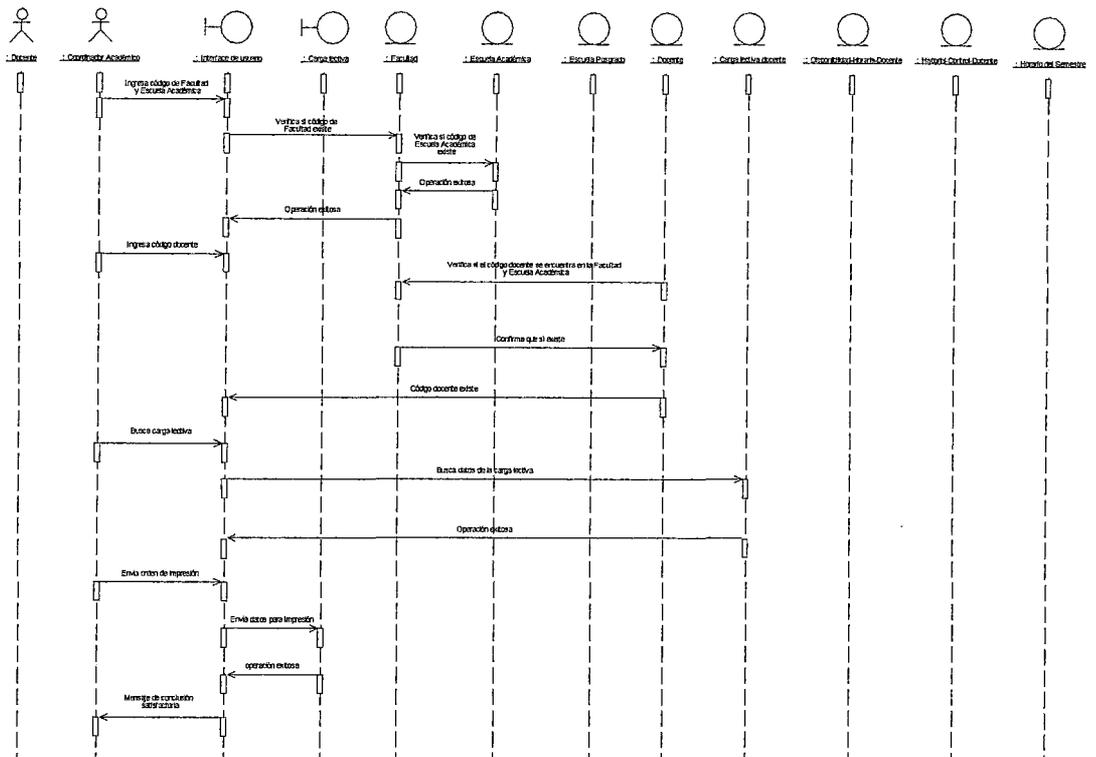
**Figura III.50 Diagrama de secuencia del escenario del caso de uso Consultar carga lectiva de docente**



**Figura III.51 Diagrama de clase que muestra las clases que realizan el caso de uso Emitir, entregar carga lectiva**



**Figura III.52 Diagrama de colaboración del escenario del caso de uso Emitir, entregar carga lectiva**



**Figura III.53 Diagrama de secuencia del escenario del caso de uso Emitir, entregar carga lectiva**

En esta parte del diseño, el diagrama de clases es presentando con sus atributos y métodos respectivamente. Como se observa en la Figura III.53.

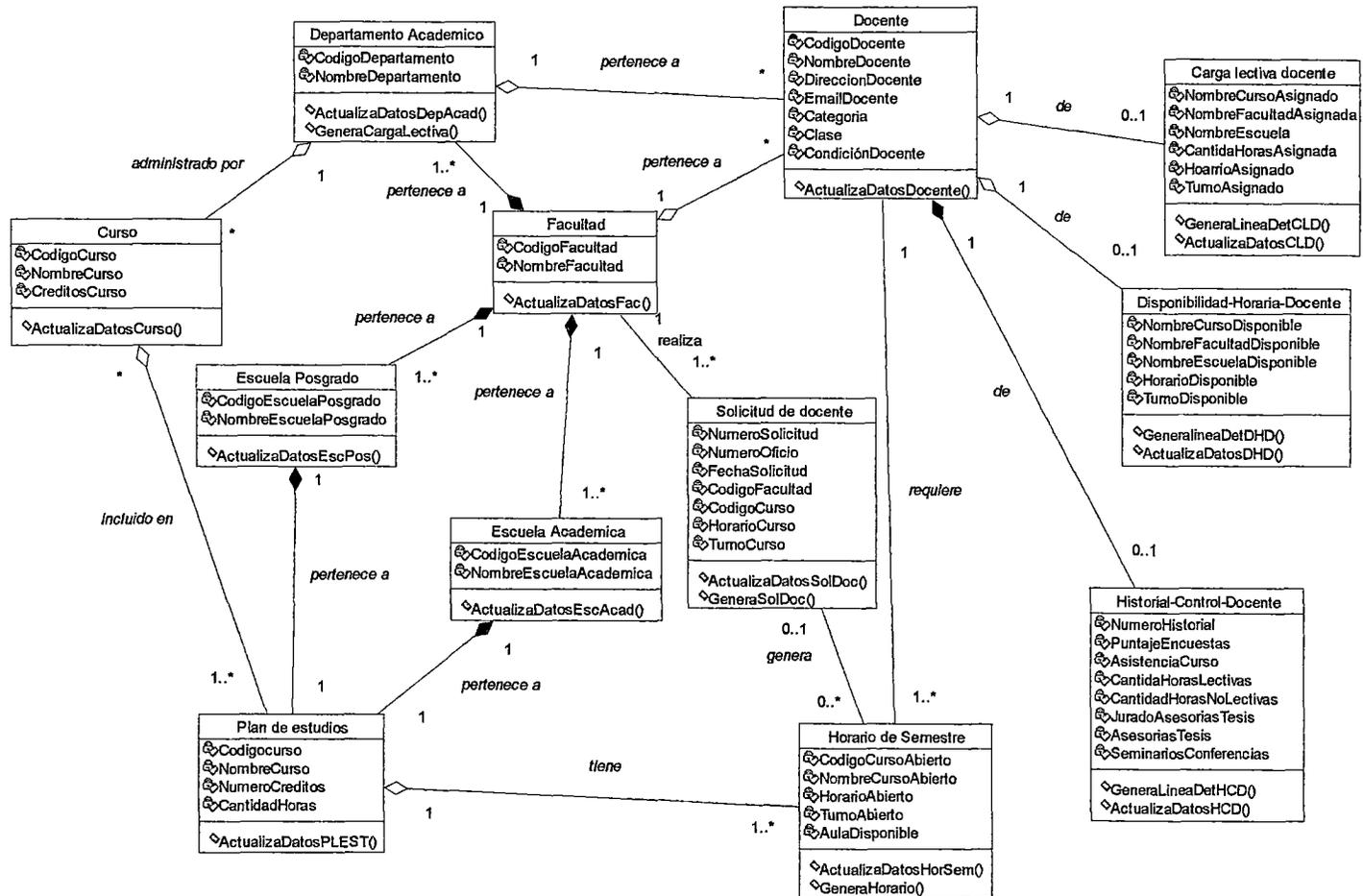
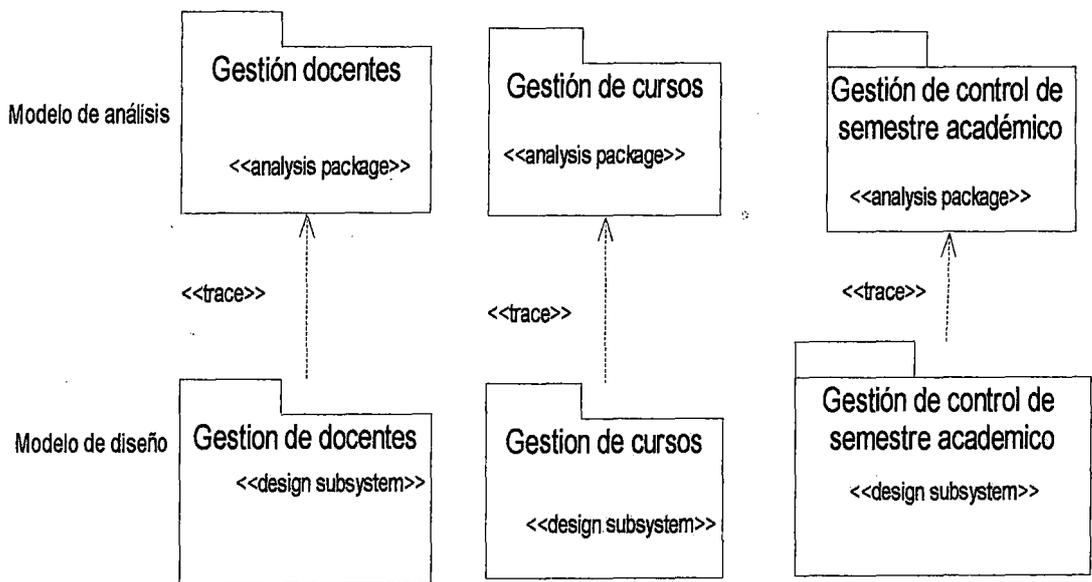


Figura III.54 Diagramas de clases con atributos y métodos.

**3.2.3.4 Diagrama de subsistemas.-** En esta parte identificaremos los subsistemas a partir de los paquetes de análisis existentes, en la Figura III.54 se muestran los subsistemas.



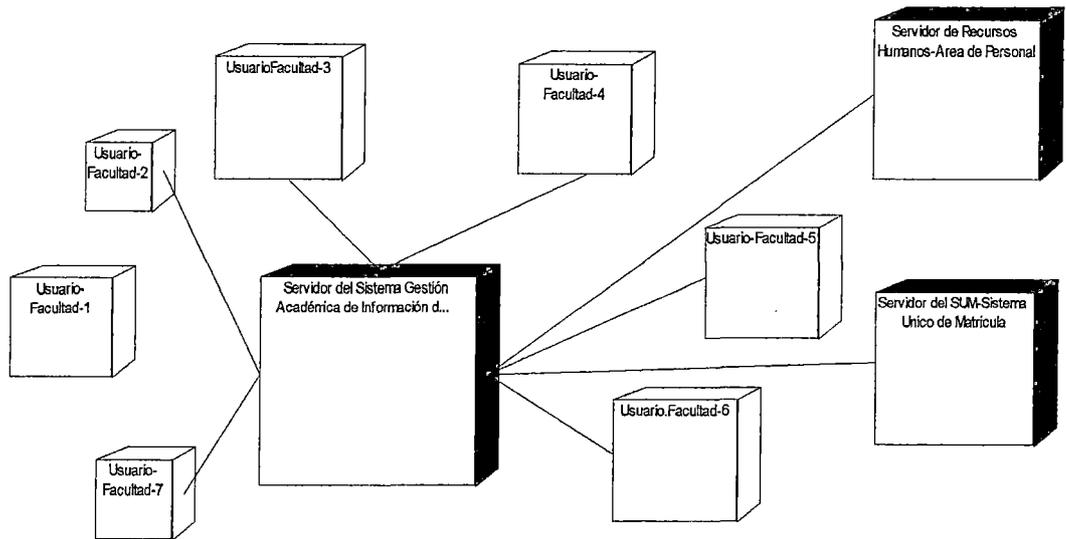
**Figura III.55 Identificación de los subsistemas a partir de los paquetes de análisis**

Siendo el proyecto no muy grande, no requiere el desarrollo de un diagrama de paquetes.

**3.2.3.5 El modelo de despliegue (nodos).**- El sistema de información está diseñado para ser implementado en una plataforma cliente-servidor. La red Cliente-servidor funcionara como una computadora distribuida. La ventaja de una red cliente servidor es que promueve el uso compartido tanto de los programas como de los datos. Los programas del Sistema de Gestión se almacenarán en el servidor, no se requiere de instalar en cada nodo. Los usuarios en este caso las Facultades de la universidad, podrán ejecutar los mismos programas desde sus computadoras personales y pueden utilizar los datos almacenados en el servidor por cualquier otro usuario. Las tablas del Sistema de Personal que contienen datos de los docentes pueden ser exportados del manejador de base de datos del Sistema de Personal por ejemplo del SQL Server al Excel y de Excel al Sql Server del Sistema de Información de Gestión.

Con respecto a los requerimientos de software, el sistema está diseñado para ser implementado en las universidades que se ciñan a la Ley Universitaria, utilizando su propia infraestructura.

En la siguiente Figura III.56 se muestra el Diagrama de Despliegue.

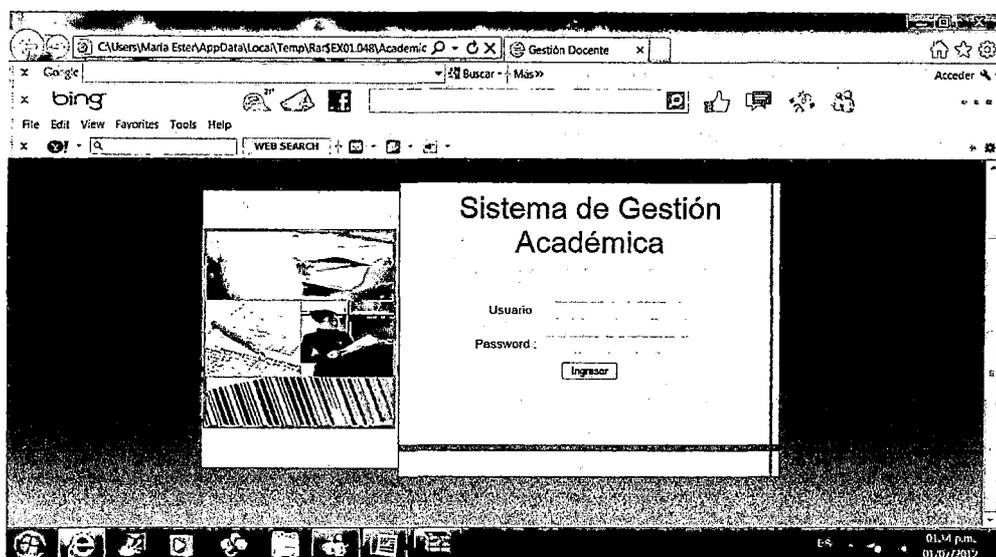


**Figura III.56 Diagrama de Despliegue del Sistema de Información**

### **3.2.4 Flujo de trabajo de implementación.-**

**3.2.4.1 Presentación de un prototipo del sistema de información.-** En esta parte presentaremos un prototipo del Sistema de Gestión de Información Académica para una Universidad.

Como se observará se ingresara el nombre del usuario y el password:



**Figura III.57 Pantalla principal**

El sistema requiere de cuatro usuarios el Docente, el Director de Escuela Académica Profesional, el Director de Postgrado y el Director del Departamento Académico.

El sistema presenta tres módulos principales: Gestión de Docentes, Gestión de Cursos y Gestión de Control Semestre Académico.

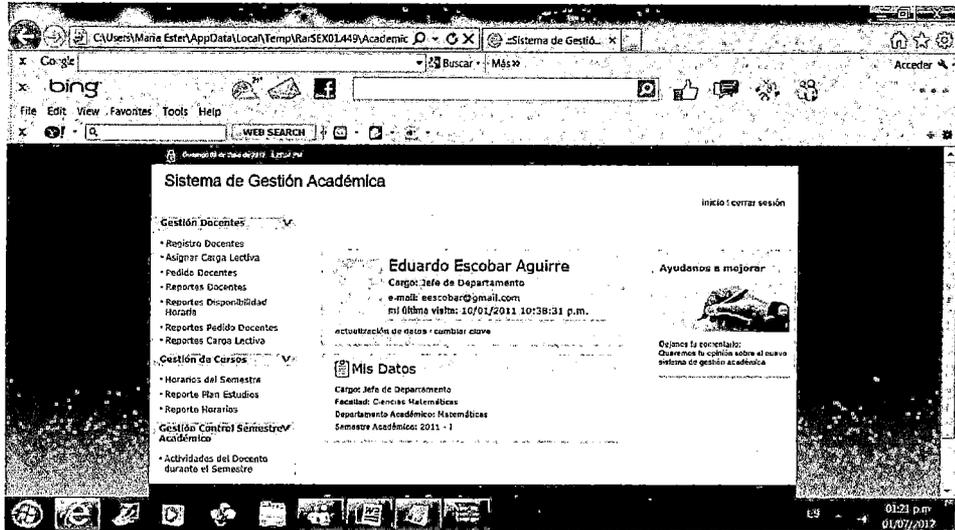


Figura III.58 Sistema de Gestión con los tres módulos

En Gestión Docentes como se observa el Registro de Docentes que pertenecen a la Facultad:

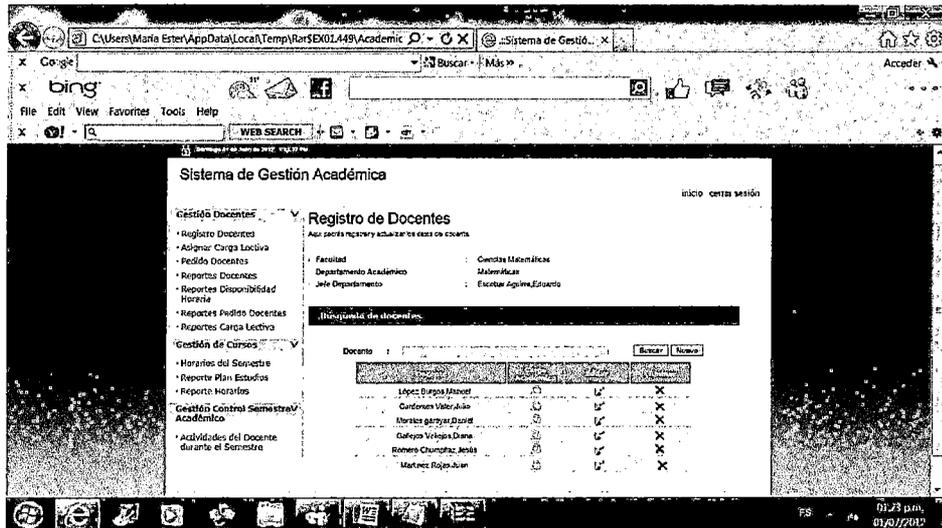


Figura III.59 Registro de Docentes

Registro Pedido de Docentes son requerimientos de docentes que las Facultades realizan cada semestre académico.

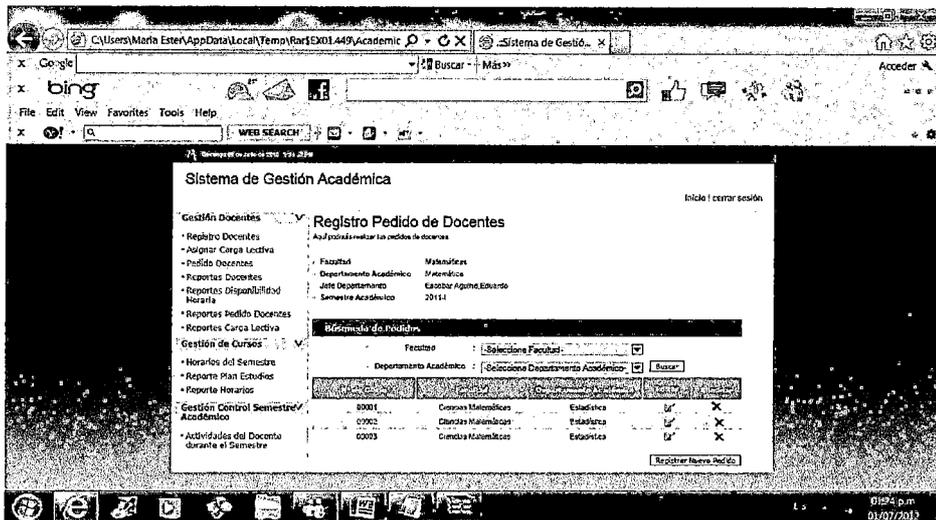


Figura III.60 Registro Pedido de Docente

Se muestra como se ingresa un pedido de docente por ejemplo el número de oficio, el nombre de la Facultad a la que corresponde el pedido.

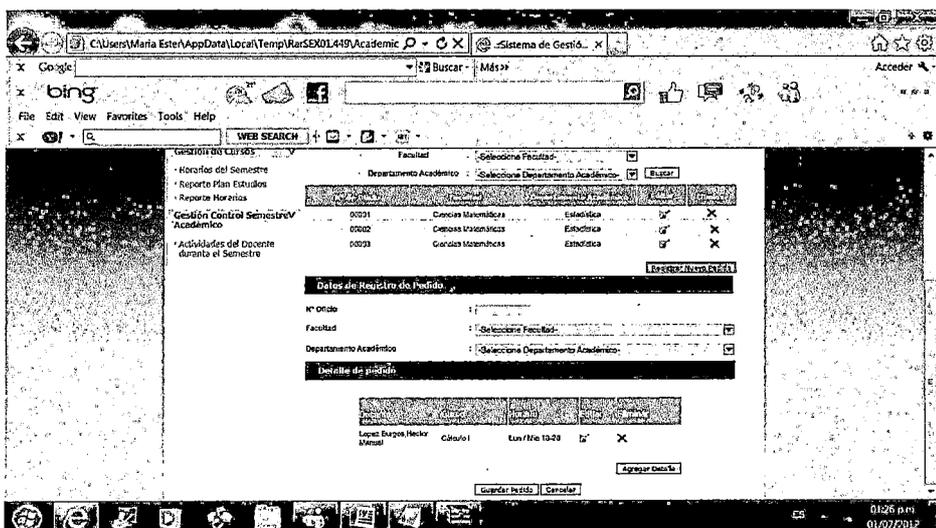


Figura III.61 Ingreso pedido de docente

En Gestión de cursos se tienen los horarios de los cursos del semestre académico:

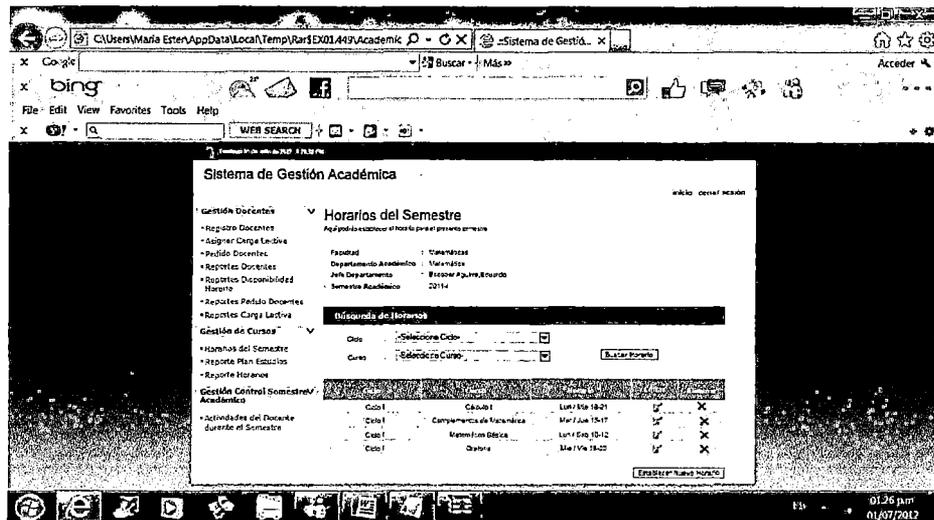


Figura III.62 Horarios del semestre

Se observa como el horario de cursos será creado:

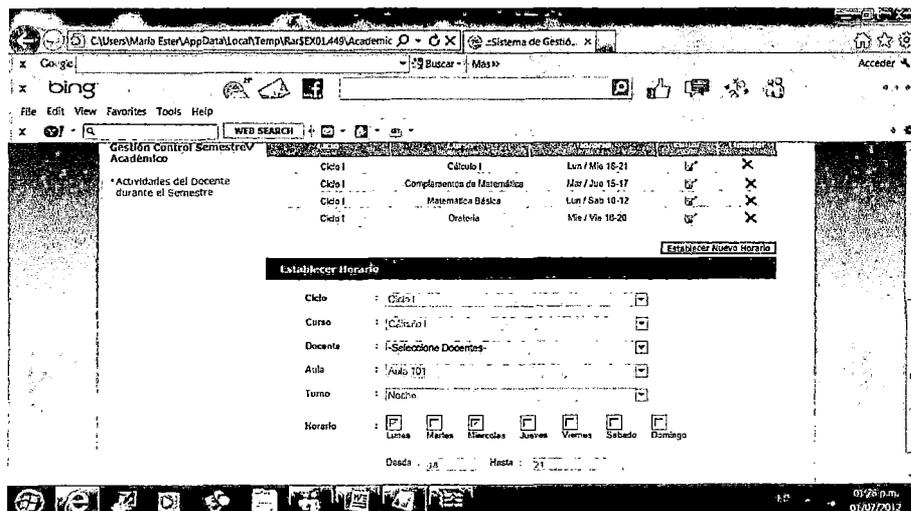


Figura III.63 Horario del curso a crear

Se selecciona un reporte de horario:

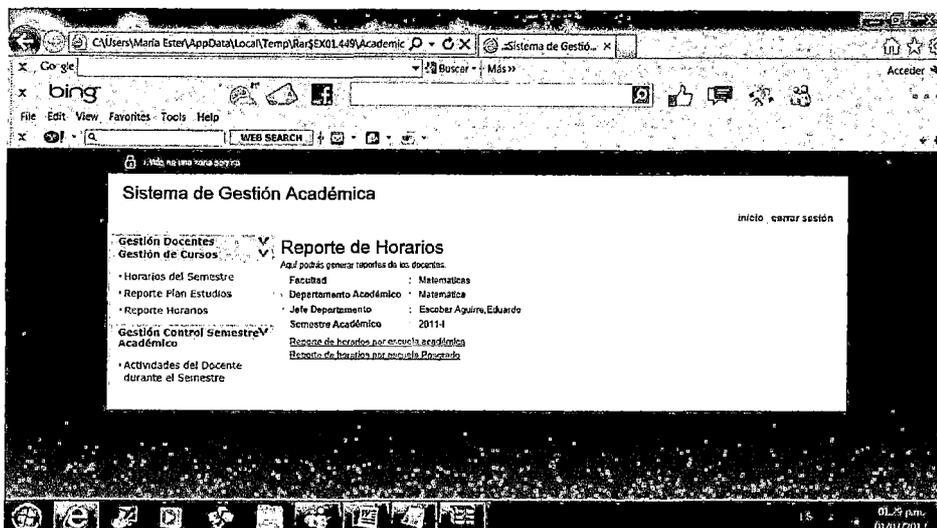


Figura III.64 Reporte de Horarios

El Reporte de Horario por Escuela Académica presenta el siguiente formato:

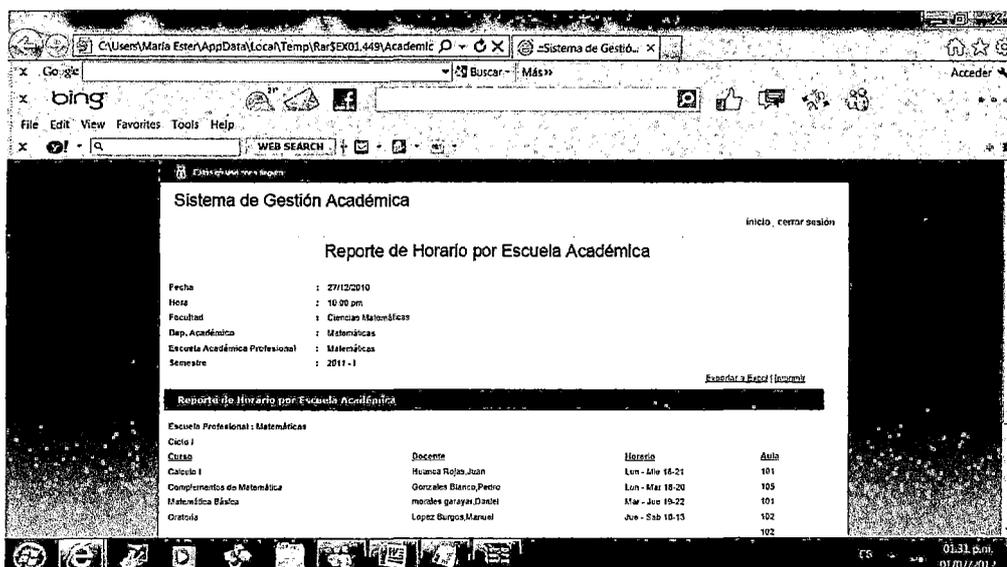


Figura III.65 El Reporte de Horario por Escuela Académica

Los docentes ingresaran su disponibilidad horaria en como se observa el gráfico:

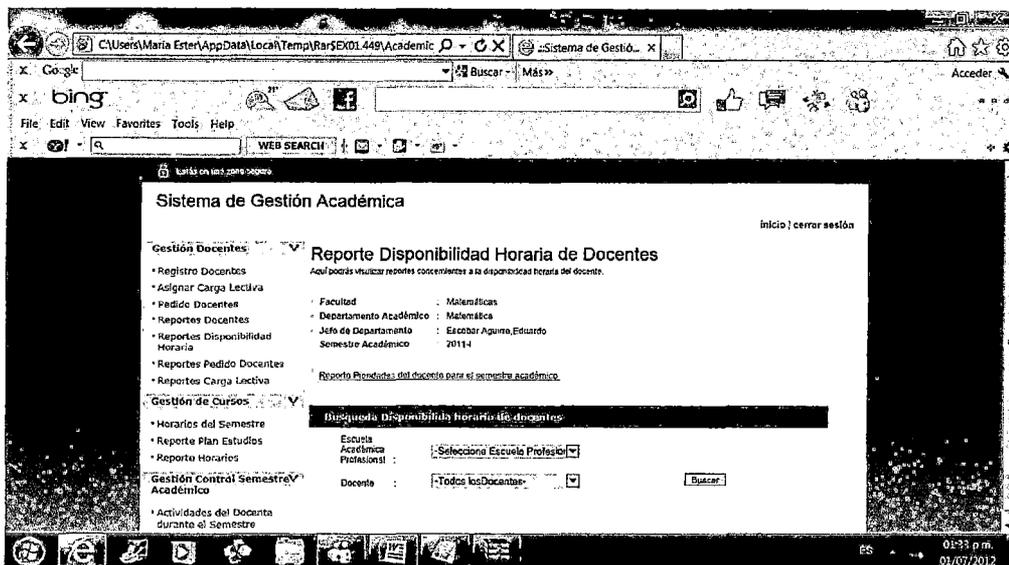


Figura III.66 Reporte de Disponibilidad Horaria de Docentes

Se genera la carga lectiva del docente:

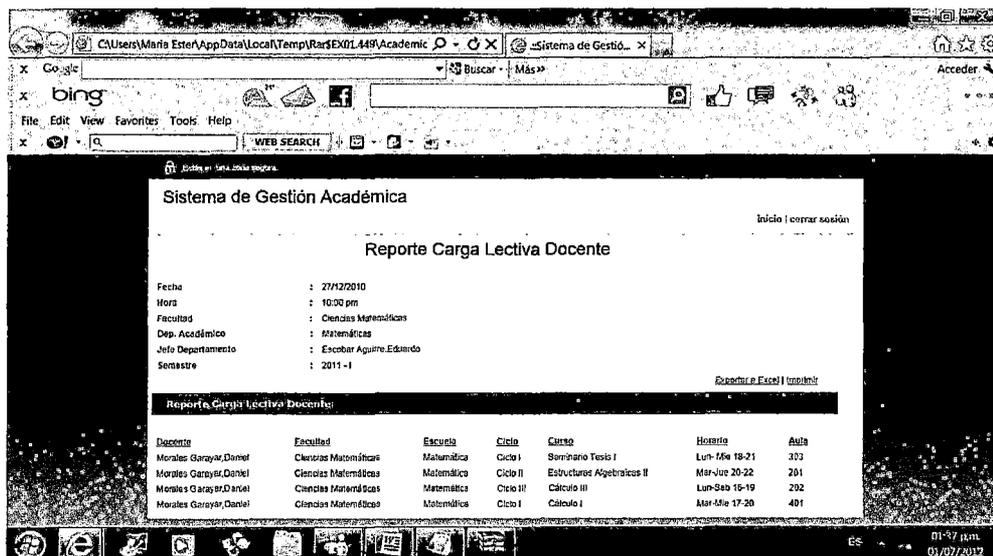


Figura III.67 Reporte de Carga Lectiva

Se presenta como serán ingresados los horarios del semestre académico.

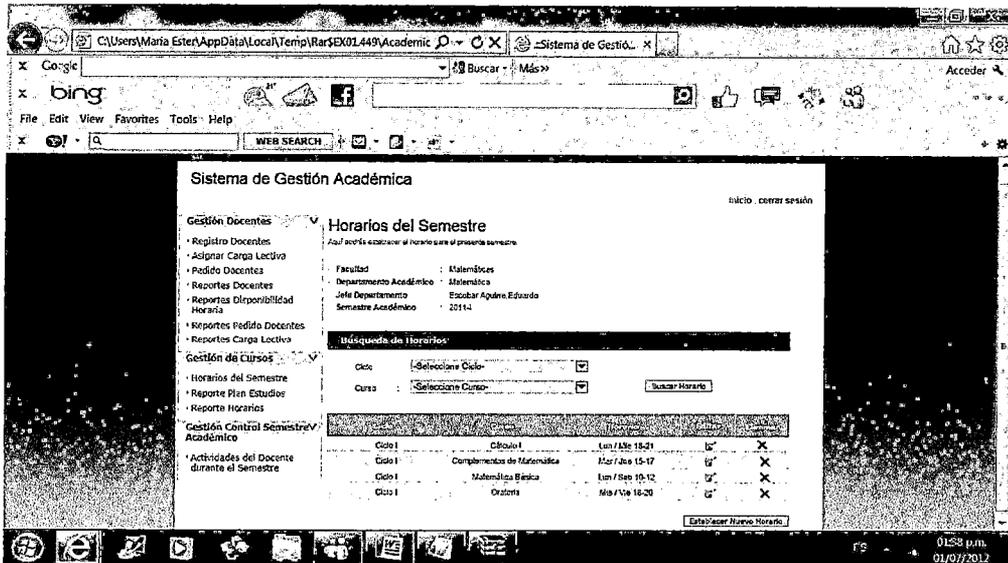


Figura III.68 Horarios del Semestre

Se completa el ingreso datos para el horario de clases:

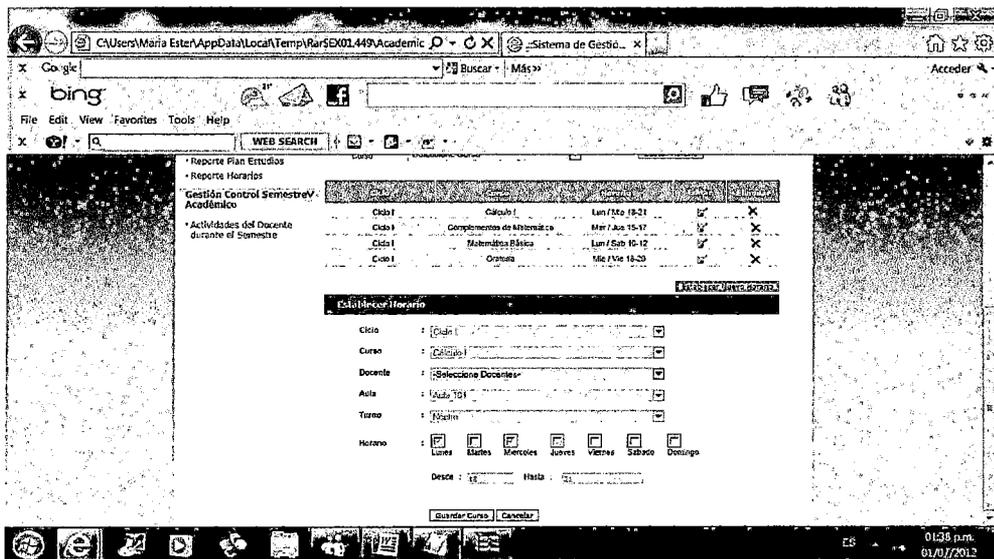
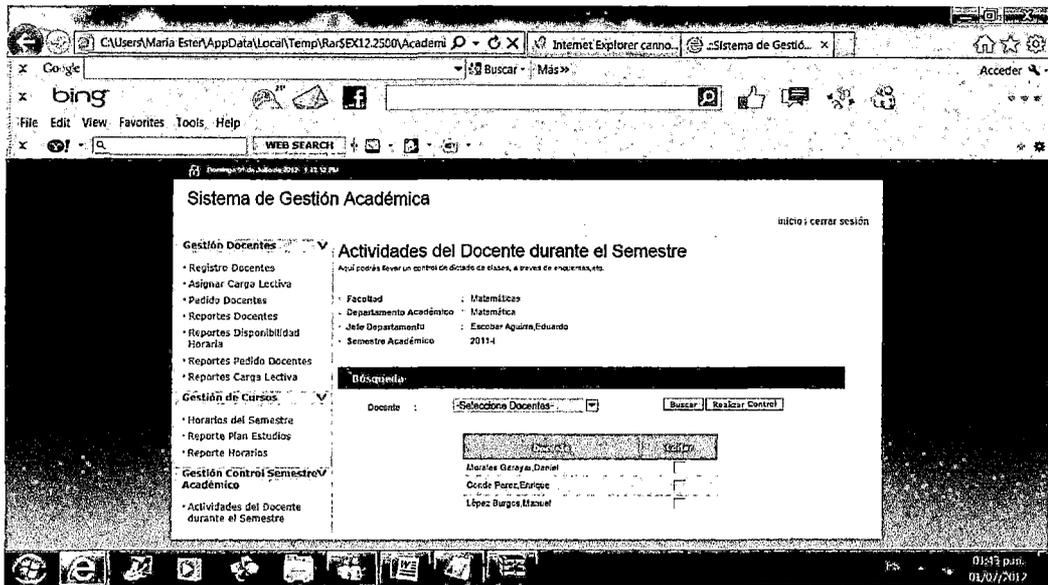


Figura III.69 Ingreso del horario de clases

También se registran las actividades del docente durante el semestre académico.



**Figura III.70 Actividades del Docente durante el Semestre**

Al presionar el botón se muestra la pantalla siguiente, donde se registran las actividades realizadas por el docente, como la asistencia al dictado de clase, entre otros datos.

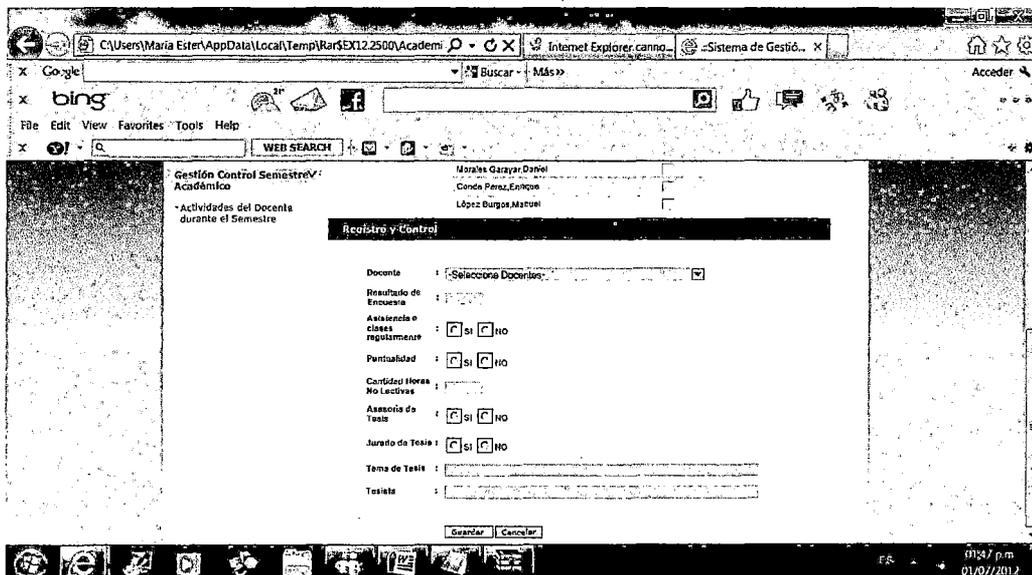


Figura III.71 Registro de actividades del docente

3.2.5 Flujo de trabajo de prueba.- Se harán las pruebas con el prototipo desarrollado.

## **CAPÍTULO IV**

### **ANÁLISIS DE COSTO BENEFICIO**

En este capítulo presentaremos un estudio de factibilidad económica que comprende el análisis de costo beneficio, costo del sistema actual y del sistema propuesto, también se presenta un presupuesto aproximado del desarrollo del sistemas de software utilizando la metodología RUP, tomando como referencia a la Universidad Nacional Mayor de San Marcos, posteriormente se presentan los beneficios tangibles y beneficios no tangibles, para finalmente concluir en base a cálculos realizados, que la implantación del nuevo sistema automatizado brindaría beneficios tangibles y no tangibles; y reducción de costos en salarios de personal y materiales de oficina, como se detallarán en las siguientes secciones.

#### **4.1 Factibilidad económica.-**

Para realizar un análisis de resultados, realizaremos un estudio de factibilidad económica para el desarrollo del nuevo sistema de información.

##### **4.1.1 Análisis de costo-beneficio.-**

Permite hacer una comparación entre los costos del sistema actual (no automatizado en el caso del trabajo de investigación), y los costos que tendría un nuevo sistema, y los beneficios que la Informática brinda.

Para el análisis costo beneficio, tomaremos como referencia a la UNMSM, teniendo en cuenta que la universidad está compuesta por 20 Facultades, y las estimaciones calculadas se basan en esta cantidad, por ejemplo el costo por papel para

apuntes es S/. 7.00 semanales, equivalentes a S/. 28.00, mensuales, que multiplicado por 20 Facultades da como resultado 560.00 soles de consumo mensual de la universidad. .

**Costo del sistema actual**, es decir se hará un cálculo aproximado de los costos que genera al realizar los procesos de manera manual y local durante un año.

##### **Costos de material de escritorio (oficina)**

<b>Material</b>	<b>Costo aproximado</b>	<b>Consumo mensual</b>	<b>Monto en soles por un año</b>
Papel para apuntes millar	7.00	560.00	6,700.00
Papel para impresora millar	30.00	2,400.00	28,000.00

Cartuchos de impresora	100.00	12,000.00	144,000.00
Accesorios de oficina	50.00	4,000.00	48,000.00
<b>Total</b>			226,700.00

### Costos de salario del Personal del Sistema Actual

Recurso humano	Salario mensual en soles por 8 horas	Salario anual en soles por Facultad
Docente Académica	1,000.00	12,000.00
Secretaria	700.00	8,400.00
Asistente	400.00	4,800.00
<b>Total</b>		25,200.00

Multiplicado por las 20 Facultades asciende a S/. 504,000

Así mismo se incurre en Gastos de mantenimiento y accesorios para los equipos de cómputo ascendente a S/.5,000 (anual)

El costo total es :  $226,700.00 + 504,000.00 + 5,000 = S/. 735,700$

**Costo del sistema propuesto**, es decir se hará un cálculo aproximado de los costos que generaría si el sistema estuviera funcionando durante un año, lo significaría una reducción de material del 50% aproximadamente.

### Costos de material de escritorio (oficina)

Material	Costo aproximado	Consumo mensual	Monto en soles por un año
Papel para apuntes	7.00	$560.00/2=280.00$	3,360.00

millar			
Papel para impresora millar	30.00	$2,400.00/2=1,200.00$	14,400.00
Cartuchos de impresora	100.00	$12,000.00/2=6,000.00$	72,000.00
Accesorios de oficina	50.00	$4,000.00/2=2,000.00$	24,000.00
<b>Total</b>			113,760.00

**Costo de hardware y software.-** Como se ha tomado como referencia la UNMSM, que cuenta con los equipos de hardware y software necesarios, lo cual no generaría un costo para la continuación del proyecto en sus dos fases de construcción y transición .

**Costo de Personal.-** Con respecto al personal encargado de llevar a cabo el proyecto de software, se presento un presupuesto, como se observa en el Presupuesto del Proyecto presentado, además siendo un trabajo de investigación para obtener el grado de Maestría, donde ya se ha desarrollado dos fases inicio y elaboración, se considera el 50% por ciento del costo total siendo igual a S/. 52,450.00, teniendo en cuenta que la UNMSM tiene los equipos de hardware y software necesarios para la continuación del proyecto. Se debe indicar que al automatizar los procesos, se reducen y se aligeran las cargas laborales del personal administrativo y académico de la jornada laboral de 8 horas a 4 horas.

## PRESUPUESTO DEL PROYECTO DE SOFTWARE

El presupuesto en base a la planeación del proyecto por 12 meses ( 1 año).

RECURSOS	Costo S/. por mes	Costo S/. por 12 meses
<b>PERSONAL</b>		
• Un Jefe de Proyecto	3,500.00	42,000.00
• Un Analista	2,800.00	33,600.00
• Dos Programadores	1,000.00	28,800.00
<b>EQUIPOS Y MATERIALES DE TRABAJO</b>		
<b>Hardware</b>		
1 servidor	4,500.00	4,500.00
5 computadoras	2,800.00	5,600.00
1 impresora	700.00	700.00
<b>Software</b>		
Rational Rose Versión 7.0	300.00	300.00
Visual .Net	300.00	300.00
Sistema Operativo Windows 7	300.00	300.00
<b>Materiales de trabajo</b>		
Libros y revistas	600.00	600.00
Materiales de escritorio	500.00	500.00
Imprevistos		500.00
	<b>TOTAL</b>	<b>117,700.00</b>

Si la empresa o institución brinda equipo y material de trabajo, se tendría un ahorro de 12,800.00, que restados al costo total 117,700.00 – 12,800.00 = 104,900.00 que sería el costo del proyecto a Tiempo Real . El proyecto se ha planificado para las dos fases inicio y elaboración estimado en seis meses, lo

cual ya ha sido desarrollado, por lo tanto a esta cantidad se le divide entre dos y se obtiene:

$$104,900.00/2=S/.52,450.00$$

Sería el costo por terminar el proyecto que corresponde a las fases de construcción y transición.

#### Costos de salario del Personal del Sistema Propuesto

Recurso humano	Salario mensual de 4 horas	Salario anual
Jefe de Proyecto	3,500	42,000
Analista Programador	2,000	24,000
Docente Académico	1,2000.00/2 x 20 Facultades	120,000.00
Secretaria	7000/2	84,000.00
Asistente	400.00/2	48,000.00
total		318,000.00

El costo total del software del proyecto , sumado los gastos operativos como : costo de material de oficina, mano de obra y adicionalmente los accesorios de los equipos de cómputo (S/. 4,000), necesarios para la gestión y puesta en marcha del proyecto, estimándose al cierre de cada periodo anual ascendería el importe a S/. 553,460

(sumando 117,700 + 113,760 + 318,000 + 4,000 )

El ahorro por tanto sería la diferencia del costo del sistema actual menos el costo del sistema propuesto:  $735,700 - 553,460 = S/. 299,940.00$

Como se observara se tiene ahorro anual, aproximado de S/. 299,940.

Cabe señalar que los beneficios del sistema propuesto están orientados a mejorar el control y velocidad en los procesos del sistema actual, y además con respecto a la inversión del proyecto, es decir el costo también es reducido en un 50% por ciento del total.

#### **Beneficios tangibles:**

- Reducción de costos de material de escritorio u oficina, reducción espacio físico.
- Reducción de horas trabajadas, lo que implica reducir los salarios de empleados.(horas/hombre)
- Optimización de Costos/Beneficio y correlación de costos, ingresos y gastos

#### **Beneficios intangibles:**

- Capacidad de almacenar datos automáticamente, lo que implica datos integrados y la seguridad en el almacenamiento de datos e información.
- Generar información eficiente, confiable, consistente y oportuna que permita tomar decisiones.
- Mayor aprovechamiento de los recursos de hardware y software instalados.
- Mayor rapidez en realizar consultas y actualizaciones de la información.
- Mejor control y seguimiento de las actividades en la administración académica.
- Optimización de tiempos.

**CUADRO COMPARATIVO ANUAL**  
Expresado en Nuevos Soles

CONCEPTOS		COSTO ACTUAL	COSTO PROPUESTO	AHORRO ANUAL	AHORRO MENSUAL
I	<b>COSTO DEL SOFTWARE</b>		<b>117,700</b>		
II	<b>GASTOS OPERATIVOS</b>				
1	MATERIAL DE ESCRITORIO	226,700	113,760	112,940	9,412
2	MANO DE OBRA	504,000	252,000	252,000	21,000
	01 Docente Académico				
	01 Secretaria				
	01 Asistente				
	01 Jefe de Proyecto		42,000	-42,000	-3,500
	01 Analista		24,000	-24,000	-2,000
3	MANTENIMIENTO DE EQUIPO DE COMPUTO	1,000	0	1,000	83
4	ACCESORIOS DE EQUIPOS DE COMPUTO				
	20 facultades x S/. 200.00 anual	4,000	4,000	0	0
	<b>TOTAL GASTOS OPERATIVOS</b>	<b>735,700</b>	<b>435,760</b>	<b>299,940</b>	<b>24,995</b>
III	<b>COSTOS TOTALES</b>	<b>735,700</b>	<b>553,460</b>		

**EVALUACION ECONOMICA DEL PROYECTO DE INVERSION**

COSTO DE INVERSION INICIAL : 117,700  
 INGRESOS ( AHORROS MENSUALES) : 299,940  
 TASA INTERES EFECTIVA ANUAL 5  
 TIEMPO 5 años  
 MONEDA NACIONAL Soles

Costo inicial	→	117,700	299,940	299,940	299,940	299,940	299,940	Ingresos ↓ 299,940
		0	1	2	3	4	5	↑ Años

**Determinacion del Valor Actual Neto (VAN )**

$$\begin{aligned}
 \text{VAN} &= -\text{COSTO INICIAL} + \frac{A}{(1+i)} + \frac{B}{(1+i)^2} + \frac{C}{(1+i)^3} + \frac{D}{(1+i)^4} + \frac{E}{(1+i)^5} \\
 \text{VAN} &= -117,700 + \frac{299,940}{(1+0.05)^1} + \frac{299,940}{(1+0.05)^2} + \frac{299,940}{(1+0.05)^3} + \frac{299,940}{(1+0.05)^4} + \frac{299,940}{(1+0.05)^5} \\
 \text{VAN} &= -117,700 + \frac{299,940}{1.51} + \frac{299,940}{2.25} + \frac{299,940}{3.375} + \frac{299,940}{5.0625} + \frac{299,940}{7.59375} \\
 \text{VAN} &= \text{S/},401,859
 \end{aligned}$$

**Determinacion de la Tasa interna de retorno (TIR)**

COSTO DE INVERSION INICIAL : 117,700  
 INGRESOS ( AHORROS MENSUALES) : 299,940  
 TASA INTERES EFECTIVA ANUAL 5  
 TIEMPO 5 años  
 MONEDA NACIONAL Soles

Periodos	año 1	año 2	año 3	año 4	año 5
Flujo de Caja Neto (Ahorro)	299,940	299,940	299,940	299,940	299,940

Para hallar la Tasa Interna de Retorno (TIR) hacemos uso de la fórmula del VAN, sólo que en vez de hallar el VAN (el cual reemplazamos por 0), estaríamos hallando la tasa de descuento:

$$0 = 299,940 / (1 + i)^1 + 299,940 / (1 + i)^2 + 299,940 / (1 + i)^3 + 299,940 / (1 + i)^4 + 299,940 / (1 + i)^5 - 117,7000$$

TIR = 254.38%

Se ha evaluado teniendo como métodos de Evaluación Económica la Tasa Interna de Retorno, representando la tasa de rentabilidad, para este proyecto, de 254.38 %, el mismo que resulta mayor y favorable para el rendimiento a largo plazo; teniendo en cuenta el costo de oportunidad del sistema financiero, a la fecha, representa una tasa de interés efectiva anual de 5% para Cuenta de Ahorros y depósitos a largo plazo en M/N.

Con respecto a los valores obtenidos en unidades monetarias de nuestro país (Soles ), aplicando el costo inicial de este proyecto de software, representado por S/. 117,700, más la corriente de beneficios del flujo de efectivo (que para este caso, vendría ser Ahorros) a lo largo del periodo, adecuadamente ajustado o descontado en cuanto a tiempo, podemos deducir del análisis respectivo según detalle, aplicando la misma tasa de interés del ente financiero del 5%, resulta el Valor Actual Neto de S/. 401.859, resultado favorable, por cuando estamos aplicando esta tasa que resulta menor a la Tasa Interna de Retorno.

## **CAPÍTULO V**

### **ANÁLISIS DE RESULTADOS**

En el este capítulo se presentan los resultados obtenidos al culminar el trabajo de investigación: la encuesta realizada en las universidades, el análisis y diseño de un sistema de gestión para una universidad, el desarrollo del prototipo del sistema de información, el análisis costo beneficio del sistema de información, aplicación de la metodología RUP y finalmente la evaluación crítica y mejoras de la metodología RUP.

Al culminar el trabajo de investigación se realizó un análisis de resultados que a continuación presentaremos:

1. Se realizó una encuesta a nivel nacional de las universidades en nuestro país con la finalidad de conocer si cuentan con sistema de gestión académica. Actualmente existen 102 universidades entre nacionales y públicas y para la encuesta se tomó una muestra de 25 universidades (públicas y privadas) 10 en Lima y 15 en provincia dando como resultado que 13 de ellas, es decir el 52%, no cuentan con un sistema de gestión académica. La investigación realizada permitió tener un conocimiento del estado en que se encuentran las universidades con respecto a las actividades académicas, pudiendo ser desarrolladas con la ayuda de un sistema de gestión académica. En la sección 1.1 se ha tratado sobre la encuesta realizada en la universidad y los resultados obtenidos respectivamente.
2. Se ha completado exitosamente el análisis y diseño de un sistema de gestión académica para una universidad pública o privada con una estructura orgánica de acuerdo a la Ley Universitaria 23733 utilizando la metodología RUP. Se desarrolló las fases de inicio y elaboración con sus respectivos entregables. De esta forma el diseño ha cubierto todos y cada uno de los casos de uso presentados en la tesis. Es importante resaltar que, a partir de los entregables de la tesis, una universidad interesada podría continuar con las demás fases de la metodología RUP: construcción y transición y obtener el sistema completo.
3. Se desarrolló un prototipo del sistema de gestión académica en el lenguaje de programación HTLM de acuerdo con el análisis y diseño presentado en la tesis, lo que permitió observar si el sistema de gestión cumplía con los requerimientos funcionales, si las interfaces eran amigables.

bles, fáciles de aprender y navegar para los usuarios, permitiendo tener una idea general de cómo funcionará el sistema cuando se encuentre terminado.

4. Se realizó un análisis costo beneficio del sistema de información tomando como referencia a la Universidad Nacional Mayor de San Marcos y se comprobó un ahorro de S/. 226,700.
5. En el análisis y diseño del sistema de gestión académica desarrollado se aplicó la metodología RUP, siendo una de las características principales "las iteraciones" que permitió afinar los requerimientos del sistema en la parte de análisis del sistema y rediseñar los diagramas en el diseño del sistema, siendo una ventaja en comparación del método estructurado . También debemos señalar que la investigación de la tesis ha permitido observar que las instituciones actualmente utilizan una mixtura del el método estructurado y la metodología RUP y varias empresas están usando las metodologías ágiles.
6. Se realizó una evaluación crítica y mejoras de la metodología RUP, con la finalidad de lograr estandarizar las notaciones, y mejorar el diagrama de clases de tal manera que sea más descriptivo y fácil de entender.

## **CONCLUSIONES Y RECOMENDACIONES**

### **CONCLUSIONES**

1. Mediante una encuesta realizada a nivel nacional tomando una muestra de 25 universidades se encontró que el 58% de las universidades no cuenta con un sistema de información para la gestión académica.
2. Con la misma encuesta se ha verificado que el 88% de las universidades se ciñe a la estructura establecida por la Ley Universitaria del Perú 23733.
3. El análisis y diseño del sistema de información realizado en el presente trabajo se basa en la estructura organizacional definida por la Ley Universitaria 23733, pero es independiente de la plataforma informática de la organización, por lo tanto puede ser implementado en cualquier universidad con estructura organizacional dictada por la referida ley.
4. El prototipo desarrollado es de utilidad en el desarrollo de sistemas de información tanto para el desarrollador de sistemas como para el usuario, porque permite comprobar si los casos de uso están siendo cubiertos por el sistema, por otro lado permite comprobar si el usuario o el actor puede ejecutar el caso de uso que le corresponde.

5. Durante el análisis y diseño del sistema de información realizado se comprobó que la metodología utilizada es muy flexible a los cambios o modificaciones que se puedan presentar durante el desarrollo del sistema, porque permite en cada fase retroceder a las actividades anteriores del desarrollo del sistema, siendo una ventaja frente a la metodología tradicional.
6. El Proceso Unificado es un modelo complejo con mucha terminología propia, pensado principalmente para el desarrollo de grandes proyectos. Es un proceso que puede adaptarse y extenderse en función de las necesidades de cada empresa.
7. Al haberse revisado diferentes textos se ha observado que el proceso unificado es una metodología que en la actualidad no se está estandarizada; especialmente con respecto a diagramas y artefactos no es muy estricta. Se podría decir que existen más de una versión.

## RECOMENDACIONES

1. De acuerdo a la encuesta realizada el 58% de las universidades no cuenta con un sistema de gestión académica. Sería muy conveniente para estas universidades implementar el sistema de información de gestión académica. Para ello podrían tomar como punto de partida el análisis y diseño realizado en el presente trabajo y continuar con las fases de construcción y transición de la metodología RUP.
2. El proceso unificado es recomendado para sistemas grandes, porque es una metodología rigurosa en sus cuatro fases inicio, elaboración, construcción y transición, implica planificar y realizar las iteraciones, lo que requiere de contar con recursos económicos y recursos humanos. En el caso de sistemas pequeños, medianos no sería rentable.
3. Es necesario el uso del prototipo durante las fases del desarrollo del sistema de información, porque permite comprobar si los casos de uso están cumpliendo con los requerimientos del sistema de información.
4. Para el desarrollo del sistema de información, se debería seleccionar la metodología de acuerdo a sus necesidades, tiempo y presupuesto de la organización.
5. Cada organización debería simplificar de acuerdo a sus propias necesidades y recursos la cantidad de artefactos en la metodología RUP, para evitar su complejidad en los casos en que resulte innecesario.

## GLOSARIO DE TÉRMINOS

1. **Artefacto.-** Es el elemento entregable producido durante las diferentes fases del desarrollo de un sistema.
2. **Carga lectiva.-** Información del docente referente a los cursos que dictará y horarios en un semestre académico de acuerdo a su disponibilidad horaria.
3. **Casos de uso.-** Es la representación grafica de los requerimientos del sistema en estudio.
4. **Diagrama.-** Representación gráfica de un artefacto en el análisis y diseño de sistemas de información.
5. **Gestión.-** Administración de un proceso o actividad en una organización.
6. **Objeto.-** Es cualquier cosa, real o abstracta del cual se almacenan datos y los métodos que controlan dichos datos.
7. **Prototipo.-** Es un entregable en el desarrollo del sistema de información. Consiste de un software que muestra el funcionamiento del sistema de información a nivel de consultas y reportes, con la finalidad de presentar a los usuarios y observar si los casos de uso están siendo cubiertos por el sistema de información.

8. **Requerimiento.-** Es una necesidad del sistema de información para su funcionamiento.
9. **Rational Rose.-** Herramienta de software utilizada en el desarrollo del sistema de información para representar los artefactos.
10. **Sistema.-** Conjunto de partes relacionadas entre sí para un objetivo común.

## BIBLIOGRAFÍA

1. Arlow, Jim, Neustadt. (2006). Programación UML 2. España: Anaya.
2. Booch, G., Rumbaugh, J., y Jacobson, I. (2005). El Lenguaje Unificado de Modelado. España: Addison Wesley.
3. Booch, G., Rumbaugh, J., y Jacobson, I. (1999). El Proceso Unificado de Desarrollo de Software. España: Addison Wesley.
4. Carranza, Zalatiel. (2006). Análisis de Sistemas de Software. Perú:Fondo Editorial. Perú.
5. Davis, G. B. y Olson, M. M. (1987). Sistemas de Información Gerencial. (2da. Ed.).Colombia: McGraw-Hill.
6. Schach, S. R. (2005). Análisis y Diseño Orientado con UML y el Proceso Unificado. México: McGraw-Hill.
7. Senn, James A. (1987). Análisis y Diseño de Sistemas de Información. Mexico: McGraw-Hill.
8. Silva, S., Rodríguez, D., y Luque, José. (2007).Legislación Universitaria Antecedentes, concordancias, adiciones, modificaciones, derogaciones y notas. Perú: Fondo Editorial de la UNMSM.

9. Larman, C. (1999). UML y Patrones – Introducción al Análisis y Diseño Orientado a Objetos. México: Pearson.
  
10. Martín, J., y Odel, James J. (1992). Análisis y Diseño Orientado a Objetos. México: Prentice Hall.

**ANEXOS**

## ANEXO 01

### ENCUESTA A LAS UNIVERSIDADES DEL PERÚ PÚBLICAS Y PRIVADAS EN LIMA Y PROVINCIAS

#### Cuestionario:

1. Nombre de la Universidad:

.....  
.....

2. La Universidad se encuentra ubicada en (Marcar con X según corresponda).

a) Lima                       b) Provincia

Nombre de la provincia

3. La Universidad es :

a) Pública                       b) Privada

4. La Universidad tiene estructura orgánica según la Ley Universitaria 23733: está compuesta por facultades, departamentos académicos, escuelas académicas profesionales y escuelas de posgrado (ver organigrama de pág. 2 y 3).

a) Si                       b) No  En este caso indicar la estructura con organigrama

explicado.

5. La Universidad tiene un Sistema Automatizado de Gestión Académica.

a) Si                       b) No

6. Si tiene un Sistema Automatizado de Gestión Académica, marque con X las funciones que realiza:

a) Registro de datos de docentes y cursos en cada Facultad.

b) Registro de disponibilidad horaria del docente en cada Facultad.

c) Designación de carga lectiva del semestre académico a iniciar.

e) Requerimiento de docentes entre facultades a través de los departamentos académicos.

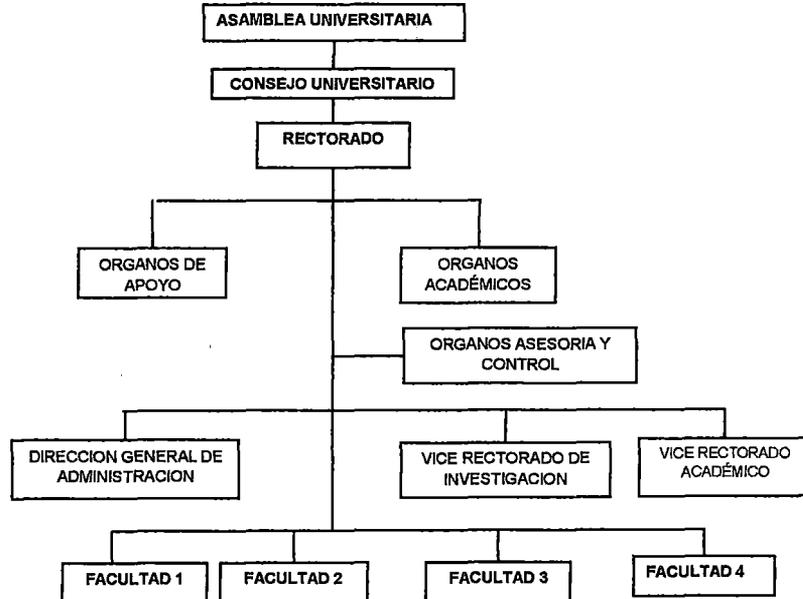
f) Control de dictado de cursos e información del historial de los docentes(asistencia, encuestas, grados, títulos obtenidos, participación en seminarios, congresos).

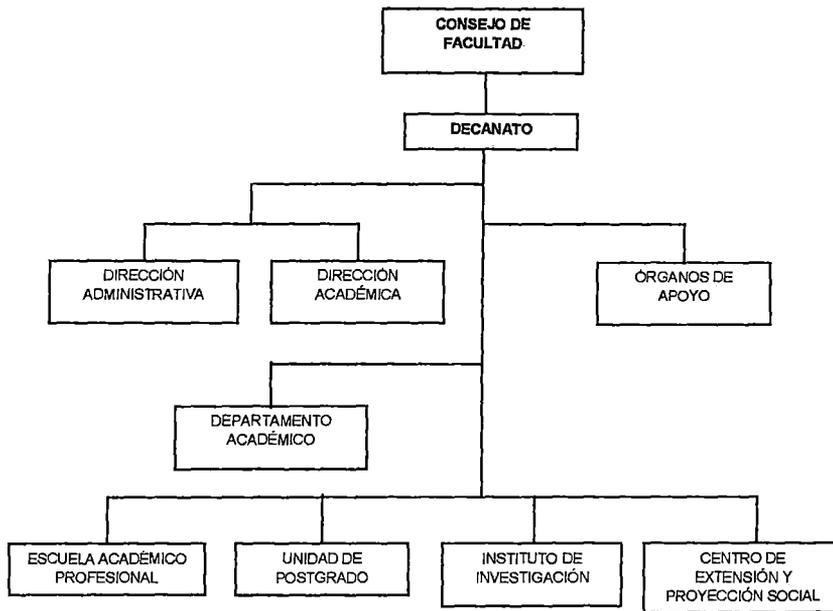
g) Otros (Indique las funciones que no han sido especificadas).

.....  
.....  
.....  
.....

7. ¿Está la universidad interesada en un Sistema Automatizado de Gestión Académica? Si la universidad estuviera interesada en contar con tal sistema, se pondría a su disposición el Análisis y Diseño de dicho sistema, efectuados como parte de una investigación académica.

a) Si       b) No





**Organigrama básico elaborado de una Universidad Pública o Privada de acuerdo con la Ley Universitaria del Perú N° 23733**

## **ANEXO 02**

### **ESTRUCTURA ORGANICA DE LA UNMSM :**

#### **A. ÓRGANOS DE GOBIERNO**

- a. La Asamblea Universitaria
- b. El Consejo Universitario
- c. El consejo de cada Facultad

#### **B. ÓRGANOS DE ALTA DIRECCIÓN**

- a. Rectorado
- b. Vicerrectorados

#### **C. ÓRGANOS DEPENDIENTES DEL RECTORADO**

##### **a. ÓRGANOS DE CONTROL**

- a.1 Oficina General de Auditoria

##### **b. ÓRGANOS DE ASESORAMIENTO**

- b.1 Oficina General de Planificación
- b.2 Oficina de Asesoría Legal
- b.3 Oficina Central de Calidad Académica y Acreditación

##### **c. ORGANOS DE APOYO**

- c.1 Secretaria General
- c.2 Oficina General de Relaciones Públicas
- c.3 Oficina general de Cooperación relaciones Interinstitucionales
- c.4 Oficina Central de Admisión

##### **d. OTROS ÓRGANOS ACADEMICOS**

- d.1 Escuela de Postgrado
- d.2 Oficina Técnica del Estudiante

- d.3 Centro Cultural
- d.4 Instituto "Raúl Porras Barrenechea"
- d.5 Seminario de Historia Rural Andina
- d.6 museo de la Historia Natural
- d.7 Teledusm
- d.8 Jardines Botánicos

## **E. ÓRGANOS DEPENDIENTES DEL VICERRECTORADO**

### **ADMINISTRATIVO**

- a. Oficina General de Servicios Generales, Operaciones y Mantenimiento.
- b. Oficina General de Bienestar Universitario.
- c. Oficina General de Economía.
- d. Oficina General de Recursos Humanos (Personal).
- e. Oficina General de Infraestructura Universitaria.

## **F. ORGANOS DEPENDIENTES DEL VICERRECTORADO DE INVESTIGACIÓN**

- a. Consejo Superior de Investigaciones.
- b. Consejo de Transferencia e Innovación(CTI).
- c. Consejo de Gestión de Investigación (CGI).

## **D. ÓRGANOS DEPENDIENTES DEL VIRRECTORADO ACADÉMICO**

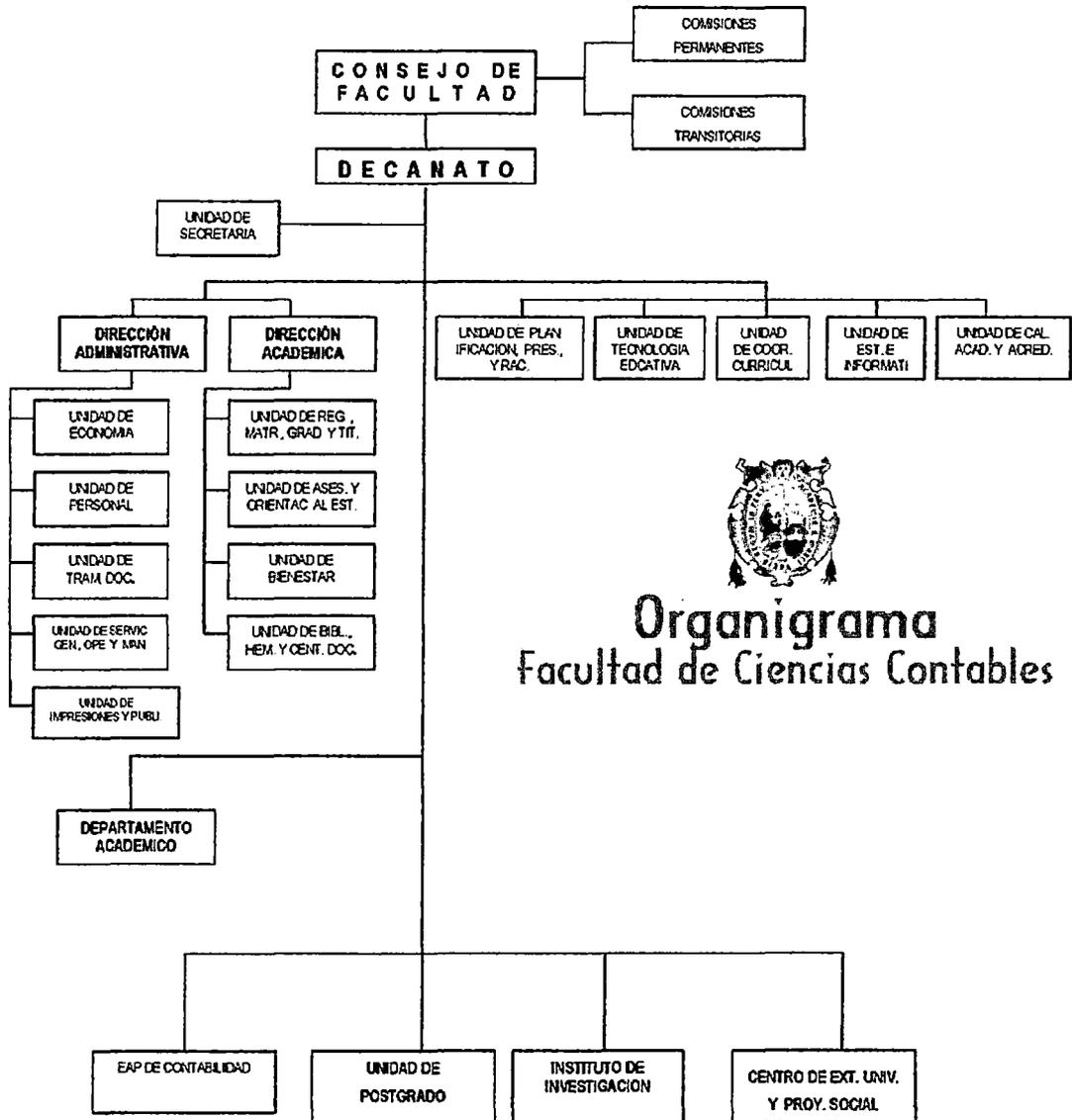
- a. Centro Preuniversitario
- b. Oficina del Sistema Único de Matricula
- c. Oficina General de Editorial, Imprenta, Biblioteca Central y Librería.

## **G. ORGANOS DE LINEA**

### **a. FACULTADES**

- a.1 Órgano de Gobierno y Dirección
- a.2 Órgano de Asesoramiento
- a.3 Órganos de apoyo
  - a.3.1 Secretaria Académica
  - a.3.2 Secretaria Administrativa
- a.4 Órganos de Apoyo Académico
  - a.4.1 Departamentos Académicos
- a.5 Órganos de Línea
  - a.5.1 Escuela Académico Profesional
  - a.5.2 Unidad de Postgrado
  - a.5.3 Instituto de Investigación
  - a.5.4 Centro de Extensión Universitaria y Proyección Social

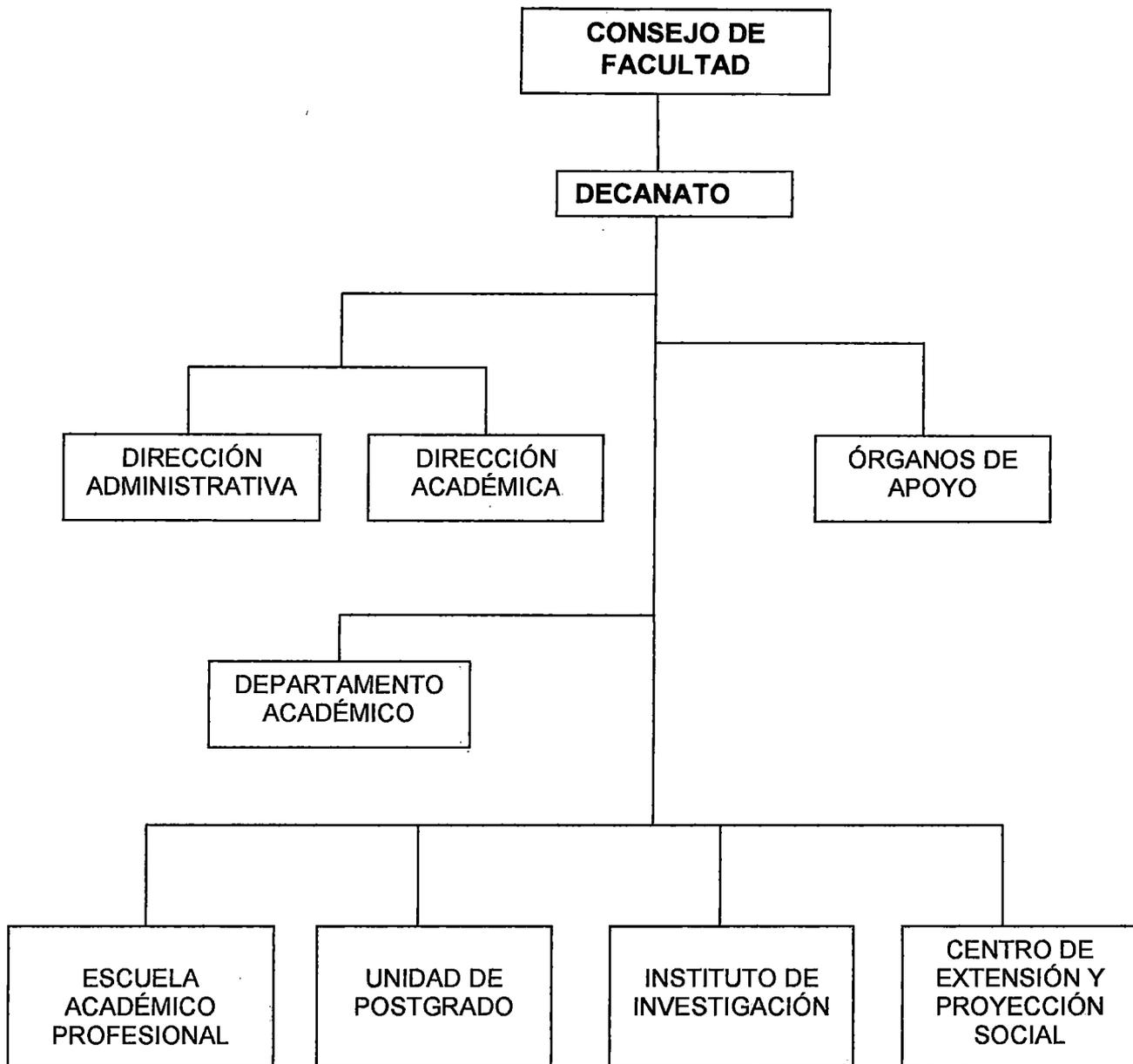
## ANEXO 03



## ANEXO 04



**Organigrama básico elaborado de una Universidad  
Pública o Privada de acuerdo con la Ley Universitaria  
del Perú N° 23733**



**Organigrama básico elaborado para una Facultad**