

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD 7**



**TÍTULO: CONTROL Y PLANIFICACIÓN DE LAS  
TAREAS DE LOS PROFESORES DE LA  
FACULTAD 7.**

**Trabajo de Diploma para optar por el título de Ingeniero en  
Ciencias Informáticas**

**Autores:**

**Serguey Rodríguez Pérez.**

**Adonis Rodríguez Delgado**

**Tutores:**

**Ing. Lourdes Escalona Peral.**

**Ing. Mainoldis Fuentes Suárez**

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

**Año 50 de la Revolución**

*"Mi confianza en el triunfo final de lo que creo, es completa."*

*Che*

## **DEDICATORIA**

A nuestros padres y abuelos, algunos de los cuales ya no se encuentran con nosotros, por su dedicación y esfuerzo por lograr que nos convirtiéramos en lo que somos hoy.

## AGRADECIMIENTOS

Queremos agradecer:

A Omar García Bonelly, Darien Menendez Molina, Lester Manuel Rangel Lorenzo, Maurich Caballero Nuñez, Ernesto Jordan, Daniel García Gonzalez y Maribel Fernandez Vega por ayudarnos a lograr lo mejor de este trabajo de diploma.

A nuestros tutores, que pese a su juventud asumieron esa tan difícil tarea.

A todos los que no pusimos, pero que de una forma u otra han contribuido a nuestra formación profesional y personal, que no se nos han olvidado, pero no habría espacio, a todos, muchas gracias.

De Serguey:

A mi abuelo Orestes y mi hermano Sergio por darme tanto animo y voluntad para seguir superándome. A mi mamá, papá y Leo mi otro padre por haberme guiado por el camino correcto, y por haber estado cuidando de mi durante toda mi vida. A Diana, mi novia, por su amor infinito, paciencia y ser una luz en mi camino. A mi tía Mirna y mi abuela Teté por haberme ayudado y cuidado tanto .A Mary y a Raulito por su derroche de bondad conmigo.

De Adonis:

A mi abuela Paula y mi Abuelo Bonifacio, a mi mamá Isaura, por todo lo que ha luchado por mí, su sacrificio, por enseñarme tanto de la vida y por ayudarme a entenderla, a mi padre Jenovevo. A todos mis primos que han hecho el papel de hermanos, a José Manuel, Antonio Luis, María Caridad, Celestino, Carlos Alberto, Michael, Juan Carlos, Yamet, Bertha, Iran y su esposa, Dunia.

A toda mis tias y tios, a mis vecinos y amigos del barrio, a Yunelis, Imirsis, Yunansi Vladimir, Osmel, Lixan, Carlos, Reinier discúlpeme si se me quedo alguno, a todos por confiar en mi desde el principio y por brindarme su apoyo incondicional.

A mis amigos de siempre, a Dayami, Dayana, Sergio, Gabriel, Pedro, Carlos, Ariel, Yunieski, Alioth, Osmani, Lindsay, por hacer de mi una mejor persona y por todos los favores.

A mis amigos de la Univesidad, a Yenly, Leydis, a mi compañero de tesis Serguey , a todos los del grupo y los que han compartido conmigo y me han apoyado estos 5 años de carrera.

A todos, gracias.

## **DATOS DE CONTACTO**

### **Ing. Lourdes Escalona Peral:**

Profesor graduado de Ingeniero Informático en el año 2004 en la Universidad de Holguín. Ha impartido las asignaturas Ingeniería de Software 1 y 2 y Seminario de Tesis. Fue líder del proyecto Atención Primaria de Salud durante dos años consecutivos y asesora de calidad de la facultad durante un año.

Ha tutorado tesis de perfil de análisis y diseño, implementación de sistema y atendiendo al área de calidad, Pruebas de software.

Posee la categoría docente de Instructor y cursa la maestría Gestión de Proyectos en la UCI. Se ha desempeñado como Jefe de Dpto. de la Especialidad de la Facultad 7, en los últimos 2 años.

Datos de contacto: Empresa: UCI Dirección: Carretera a San Antonio Km. 2 1/2 Reparto Torrens, Infraestructura productiva de la UCI, Ciudad Habana.

Teléfono: 835-8131. e-mail: [lescalonap@uci.cu](mailto:lescalonap@uci.cu).

### **Ing. Mainoldis Fuentes Suárez:**

Profesor graduado de Ingeniero en Ciencias Informáticas en el año 2007 en la Universidad de Ciencias Informáticas. Ha impartido las asignaturas de Programación I, Programación II e Inteligencia Artificial. Tiene la categoría de Instructor Recién Graduado. Cursa la maestría en Informática Aplicada. Es Líder de Proyecto de la facultad 7.

Datos de contacto: Empresa: UCI Dirección: Carretera a San Antonio Km. 2 1/2 Reparto Torrens, Infraestructura productiva de la UCI, Ciudad Habana.

e-mail: [mfuentes@uci.cu](mailto:mfuentes@uci.cu)

# RESUMEN

La Universidad de las Ciencias Informáticas participa en el proceso de informatización de la sociedad. En la facultad 7, surge la necesidad de controlar el trabajo de los profesores y sustituir la forma manual o semi-automatizada del trabajo que en este sentido realizan los directivos. Lo que resulta de gran importancia ya que influye de manera decisiva en el buen funcionamiento de la facultad. Toda esta situación, ha llevado a concebir el presente trabajo de diploma que plantea como objetivo desarrollar un sistema para el Control y Planificación de las Tareas de los Profesores de la Facultad 7.

Para el desarrollo se siguió como metodología de desarrollo el Proceso Unificado de Desarrollo del Software de Rational, siguiendo todas las etapas que esta propone y generando los artefactos necesarios. El sistema está soportado sobre la plataforma Microsoft .Net, en el lenguaje C#, a través de la herramienta Visual Studio.

La explotación de dicho sistema, permitirá al personal de la facultad, involucrados en la gestión de información de los profesores, fundamentalmente la información asociadas a la planificación y seguimiento de las tareas, realizar una evaluación del desempeño de los profesores así como almacenar información individual de cada uno de manera organizada y centralizada, permitiéndose la realización y emisión de los reportes necesarios a cualquier instancia, dentro y fuera de la facultad.

## PALABRAS CLAVES

- ✓ Aplicación Web
- ✓ Gestión de Recursos Humanos
- ✓ Plataforma .NET
- ✓ Microsoft Visual Studio
- ✓ Bases de Datos
- ✓ Planificación de tareas.
- ✓ Evaluación y seguimiento de tareas.
- ✓ Control de tareas.
- ✓ Sistemas de planificación

# ÍNDICE GENERAL

INTRODUCCIÓN .....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....	6
1.1 Sistema de Control de Gestión.....	6
1.2 Necesidad de un Sistema Control de Gestión de Profesores.....	8
1.3 Objeto de estudio.....	8
1.4 Análisis de otras soluciones existentes.....	10
1.5 Aplicación Web.....	11
1.6 Tecnologías .....	12
1.7 Arquitectura .....	18
1.8 Bases de Datos .....	23
1.9 Metodologías.....	27
Conclusiones.....	29
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA .....	31
2.0 Modelo de Negocio .....	31
2.1 Requisitos o Requerimientos. ....	43
Conclusiones.....	58
CAPÍTULO 3 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA .....	59
3.0 Diagramas de colaboración.....	59
3.1 Diagramas de clases de Diseño .....	59
3.2 Descripción de las clases persistentes .....	65
3.4 Principios de diseño.....	84
3.5 Generalidades de la Implementación. ....	85
3.6 Prueba del sistema propuesto.....	87
Conclusiones.....	91
CONCLUSIONES .....	92
RECOMENDACIONES .....	93
REFERENCIAS BIBLIOGRÁFICAS.....	94
BIBLIOGRAFÍA .....	97
ANEXOS.....	100
GLOSARIO DE TÉRMINOS .....	103

## ÍNDICE DE FIGURAS

Figura 1 Modelo de programación de tres capas. ....	19
Figura 2 Diagrama de Casos de Uso del Negocio .....	34
Figura 3 Diagrama de Actividades Caso de Uso: Contratar Profesor .....	39
Figura 4 Diagrama de Actividades Caso de Uso: Controlar Asistencia Actividades.....	39
Figura 5 Diagrama de Actividades Caso de Uso: Elaborar Plan de Actividades .....	40
Figura 6 Diagrama de Actividades Caso de Uso: Buscar Datos de los Profesores.....	41
Figura 7 Diagrama de Actividades Caso de Uso: Captación Alumno Ayudante .....	42
Figura 8 Diagrama de Objetos .....	43
Figura 9 Diagrama de caso de uso del sistema .....	48
Figura 10 Relaciones entre las clases principales que conforman la extensión de UML para Web .....	60
Figura 12 Diagrama de clase de diseño Gestionar Profesor .....	61
Figura 13 Diagrama de clase de diseño Registrar Asistencia .....	62
Figura 14 Diagrama de clase de diseño Buscar .....	63
Figura 14 Diagrama de clase de diseño Gestionar Tareas .....	64
Figura 17 Diagrama de clases persistentes .....	75
Figura 18 Modelo de Datos .....	76
Figura 19 Diagrama de Despliegue.....	86
Figura 20 Diagrama de Componentes .....	87
Figura 21 Complejidad Ciclomática.....	88
Figura 22 Prueba de caja negra .....	89
Figura 23 Diagrama de colaboración Gestionar Profesor.....	100
Figura 24 Diagrama de colaboración Gestionar Alumno Ayudante .....	101
Figura 25 Diagrama de colaboración Buscar .....	102



## ÍNDICE DE TABLAS

Tabla 1 Actores del negocio.....	32
Tabla 2 Trabajadores del negocio .....	32
Tabla 3 Descripción textual del Casos de Uso Contratar Profesores .....	35
Tabla 4 Descripción textual del Casos de Uso Controlar Asistencia Actividades .....	35
Tabla 5 Descripción textual del Casos de Uso Elaborar Plan de Actividades .....	36
Tabla 6 Descripción textual del Casos de Uso Buscar Datos del Profesor.....	37
Tabla 7 Descripción textual del Casos de Uso Captación Alumno Ayudante ...	38
Tabla 8 Descripción de los actores del sistema .....	47
Tabla 9 Descripciones del Caso de Uso del Sistema Gestionar Datos del Profesor.....	50
Tabla 10 Descripciones del Caso de Uso del Sistema Gestionar Asistencia Actividades.....	51
Tabla 11 Descripciones del Caso de Uso del Sistema Buscar Información .....	53
Tabla 12 Descripciones del Caso de Uso del Sistema Gestionar Actividad.....	55
Tabla 13 Descripciones del Caso de Uso del Sistema Autenticarse .....	55
Tabla 14 Descripciones del Caso de Uso del Sistema Gestionar Asistencia .....	55
Tabla 15 Descripciones del Caso de Uso del Sistema Gestionar Privilegio de Usuario .....	57
Tabla 16 Clase entidad profesor general. ....	67
Tabla 17 Clase entidad alumno ayudante.....	67
Tabla 18 Clase entidad asistencia. ....	68
Tabla 19 Tabla clase entidad tarea.....	68
Tabla 20 Clase interfaz Jefe_Departamento_GestionarDatosAAyudante.....	69
Tabla 21 Clase interfaz Jefe_Departamento_GestionarDatosProfesor. ....	70
Tabla 22 Clase interfaz Directivos_GestionarActividades.....	70
Tabla 23 Clase interfaz Directivos_GestionarActividades.....	70
Tabla 24 Tabla clase AccesoDatos. ....	73
Tabla 25 Tabla controladora cc_gestionartareas. ....	73
Tabla 26 Tabla controladora cc_buscar_informacion. ....	73
Tabla 27 Profesor general.....	77
Tabla 28 Proyecto .....	78
Tabla 29 Área temática.....	78
Tabla 30 Polo científico .....	79
Tabla 31 Cargos.....	79
Tabla 32 Tipo contrato.....	79
Tabla 33 Categoría docente .....	79
Tabla 34 Evaluación.....	80
Tabla 35 Títulos universitarios .....	80
Tabla 36 Grado Científico.....	80
Tabla 37 Provincia.....	80
Tabla 38 Municipio .....	80
Tabla 39 Profesor Adjunto.....	81
Tabla 40 Profesor Interno.....	81

Tabla 41 Profesor títulos universitarios .....	81
Tabla 42 Profesor asignatura .....	81
Tabla 43 Asignatura .....	82
Tabla 44 Disciplina .....	82
Tabla 45 Departamento .....	82
Tabla 46 Alumno Ayudante .....	83
Tabla 47 Tarea .....	83
Tabla 48 Asistencia .....	84
Tabla 49 Usuarios .....	84
Tabla 50 Privilegio .....	84
Tabla 51 Tabla de comparación de la complejidad ciclométrica .....	88
Tabla 52 Prueba de caja negra caso de prueba 1 .....	90
Tabla 53 Prueba de caja negra caso de prueba 2 .....	90
Tabla 54 Prueba de caja negra caso de prueba 3 .....	91
Tabla 55 Prueba de caja negra caso de prueba 4 .....	91

# INTRODUCCIÓN

Con el paso del tiempo, la informática va tomando auge y desarrollándose, a tal punto que actualmente se desarrolla una etapa de informatización masiva, en la que la mayoría de los países buscan soluciones eficientes y rápidas a sus problemas, apoyándose principalmente en esta ciencia.

La industria del software en Cuba ha ido en ascenso a través de los años logrando alcanzar producciones de alta calidad que le ha permitido adentrarse en las más diversas esferas de la economía y la sociedad cubana, con el objetivo de alcanzar mayor eficiencia. Sin embargo en la actualidad aún existen un gran número de empresas e instituciones que se encuentran envueltas en un gran flujo de información procesado de forma manual lo que provoca cierta lentitud en el manejo de dicha información.

En la Universidad de las Ciencias Informáticas (UCI), creada bajo pleno desarrollo de la informática en el país, se impulsa una estrategia para informatizar las diferentes áreas administrativas y de servicios. La UCI está integrada por 10 facultades, donde cada una, cuenta con una estructura administrativa formada por: un decano y tres vicedecanos (Vicedecano de Docencia, Vicedecano de Residencia, Vicedecano de Producción) y 5 departamentos (Técnicas de programación, Ciencias básicas, Sistemas digitales, Humanidades, Ingeniería de software y Práctica profesional).

Los departamentos a su vez, cuentan con una estructura interna conformada por un jefe de departamento, jefes de disciplinas y jefes de asignaturas. La información que se gestiona en cada uno de los departamentos, asociadas a las tareas y actividades que deben cumplir los profesores, se torna un tanto compleja cuando coexisten un conjunto de variables que pudieran afectar la planificación y seguimiento de las mismas.

Para el estudio y análisis de dichas variables, se tomó como punto de partida el departamento de Ingeniería de Software y Práctica Profesional (ISW-PP). Este es uno de los departamento más complejos de la facultad, actualmente manipula información asociada a 47 profesores y alrededor de 40 alumnos ayudantes. Está integrado por dos disciplinas: Práctica profesional e Ingeniería y Gestión de Software, en el caso de Práctica profesional la misma está compuesta por 7 asignaturas fundamentales:

Práctica Profesional (van desde 1, en primer año hasta 5, en quinto), Historia de la informática, Metodologías de la Investigación Científica. También tiene en cuenta los perfiles de la Facultad, de Salud e Imágenes. Es importante señalar, que a pesar de que el Perfil de Calidad, es dirigido a nivel de Universidad, el departamento da seguimiento a la formación de los estudiantes que pertenecen al Grupo de Calidad de la Facultad.

Por su parte la disciplina Ingeniería y Gestión de Software está compuesta por 4 asignaturas fundamentales: Ingeniería de Software1 y 2, Gestión del Software y Base de Datos. De forma general, ambas disciplinas son rectoras en la formación del Ingeniero en Ciencias Informáticas y sus procesos docentes educativos aún están en un proceso de organización y consolidación.

La preparación y superación del claustro para enfrentar disímiles tareas asignadas por la dirección de la UCI, de la Facultad, y específicamente la dirección del departamento, es objetivo primordial para el buen funcionamiento de la organización, por lo que se requiere fundamentalmente de una buena preparación de los cuadros de dirección, que en su mayoría son profesores jóvenes e inexpertos.

Estos últimos, desde su posición de dirigentes administrativos, tienen la máxima responsabilidad de gestionar la información asociada a los profesores y alumnos ayudantes, de definir la carga docente de los mismos, guiar su superación tanto política como profesional y metodológica, así como promover las actividades de extensión, producción e investigación, quedando reflejado en el plan de trabajo individual del profesor o alumno ayudante, según sea el caso.

Para la elaboración de dicho plan de trabajo los jefes de departamentos tienen en cuenta los intereses particulares de sus profesores, los objetivos y estrategias de la facultad, los del departamento y específicamente los de la disciplina y asignatura.

Sin embargo este sistema de elaboración de los planes de trabajo aún carece de la calidad a la que se aspira en la universidad: no se ha logrado concretar las tareas específicas a cumplir según el rubro indicado; no existe motivación para la elaboración del mismo por parte de los profesores, ya que les resulta tedioso y en muchas ocasiones complejo y nuevo, para el caso de los adiestrados, entre otras.

La planificación y concepción de los planes de trabajo resulta un tanto engorroso, pero a su vez, también es complicado dar seguimiento a las tareas de cada de ellos,

por todas las razones expuestas anteriormente y sin dejar de destacar el cúmulo de actividades y misiones adicionales, que diariamente hay que asignar a profesores y estudiantes en la facultad y que de una forma u otra contribuyen a enriquecer el plan de trabajo.

Un Sistema de Gestión de Recursos Humanos facilita la gestión de todos los datos relacionados con la disponibilidad de los recursos para la definición de categorías ocupacionales (dirigentes, técnicos, administrativos, obrero, servicio), áreas de trabajos, horarios de trabajo, asistencias del personal, contratación y movimientos del personal, fundamentos necesarios para realizar los controles y verificaciones de todos los datos afines con el personal que labora en la entidad.

La elaboración de reportes que hoy en día pueden tornarse dificultosos por el número de involucrados, podrán realizarse con mucha facilidad y rapidez a través de una aplicación informática. El valor de los recursos humanos, sus conocimientos, estudios y las diferentes actividades que realiza, hace necesario el control del mismo, a través de una herramienta que centralice y recopile la mayor cantidad de información y la tenga accesible y disponible en cualquier situación que se plantee, estos reportes puede ser por ejemplo: por edad, sexo, antigüedad, nivel académico, asignaturas u otra instancia.

En la actualidad, la mayoría de esta Información que gestionan los directivos de la facultad se realiza de forma manual, apoyados por herramientas informáticas tales como Microsoft Office del cual se usan programas como: Word, Excel y PowerPoint, lo que provoca un ineficiente acceso a los datos de los profesores y alumnos ayudantes, de modo concentrada y organizada, lo que trae como consecuencias pérdida de tiempo, entrega de información tardía, redundancia de información.

Dadas las dificultades antes mencionadas se plantea como **Problema a resolver:**

¿Cómo facilitar la gestión de la información referente a los profesores y alumnos ayudantes del departamento de Ingeniería de Software y Práctica Profesional de la facultad 7?

### **Objeto de estudio:**

La investigación es, en el proceso de gestión de los Departamentos Docentes de la facultad 7.

### **Campo de acción:**

La gestión de la Información, Planificación y Control de las Tareas de los Profesores del departamento de Ingeniería de Software y Práctica profesional de la Facultad 7.

### **Objetivo general:**

Desarrollar una aplicación que facilita la gestión de la información en los departamentos docentes de la facultad 7.

Para alcanzar el objetivo propuesto se definieron las siguientes **tareas de la investigación:**

- ✓ Valorar las tendencias actuales sobre planificación y gestión de tareas.
- ✓ Analizar el proceso actual de la planificación y control de tareas de profesores en los departamentos de la facultad 7.
- ✓ Seleccionar o definir las diferentes tecnologías a utilizar.
- ✓ Elaborar el modelamiento de negocio, captura de requisitos, modelo análisis y diseño de la aplicación propuesta.
- ✓ Desarrollar prototipo no funcional del sistema.
- ✓ Implementar versión del producto.

Con el despliegue de esta aplicación en los departamentos docentes, se espera que la facultad pueda tener los procesos relacionados con la gestión de los profesores, el control de los mismos y la gestión de las tareas de forma informatizadas; que se garantice la seguridad y confiabilidad de la información, así como que se viabilice el flujo de información hacia cualquier instancia.

El documento está estructurado por 3 capítulos:

**Capítulo 1 Fundamentación Teórica:** muestra aspectos generales de los sistemas de control y planificación de las tareas, utilizados fundamentalmente en la facultad. Se fundamenta el objeto de estudio de la investigación y se hace una propuesta, a partir

del estudio de las tendencias de las tecnologías actuales, de las herramientas y tecnologías a utilizar para el desarrollo de la solución propuesta.

**Capítulo 2 Presentación de la solución propuesta:** se muestra la modelación del negocio, se recogen los requerimientos que debe cumplir la aplicación, tanto funcionales como no funcionales y por último se realiza el modelo de casos de uso del sistema, que incluye el diagrama de casos de uso, así como la descripción de cada uno de los que se consideraron críticos.

**Capítulo 3 Construcción de la solución propuesta:** se modelan los artefactos asociados a los flujos de trabajo de Análisis y Diseño, Implementación y Prueba. Y se abordan además un conjunto de principios de diseño como estándares de la interfaz de la aplicación y patrones de diseño.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se analizan los principales elementos teóricos que sustentan el problema a resolver del trabajo. El mismo tiene como finalidad ofrecer un análisis del marco teórico del tema de planificación de tareas y recursos, basándose en el estudio de tendencias actuales y sistemas informáticos para este fin.

Se explica en qué consiste la Gestión de Información de Profesores y como se realiza este proceso en una facultad de la UCI, tomando como base el trabajo de los directivos, fundamentalmente los Jefes de Departamentos, en la gestión de la información, planificación y control de las tareas de los profesores.

Además se fundamenta la selección de las herramientas y tecnologías que se utilizan para el modelado, diseño e implementación de la aplicación web. Se describen la arquitectura y el gestor de bases de datos que se utilizará y de esta forma se formulan las premisas a la propuesta de solución informática para resolver la situación problemática planteada.

## 1.1 Sistema de Control de Gestión

Antes de analizar en qué consiste o las características fundamentales de un Sistema de Control de Gestión, se señalan los conceptos más importantes relacionados con el control y la gestión.

### *Control:*

Es un mecanismo preventivo y correctivo adoptado por la administración de una entidad, que permite la oportuna detección y corrección de desviaciones, ineficiencias o incongruencias en el curso de la formulación, instrumentación, ejecución y evaluación de las acciones, con el propósito de procurar el cumplimiento de la normatividad que las rige, y las estrategias, políticas, objetivos, metas y asignación de recursos.[1]

### *Gestión:*

Es el conjunto Planificación - Organización; Planificación equivale a la formulación de objetivos y las líneas de acción para alcanzarlos, se centra en seleccionar los objetivos, elaborarlos en términos productivos; Organización es la estructuración de tareas, distribución de responsabilidades y autoridad, dirección de personas y coordinación de esfuerzos en vías de la consecución de los objetivos, establecimiento



de las estructuras formales de división del trabajo dentro del subsistema, determinar, enumerar y definir las actividades requeridas, la responsabilidad de realizarlo.[2]

El **control de gestión** es un proceso que se utiliza para guiar la gestión empresarial hacia los objetivos de la organización y a la vez constituye un instrumento para evaluarla.[3]

El control de la gestión tiene otros fines más específicos que permiten a las empresas alcanzar mejores resultados administrativos. A continuación se detallan algunos de estos aspectos:

✓ **Informar:**

Transmitir y comunicar la información necesaria para la toma de decisiones.

✓ **Coordinar:**

Encaminar todas las actividades eficazmente a la consecución de los objetivos.

✓ **Evaluar:**

Valorar el desenvolvimiento de los recursos humanos en el cumplimiento de metas u objetivos trazados.

✓ **Motivar:**

Impulsar y encaminar el trabajo de los dirigentes y demás responsables en consecuencia del alcance de los objetivos.

Actualmente existen algunos Sistemas Informáticos de Control y Gestión, los cuales deben cumplir con una serie de requisitos para su funcionamiento eficiente:[4]

- Ser entendibles.
- Seguir la forma de organización.
- Rápidos.
- Flexibles.
- Económicos

Después de fundamentar en qué consiste un Sistema de Control y Gestión, se plantea la necesidad de un Sistema de Control y Gestión de Recursos Humanos.

## **1.2 Necesidad de un Sistema Control de Gestión de Profesores.**

Actualmente en la facultad 7, los Jefes de departamentos, con la colaboración de las secretarías, son los máximos responsables de atender los recursos humanos internos en el área, teniendo en cuenta sus necesidades, derechos laborales de cada uno de ellos, así como las estrategias de la facultad y la universidad.

Las técnicas o métodos tradicionales que usan hoy en día, los encargados de realizar todo el proceso de gestión de recursos humanos, son insatisfactorios, debido a que generar cualquier tipo de información relacionada con este tema implica horas de trabajos.

La productividad y la satisfacción laboral están vinculadas a las condiciones de trabajo de la institución. En la UCI, los directivos, profesores y estudiantes se ven involucrados diariamente en disímiles actividades, cuyo cumplimiento y seguimiento se ve afectado por una inadecuada planificación, en algunos casos por la escasez de recursos materiales y generalmente por la inexperiencia del claustro joven que la integra.

## **1.3 Objeto de estudio**

### **1.3.1 Gestión de Información de Profesores y Alumnos Ayudantes**

La implantación de un Sistema de Control para la Gestión de Información de Profesores y Alumnos ayudantes por los diferentes departamentos, permiten un mejor desarrollo de la facultad como célula base del funcionamiento de la universidad y con ello el bienestar de todos los vinculados con esta tarea, tendrán a la mano información detallada sobre cualquier profesor o actividad desarrollada, así como la asistencia a las mismas.

A través de este medio se espera que profesores y alumnos ayudantes tengan datos personales y profesionales de forma informatizada, eliminando el uso de manuscritos que imposibilitan que informaciones importantes de los mismos, no estén al alcance de los directivos de la facultad, en cualquier momento que pudieran necesitarlos.

La gestión de los recursos humanos en cualquier centro educacional constituye un eslabón fundamental para su correcto funcionamiento. Los sistemas de gestión de información de profesores y alumnos ayudantes tienen como objetivo principal recopilar y procesar gran parte de la información, fundamentalmente las asociadas a las tareas o misiones que estos deben cumplir, mostrando al usuario una versión siempre actualizada de los datos que el mismo solicite.

Estos sistemas almacenan todo lo concerniente a los profesores como sus datos particulares, su currículo, la asistencia a las diferentes actividades, así como las asignaturas que imparte, entre otros. Esto lo convierte en una poderosa herramienta de trabajo que le permite a cualquier centro de estudios hacer eficientes sus procesos, optimizando así sus recursos.

### **1.3.2 El proceso de Gestión de Información de Profesores y Alumnos Ayudantes en la Facultad 7**

La Facultad 7 cuenta con alrededor de 140 profesores en su claustro, los cuales son: profesores que son prestación de servicios, adjuntos a la facultad y los que están a tiempo completo.

Entre las informaciones importantes que se gestionan tanto de los profesores como de los alumnos ayudantes, se tienen:

- Planilla de datos generales del profesor.

- Plan de trabajo individual del profesor.

- Evaluación del plan de trabajo del profesor semestral y anualmente.

- Planilla de datos generales de los alumnos ayudantes.

- Plan de trabajo de los alumnos ayudantes.

- Evaluación semestral de los alumnos ayudantes.

La realización de dichos procesos en cierta medida se ven influenciados por diferentes factores. Como se ha dicho anteriormente, falta experiencia y capacitación de los responsables para la elaboración de dicha documentación.

Dificultades para el llenado de estos modelos por parte de los profesores ya que los mismos hacen rechazo a algunos de estos documentos, pues son extensos y engorrosos.

La gestión de información de los departamentos se resume a la realización de estos modelos en herramientas ofimáticas o de forma manual.

Los reportes de las informaciones asociadas con los profesores y alumnos ayudantes de la facultad, en la mayoría de las ocasiones, privan al jefe de departamento de realizar otras actividades ya que se pierde un tiempo considerable adaptando, pues el formato en que se brinda la información, varía en dependencia de la instancia o área que la solicita.

La evaluación de las actividades y planes de trabajo del profesor, es otro de los puntos

importantes y cabe centrar la atención, en este aspecto, ya que existen un conjunto de factores que emiten un criterio del profesor.

En el caso del profesor, los vicedecanos de los diferentes rubros, emiten su criterio acerca de la evaluación del profesor, también se tiene en cuenta la valoración de las organizaciones políticas y de masas, así como la de los coordinadores de años que son los encargados de velar por la promoción, actividades, que se realicen por cada .

Cuando se trata de un alumno ayudante, aunque la información que se gestiona es mucho más simple, hay que señalar que no debe dejar de elaborarse el plan de trabajo de los mismos y emitir una evaluación de su trabajo, teniendo en cuenta el criterio del profesor tutor, jefes de asignaturas, entre otros.

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

A nivel mundial se conoce la existencia de varios sistemas para la gestión y planificación, a continuación se analizan algunos:

Docproject y el Phproject ambos son **open source** o de código abierto, y permiten modelar y adaptarse a las exigencias del cliente. Estas herramientas son ineficientes en la gestión de la información de los recursos humanos ya que no permite configurar este modulo. En este caso para gestionar la información de los profesores del departamento se necesitan atributos específicos como por ejemplo universidad de procedencia, cargo, si es un profesor interno o externo entre otras, que estas no permiten configurar.

También se necesita cierta capacitación para poder operar con estas herramientas, lo que traería consigo una dificultad, ya que estos sistemas serian utilizados por personas que pudieran no ser informáticos y se necesitaría un nivel de capacitación mayor, pues están enfocados a la gestión de grandes proyectos.

Estas aplicaciones a pesar de no tener las características que se desean constituyen un punto de partida para la definición del diseño del sistema propuesto, de la gestión y seguimiento de las tareas.

Existen otras herramientas como el **microsoft project 2002** programa para el control de proyectos de planificación, costos, materiales. Este no exporta datos a ningún gestor de base de datos,

## 1.5 Aplicación Web.

Una aplicación es un programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo, suele resultar ser una solución informática para la informatización de ciertas tareas.[5]

Las aplicaciones web son aquellas en que los usuarios acceden a ellas en un servidor web a través de Internet o de una intranet. La popularidad de las aplicaciones web se debe a que basta con tener solamente una computadora, un navegador web y acceso a la Internet para poder acceder a la información requerida. Se destacan por su facilidad de uso, capacidad para actualizar y mantenerse sin distribuir e instalar software. A continuación se explican varias características:[6]

1. **Compatibilidad multiplataforma.** Tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software de escritorio. Incluyen varias tecnologías: Java, Flash, ASP y Ajax entre otras, permitiendo un desarrollo efectivo de programas, soportando los sistemas operativos principales.
2. **Actualización.** Están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo con la esperanza de que va a iniciar nuevas descargas y procedimientos de instalación (algunas veces imposible cuando usted está trabajando dentro de grandes organizaciones).
3. **Inmediatez de acceso.** No necesitan ser descargadas, instaladas y configuradas. Usted accede a su cuenta online y está lista para trabajar sin importar cuál es su configuración o su hardware.
4. **Menos requerimientos de memoria.** Las aplicaciones basadas en tecnologías web tienen muchas más razonables demandas de memoria RAM de parte del usuario final que los programas instalados localmente. Al residir y correr en los servidores del proveedor, esas aplicaciones basadas en web usan en muchos casos la memoria de las computadoras que funcionan de servidores, dejando más espacio para correr múltiples aplicaciones del mismo tiempo sin incurrir en frustrantes deterioros en el rendimiento.

Partiendo de la necesidad que plantea el cliente de que los usuarios tengan las facilidades que se exponen a continuación: Pues se necesita que los profesores y

demás usuarios de la aplicación puedan acceder al sistema desde cualquier computadora de la universidad ya que estos necesitan estar constantemente actualizándose dada la dinámica de esta universidad que siempre surgen tareas de repente, solamente utilizando un navegador y la dirección del sitio. Por todo lo descrito anteriormente se decide que la propuesta de aplicación sea web.

### 1.6 Tecnologías

A continuación se explican las diferentes tecnologías a utilizar pues estas son imprescindibles en la elaboración de la aplicación propuesta y se han escogido por su eficacia, compatibilidad entre ellas y experiencia y conocimiento de los desarrolladores.

#### 1.6.1 Plataforma .NET

Con el objetivo de simplificar el desarrollo de la aplicación, se propone que la misma se desarrolle teniendo en cuenta las bondades de la plataforma .NET. Esta plataforma soporta los estándares sobre los cuales se basan los servicios web. La plataforma .NET es una capa de software que se coloca entre el Sistema Operativo (SO) y el programador y que abstrae los detalles internos del SO. Se optó por utilizar esta plataforma para la construcción de la aplicación debido a que la misma presenta las siguientes características:[7]

- ✓ **Portabilidad:** Debido a la abstracción del programador respecto al SO, una aplicación .NET puede ser ejecutada en cualquier SO de cualquier máquina que disponga de una versión de la plataforma. En estos momentos la plataforma .NET tan solo está disponible para la familia Windows aunque se está desarrollando una versión para Linux de Corel.
- ✓ **Multilinguaje:** Cualquier lenguaje de programación puede adaptarse a la plataforma .NET y ejecutarse en ella.
- ✓ **Interoperabilidad:** La interoperabilidad entre diferentes trozos de código escritos en diferentes lenguajes es total.

## 1.6.2 Microsoft Visual Studio.NET

Después de decidido trabajar con la plataforma.Net se buscó que herramienta utilizar, se conoció de la existencia de la herramienta Visual Studio.NET. Se decidió trabajar con Visual Studio.NET pues el mismo permite diseñar, desarrollar, depurar e instalar soluciones basadas en .NET. Se pueden crear aplicaciones con formularios Windows o formularios Web que integran datos y lógica propia de la aplicación, así como servicios web.

Todos los lenguajes que son incorporados a Visual Studio pueden acceder a un conjunto común de herramientas, editores y diseñadores. Facilita la creación y personalización visual de controles de formulario Windows y Web. Brinda facilidades para el acceso a datos y asistentes para realizar determinadas acciones. Además es una herramienta de desarrollo multilenguaje integrado (Visual Basic .NET, C++, C# y J#) lo que la hace una de las herramientas más completa alcanzadas en la actualidad. Además brinda la posibilidad de utilizar un conjunto de tecnologías de desarrollo de aplicaciones web de ASP.NET.

### Algunas características de Visual Studio.Net [8]

- ✓ **Detección de errores automáticos:** Es posible ahorrar muchas horas de trabajo al utilizar Visual Studio.Net para detectar los errores antes de intentar ejecutar su aplicación .Los errores se subrayan de la misma forma que lo hacen algunos procesadores de textos mientras se escribe.
- ✓ **Herramientas de depuración:** Visual Studio.Net mantiene sus herramientas de depuración legendarias que le permiten observar su código en acción y seguir la pista del contenido de las variables.
- ✓ **Diseño de página:** Es posible crear una página atractiva con la facilidad de arrastrar y soltar integrada en el diseñador de formulario Web de Visual Studio.Net.

### Algunas ventajas de Visual Studio .Net[9]

- **Arquitectura flexible y ágil de aplicaciones**
  - ✓ Creada para los servicios Web XML
  - ✓ Constituye un modelo sencillo, flexible y basado en estándares para integrar, ampliar y publicar aplicaciones.

- **Máxima productividad del desarrollador**
  - ✓ Soporta varios lenguajes de programación.
  - ✓ Pueden reutilizar las aplicaciones escritas en cualquier lenguaje de programación.
- **Operaciones mejoradas**
  - ✓ Aumenta el rendimiento, la escalabilidad y la fiabilidad.
  - ✓ Ofrece al usuario un tiempo en actividad percibido del 100%.
  - ✓ Elimina el problema de conflictos entre versiones. Las nuevas aplicaciones pueden ejecutar aplicaciones existentes.

### 1.6.3 ASP .NET.

ASP.NET es una herramienta de trabajo de programación que puede utilizarse en un servidor para generar eficaces aplicaciones web, forma parte de la plataforma .NET de Microsoft y ofrece varias ventajas importantes acerca de los modelos de programación web entre ellas se encuentran:[10]

- ✓ **Mejor rendimiento.** ASP.NET es un código de compilado que se ejecuta en el servidor. A diferencia de sus predecesores, ASP.NET puede aprovechar las ventajas del enlace anticipado, la compilación *just-in-time*, la optimización nativa y los servicios de caché desde el primer momento. Esto supone un incremento del rendimiento antes de siquiera escribir una línea de código.
- ✓ **Compatibilidad con herramientas de primer nivel.** El marco de trabajo de ASP.NET se complementa con un diseñador y una caja de herramientas muy completos en el entorno integrado de programación de Visual Studio. Los controles de servidor de arrastrar y colocar y la implementación automática son sólo algunas de las características que proporciona esta eficaz herramienta.
- ✓ **Eficacia y flexibilidad.** La eficacia y la flexibilidad de toda esa plataforma se encuentra disponible para los programadores de aplicaciones Web. La biblioteca de clases de .NET Framework, la mensajería y las soluciones de acceso a datos se encuentran accesibles desde el web de manera uniforme. ASP.NET es también independiente del lenguaje, por lo que puede elegir el lenguaje que mejor se adapte a la aplicación o dividir la aplicación en varios lenguajes.
- ✓ **Simplicidad.** ASP.NET facilita la realización de tareas comunes, desde el sencillo envío de formularios y la autenticación del cliente hasta la implementación y la configuración de sitios.



- ✓ **Facilidad de uso.** Emplea un sistema de configuración jerárquico, basado en texto, que simplifica la aplicación de la configuración al entorno de servidor y las aplicaciones Web. Debido a que la información de configuración se almacena como texto sin formato, se puede aplicar la nueva configuración sin la ayuda de herramientas de administración local.
- ✓ **Escalabilidad y disponibilidad.** Se ha diseñado teniendo en cuenta la escalabilidad, con características diseñadas específicamente a medida, con el fin de mejorar el rendimiento en entornos agrupados y de múltiples procesadores. Además, el motor de tiempo de ejecución de ASP.NET controla y administra los procesos de cerca, por lo que si uno no se comporta adecuadamente (filtraciones, bloqueos), se puede crear un proceso nuevo en su lugar, lo que ayuda a mantener la aplicación disponible constantemente para controlar solicitudes.
- ✓ **Posibilidad de personalización y extensibilidad.** Presenta una arquitectura bien diseñada que permite a los programadores insertar su código en el nivel adecuado. De hecho, es posible extender o reemplazar cualquier subcomponente del motor de tiempo de ejecución de ASP.NET con su propio componente escrito personalizado.
- ✓ **Seguridad.** Con la autenticación de Windows integrada y la configuración por aplicación, se puede tener la parcialmente la seguridad de que las aplicaciones están a salvo.

También permite crear código muy eficiente, en aquellos puntos de la aplicación que son críticos. Además, permite aprovechar la experiencia creciente por parte del equipo de desarrollo; el alto rendimiento y escalabilidad, la seguridad mejorada sobre otras tecnologías web existentes.

### 1.6.4 Lenguaje de programación C#.

Se utiliza C# ya que es un lenguaje de programación orientado a objetos, aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado sobre ella. Por lo que programar usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes, disponibles en .NET.[11]

También este lenguaje posee una serie de características que favorece a la hora de su utilización las cuales se exponen a continuación:[12]

**Sencillez:** C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. Por ejemplo:

- El código escrito en C# es **autocontenido**, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera o ficheros IDL
- No se incluyen elementos poco útiles de lenguajes como C++ tales como macros, herencia múltiple o la necesidad de un operador diferente del punto (.)

**Modernidad:** C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular. La inclusión de una instrucción **foreach** que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario.

- **Orientación a objetos:** Como todo lenguaje de programación de propósito general moderno, C# es un lenguaje orientado a objetos. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro en tanto que no admiten ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.
  - C# soporta todas las características propias del paradigma de programación orientada a objetos: **encapsulación, herencia y polimorfismo**.
  - Respecto a la herencia, a diferencia de C++ y al igual que Java, C# sólo admite herencia simple de clases ya que la herencia múltiple y en la mayoría de los casos su utilidad puede ser simulada con facilidad mediante herencia múltiple de interfaces. De todos modos, esto vuelve a ser más bien una característica propia del CTS que de C#.
- Por otro lado y a diferencia de Java, en C# se ha optado por hacer que todos los métodos sean por defecto privados y que los redefinibles hayan de marcarse con el modificador **virtual** (como en C++), lo que permite evitar errores derivados de redefiniciones accidentales.
- **Orientación a componentes:** La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular

mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente **propiedades** (similares a campos de acceso controlado), **eventos** (asociación controlada de funciones de respuesta a notificaciones) o **atributos** (información sobre un tipo o sus miembros)

- **Gestión automática de memoria:** Como ya se comentó, todo lenguaje de .NET tiene a su disposición el recolector de basura que su función es eliminar automáticamente la memoria que ya no se está utilizando en el programa. Esto tiene el efecto en el lenguaje de que no es necesario incluir instrucciones de destrucción de objetos.
- **Seguridad de tipos:** C# incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que evita que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura.
- **Instrucciones seguras:** Para evitar errores muy comunes, en C# se han impuesto una serie de restricciones en el uso de las instrucciones de control más comunes. Por ejemplo, la guarda de toda condición ha de ser una expresión condicional y no aritmética, con lo que se evitan errores por confusión del operador de igualdad (==) con el de asignación (=); y todo caso de un **switch** ha de terminar en un **break** o **goto** que indique cuál es la siguiente acción a realizar, lo que evita la ejecución accidental de casos y facilita su reordenación.
- **Versionable:** C# incluye una **política de versionado** que permite crear nuevas versiones de tipos sin temor a que la introducción de nuevos miembros provoquen errores difíciles de detectar en tipos hijos previamente desarrollados y ya extendidos con miembros de igual nombre a los recién introducidos.
  - **Compatible:** Para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes, sino que también ofrece, a través de los llamados **Platform Invocation Services (PInvoke)**, la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLLs de la API Win32. Nótese que la capacidad de usar punteros en código inseguro permite que se pueda acceder con facilidad a este tipo de funciones, ya que éstas muchas veces esperan recibir o devuelven punteros.

Se ha decidido inclinarse por software propietario pues este brinda una serie de herramientas con facilidades a la hora de desarrollar la aplicación con rapidez y con excelente calidad ya que el tiempo era un factor fundamental pues los clientes necesitan con urgencia la aplicación propuesta, además no es un software para la comercialización por lo que no es necesario desarrollarlo en plataforma libre, también se garantiza la seguridad del sistema con los componentes que ofrece el Visual Studio 2005 que son de fácil manejo y configuración .

### 1.7 Arquitectura

#### 1.7.1 Arquitectura de tres capas. [13]

Es necesario que cualquier aplicación de mediana o gran envergadura esté separada en las capas de presentación, negocio y datos. Esto es para delimitar responsabilidades entre quien debe realizar ciertas tareas, y por que las capas no deben ni tienen por qué saber de cómo y con quien interactúan las otras capas.

El modelo actual de desarrollo ha demostrado que organizar los elementos de las aplicaciones en capas independientes puede lograr una mayor eficiencia durante el tiempo de desarrollo y mantenimiento. La programación en múltiples capas es la técnica más efectiva para aplicaciones empresariales, dividir los componentes de la aplicación en capas implica una fácil administración y rapidez en entornos cliente-servidor. Esta arquitectura consiste en dividir los componentes primarios de una aplicación, programarlos por separado y después unirlos, ya sea en tiempo de diseño o de ejecución.

Con el avance de la tecnología de Internet, han surgido las aplicaciones Web y ha aparecido un nuevo estándar, el modelo de las aplicaciones en n-capas. Este modelo por lo general está basado en un esquema de tres partes o de tres capas:

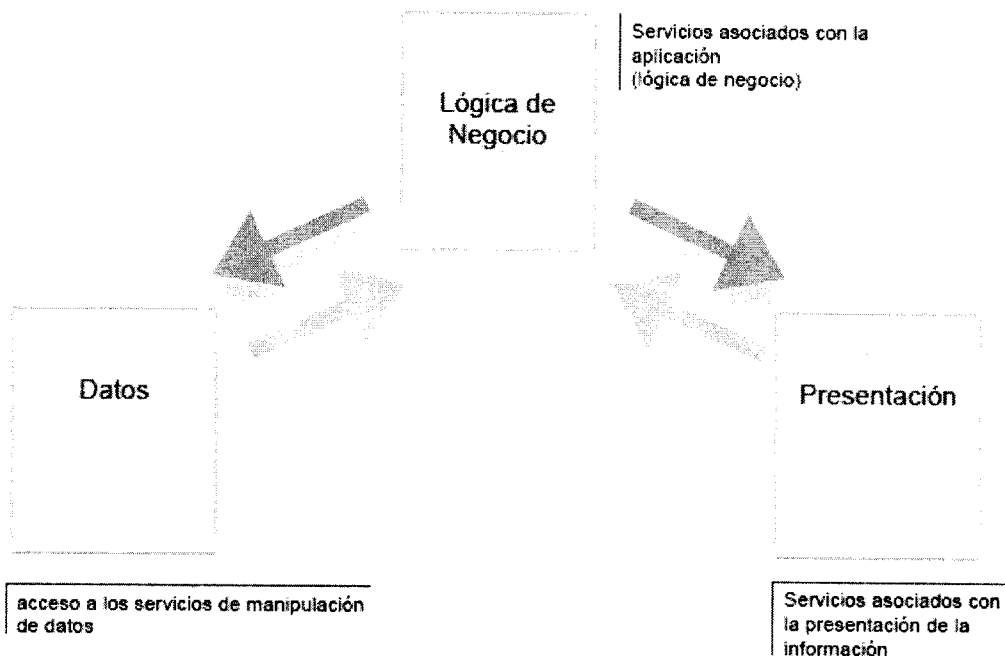
- Una capa para guardar y acceder a los datos (**capa de acceso a datos**)
- Una capa para centralizar la lógica del negocio (**capa de lógica del negocio** )
- Una interfaz gráfica para facilitar al usuario el uso del sistema (**capa de presentación**).

Se pueden alojar todas las capas en el mismo servidor, aunque también es posible alojar cada capa en varios servidores, en caso de la solución de esta aplicación que se propone, la capa de presentación y la capa lógica del negocio se alojan en un servidor mientras que la capa de datos se aloja en otro. Las tres capas, consisten en:

**Capa de presentación:** En esta capa se diseña todo lo que constituye la interfaz gráfica y la interacción del usuario con el software. Esta capa se comunica únicamente con la capa de lógica del negocio.

**Capa de lógica del negocio:** Es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario desde la capa de presentación y enviándole las respuestas tras el proceso. Se denomina capa de lógica del negocio pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para almacenar o recuperar datos de la base de datos.

**Capa de datos:** Es donde residen los datos. Está formada por una o más bases de datos que soportan todo el almacenamiento de datos. Reciben solicitudes de almacenamiento o recuperación de información desde la capa de lógica del negocio.



**Figura 1** Modelo de programación de tres capas.

Las principales ventajas de este modelo son las siguientes:

- Proporciona escalabilidad, capacidad de administración y utilización de recursos.

- Cada capa es un grupo de componentes que realiza una función específica, con lo que se puede actualizar una capa con independencia de las demás.
- Centralización de los aspectos de seguridad.
- No replicación de la lógica de actividad en los clientes, lo que permite que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios, reduciendo los costes de mantenimiento.[14]

### 1.10.2 Arquitectura Cliente-Servidor.[15]

La arquitectura **Cliente-Servidor** es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.[16]

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

Los clientes realizan generalmente funciones como:

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

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

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

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

Entre las principales características de la arquitectura **Cliente-Servidor** se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.

### **Servidores.[17]**

El servidor se utiliza para referirse al ordenador físico en el cual funciona un software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

En el caso de un servidor web, este término podría referirse a la máquina que almacena y maneja los sitios Web, y en este sentido es utilizada por las compañías que ofrecen hosting o hospedaje. Alternativamente, el servidor web podría referirse a software, como el servidor de aplicaciones de Internet Information Server, que funciona en la máquina y maneja la entrega de los componentes de las páginas Web como respuesta a peticiones de los navegadores de los clientes.

Los archivos para cada sitio de Internet se almacenan y se ejecutan en el servidor. Hay muchos servidores en Internet y muchos tipos de servidores, pero comparten la función común de proporcionar el acceso a los archivos y servicios.

Un servidor provee información a los ordenadores que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor.

### **Cliente.**

Es el que inicia un requerimiento de servicio. El requerimiento inicial puede convertirse en múltiples requerimientos de trabajo a través de redes. La ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente.

#### **Características de un cliente.**

- Envía peticiones
- Espera y recibe contestaciones del servidor.

En el modelo cliente/servidor se pueden encontrar las siguientes características:

1. El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
2. Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.
3. Un servidor da servicio a múltiples clientes en forma concurrente.
4. Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
5. La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
6. Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones Clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un solo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final.

También es importante hacer notar que las funciones Cliente/Servidor pueden ser dinámicas. Ejemplo, un servidor puede convertirse en cliente cuando realiza la solicitud de servicios a otras plataformas dentro de la red. Su capacidad para permitir integrar los equipos ya existentes en una organización, dentro de una arquitectura informática descentralizada y heterogénea.

7. Además se constituye como el nexo de unión más adecuado para reconciliar los sistemas de información basados en mainframes o minicomputadores, con aquellos otros sustentados en entornos informáticos pequeños y estaciones de trabajo.
8. Designa un modelo de construcción de sistemas informáticos de carácter distribuido. Su representación típica es una computadora, en donde el usuario dispone de sus propias aplicaciones de oficina y sus propias bases de datos, sin dependencia directa del sistema central de información de la organización, al tiempo que puede acceder a los recursos de este host central y a otros sistemas que la organización pone a su servicio.



## Principales ventajas de este modelo: [18]

- ✓ Con el uso de este esquema, se reducen los costos de producción de software y se disminuyen los tiempos requeridos.
- ✓ Reduce el costo del hardware requerido, llevando las aplicaciones a plataformas más baratas, aprovechando el poder de cómputo de los diferentes elementos de la red, y facilitando la interacción entre las distintas aplicaciones de la organización.
- ✓ Contribuye a una disminución de los costos de entrenamiento de personal, pues favorecen la construcción de interfaces gráficas interactivas, las cuales son más intuitivas y fáciles de usar por el usuario final.
- ✓ Facilita el suministro de información a los usuarios.
- ✓ Permite llevar más fácilmente la información a donde se necesita, contribuye a aumentar su precisión pues se puede obtener de la fuente (el servidor) y no de una copia en papel o en medio magnético.

## 1.8 Bases de Datos

Una **base de datos** es un conjunto de datos pertenecientes al mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar información.[19]

Un **sistema gestor de bases de datos** (SGBD) o **database management system** (DBMS) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un sistema gestor de base de datos permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.

Algunos ejemplos de sistema gestor de base de datos son Oracle, DB2, PostgreSQL, MySQL, MS SQL Server.

Un SISTEMA GESTOR DE BASE DE DATOS debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo sistema gestor de base de datos.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Las características de un Sistema Gestor de Base de Datos son:[20]

**Abstracción de la información.** Los Sistema Gestor de Base de Datos ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.

**Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

**Consistencia.** En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.

**Seguridad.** La información almacenada en una base de datos puede llegar a tener un gran valor. Los sistemas gestores de base de datos deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques con la intención de manipular o destruir la información. Normalmente, los sistemas gestores de base de datos disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

**Integridad.** Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada pero a pesar de eso los datos no quedan exentos de daños.

**Respaldo y recuperación.** Los SISTEMA GESTOR DE BASE DE DATOS deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.

**Control de la concurrencia.** En la mayoría de los entornos, lo más habitual es que sean muchas las personas que acceden a una base de datos ya sea para recuperar información o para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea.

### 1.8.1 SQL Server 2000[21]

SQL Server 2000 una solución integral de base de datos y análisis, que ofrece el rendimiento, escalabilidad y confiabilidad que requieren el exigente. El nuevo soporte de XML y HTTP simplifica el acceso a datos y el intercambio, mientras que las poderosas capacidades de análisis mejoran el valor de los datos. Las características de disponibilidad mejoradas maximizan el tiempo de actividad, las funciones de la administración avanzadas automatizan las tareas rutinarias y las herramientas mejoradas de programación y los servicios aceleran el desarrollo.

SQL Server 2000 es el eje principal de la administración y análisis de datos de la siguiente generación de productos y servicios de Microsoft .NET.

#### **Consulta, análisis y manipulación de datos en el Web.**

- Simplifica la integración de sistemas del back-end y la transferencia de datos por medio de firewalls que usan la completa funcionalidad de SQL Server 2000 XML y el soporte de otros estándares de Internet, como XPath, XSL y XSLT. Los desarrolladores de web pueden acceder a los datos usando XML sin la programación de base de datos relacional, mientras que los administradores de bases de datos pueden manipular los datos fácilmente en formato XML usando Transact-SQL (T-SQL) y los procedimientos almacenados.
- Conecta las bases de datos de SQL Server 2000 y cubos de OLAP usando el Web flexiblemente sin la programación adicional. Usa la segura conectividad de bases de datos en HTTP para asegurar que se puedan consultar los datos a través de un URL y las interfaces de usuario intuitivas.
- Habilita para el web e integra los sistemas de negocios existentes a través del penetrante soporte de XML.

- Aprovecha por completo sus recursos de hardware ejecutando aplicaciones aisladas múltiples en una sola computadora que use el soporte de instancias múltiples de SQL Server 2000.

**SQL Server 2000 construye aplicaciones rápidamente y asegura que se mantengan funcionando.**

- Realiza rápidamente aplicaciones de bases de datos robustas y escalables que usan las herramientas mejoradas de desarrollo de SQL Server 2000. El Analizador de Consultas de SQL Server 2000 ahora incluye un depurador de procedimientos almacenados. Los Data Transformation Services (DTS) extienden la capacidad para mover y transformar los datos desde cualquier fuente.
- Mejora la productividad con las mejoras de T-SQL. Las nuevas funciones definidas por el usuario permiten la reutilización de código y simplifican el desarrollo. Mantiene datos consistentes con la integridad referencial para controlar la propagación en operaciones de eliminación y actualización.

Por estas razones se decidió utilizar SQL Server 2000 como gestor de base de datos además admite la programación cliente-servidor que es la arquitectura que se tendrá como base para la elaboración de la aplicación.

### **1.8.2 Embarcadero ER Studio.[22]**

Embarcadero ER Studio es una de las herramientas CASE de diseño de bases de datos bases de datos relacionales a nivel lógico y físico. Permite hacer ingeniería inversa de bases de datos existentes. Ofrece soporte para todas las bases de datos relacionales más populares del mercado, además es una herramienta visual de diseño y modelado de objetos UML, que genera código para los lenguajes más populares.

Permite integrar los procesos de diseño de bases de datos y de objetos, frecuentemente desconectados, que ayuda a generar, mantener alta calidad y gran rendimiento en las aplicaciones de la base de datos desde un modelo lógico de los requerimientos de información y las reglas de negocio que definen la base de datos al modelo físico optimizado por las características específicas de esta. Permite visualizar la estructura, elementos clave y optimizar el diseño de las bases de datos, genera tablas u otras especificaciones en dependencia de la plataforma seleccionada. Tiene como ventajas:

- Facilidades de diseño de diagramas Entidad-Relación y Entidad-Relación extendido y transformación de este al modelo relacional, preservando las dependencias funcionales y sin pérdidas de información).
- Comparación comprensiva entre el modelo de datos y la base de datos.
- Soporta la separación del modelo lógico y del físico.

### 1.9 Metodologías

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no se lleva una metodología de por medio, se obtienen es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Sin embargo, muchas veces no se toma en cuenta el utilizar una metodología adecuada, sobre todo cuando se trata de proyectos pequeños de tres o cuatro meses.

El Lenguaje Unificado de Modelamiento UML (Unified Modeling Language) es la herramienta usada en la descripción y construcción de software reconocida por la industria como estandar mientras que el Proceso Unificado Rational (Rational Unified Process) es la metodología de Rational para el desarrollo y construcción de software basado integralmente en UML como soporte a la metodología.

#### 1.9.1 El Proceso Unificado de Rational (RUP). [23]

**RUP** en inglés significa (*Rational Unified Process*), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

En la actualidad existen varias metodologías. Se concluyó utilizar como base en el desarrollo de la solución **RUP** pues el mismo es un proceso que de manera ordenada define las tareas y responsables y aunque es un proceso que se concibió para proyectos grandes o robustos la metodología, dada las tendencias actuales. Otras características que se tuvieron en cuenta fueron:

- ✓ Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- ✓ Pretende implementar las mejores prácticas en Ingeniería de Software.
- ✓ Desarrollo iterativo e incremental.
- ✓ Administración de requisitos.
- ✓ Control de cambios.

- ✓ Modelado visual del software.
- ✓ Verificación de la calidad del software.

El RUP es un producto de Rational. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, entre otros.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). Además como esta metodología no es un estándar es decir que se puede adaptar a las necesidades del cliente facilita el trabajo de los desarrolladores.

### 1.9.2 Lenguaje Unificado de Modelado (UML).[24]

Para el modelado de la herramienta propuesta se utilizara **UML (Lenguaje Unificado de Modelado**, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está respaldado por el OMG (Object Management Group).

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar.

En el caso de las tecnologías que se escogen para el modelado de la aplicación se va a utilizar la metodología RUP (Proceso Unificado de Rational) cuyo lenguaje de modelado es UML. El desarrollo de sistemas con UML siguiendo el proceso unificado incluye actividades específicas, cada una de ellas a su vez contienen otras subactividades las cuales sirven como una guía de cómo deben ser las actividades desarrolladas y secuenciadas con el fin de obtener sistemas exitosos.

Consecuentemente el desarrollo de los sistemas puede variar de desarrollador en desarrollador, de proyecto en proyecto, de empresa en empresa adoptando siempre un Proceso de Desarrollo.

### **1.9.3 Herramienta Case utilizada:**

Como herramienta cases se utilizó la suite de rational rose la cual es una poderosa herramienta de modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo. Cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases. Contiene una serie de componentes para la captura de requisitos, gestión de configuración y para la generación de reportes entre otros.

### **Conclusiones**

Después de haber realizado un análisis de la información recopilada sobre los diferentes sistemas de gestión de información y planificación y control de tareas, se puede concluir que ningún sistema de los estudiados se adecua a las características requeridas por los objetivos del sistema propuesto.

Por lo que se definió la creación de un sistema de Control y Planificación de las Tareas de los Profesores de la Facultad 7 que permita un mejor desarrollo de la facultad en cuanto a la gestión de los datos de los profesores. Como tecnología, las arquitecturas multicapas proporcionan una gran cantidad de beneficios en soluciones flexibles y fiables para resolver complejos problemas inmersos en cambios constantes además permiten a los componentes de negocio correr en una LAN, WAN o Internet.

Esto significa que cualquiera con un ordenador y conexión a la red posee toda la funcionalidad que tendría si se encontrase delante de su sistema de escritorio, de ahí que se utilicen estas características para el desarrollo de la aplicación. Con las características expuestas sobre servicios Web se puede concluir que son parte de la revolución informática de la nueva generación de aplicaciones que trabajan en colaboración y el objetivo principal que se logra con el uso de mismos es la interoperabilidad y la integración, pues permiten que las aplicaciones compartan información e invoquen funciones de otros sistemas.

Por lo que el sistema se implementa a través de una aplicación web que está soportada en la plataforma .NET utilizando el Visual Studio como herramienta de

## FUNDAMENTACIÓN TEÓRICA

diseño debido a que es una completa herramienta para generar e integrar con rapidez aplicaciones web. Así como, el SQL Server 2000 como gestor de base de datos, es el que más posibilidades brinda por sus características propias, que lo hacen ideal para sistemas como el que se propone desarrollar. Pues es capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. La metodología puesta en práctica es el RUP que utiliza el UML como lenguaje de representación visual a través de la herramienta case Rational Rose.



## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Definir las características del sistema es uno de los aspectos más importantes a tener en cuenta para el desarrollo y elaboración del software, para ello se partió de la modelación del negocio, se describieron requisitos funcionales y no funcionales que contribuyeron a la modelación del sistema. También se definió el sistema propuesto logrando delimitar correctamente todas las funcionalidades requeridas por el cliente.

### 2.0 Modelo de Negocio

En la Universidad de las Ciencias Informáticas (UCI) hoy por hoy no se utiliza ningún sistema informatizado que permita la gestión de información de profesores a nivel de facultad, de forma que este, haga eficiente la labor de todos aquellos involucrados de alguna manera en gestionar dicha información.

La comprensión de la estructura y la dinámica de la facultad, permitió identificar los problemas existentes en los procesos mencionados con anterioridad, mejoras viables que constituyeron requerimientos que debe cumplir el sistema a desarrollarse para que este satisfaga al usuario final.

En el desarrollo de este capítulo se realiza el modelo de negocio, el que está compuesto por el diagrama de caso de uso del negocio, el modelo de objetos del negocio y la descripción de los casos de usos implicados en el negocio así como los diagramas de actividades.

#### 2.0.1 Actores y trabajadores del negocio

Actores del negocio	Justificación
Directivos	Se benefician que haya un proceso de planificación y control del trabajo de los profesores y alumnos ayudantes. Estos pueden ser: decanos, Jefes de Departamento y Vice-Decanos.
Jefe de Departamento	Es el encargado de orientarle a la secretaria la planificación de las actividades y llevar el control de la asistencia de los profesores a las actividades.
Profesor	Se beneficia que haya un proceso de contratación en la facultad. Donde al incorporarse la secretaria le toma sus

	datos personales y profesionales.
Alumno Ayudante	Es un estudiante que imparte docencia en una asignatura específica, se beneficia que haya un proceso de captación por parte de los colectivos de asignatura.

**Tabla 1 Actores del negocio**

<b>Trabajadores del Negocio</b>	<b>Justificación</b>
Secretaria	Es la encargada de llenar las planillas de datos de profesores y alumnos Ayudantes. También de la planificación de las actividades orientada por el jefe de departamento.
Controlador de Asistencia	Es el encargado de verificar la asistencia y darle al jefe de departamento el listado de asistencia con las posibles causas de las ausencias.

**Tabla 2 Trabajadores del negocio**

## **2.0.2 Procesos de Negocio**

Proceso de captación de profesores: Este proceso se inicia a partir de la incorporación de un profesor a la facultad, la secretaria es la responsable de capturar los datos personales del profesor y registrarlos en una planilla específica para eso. Actualmente este proceso se realiza de forma manual lo que impide una correcta manipulación de la información relacionada con los profesores, por parte de todos aquellos directivos e interesados que necesiten cualquier información relacionada con los profesores.

Después de realizado este proceso el profesor es incorporado a un departamento donde inmediatamente se le asigna la asignatura que va a impartir y el grupo o los grupos en los cuales trabajará, además le es asignado un conjunto de actividades o tareas, reflejadas en su plan de trabajo individual, entre las que se pudieran mencionar:

- Participar en las actividades del departamento y colectivo de la asignatura al que pertenezca.
- Impartir la asignatura cumpliendo con el P1 y horario de la asignatura.
- Brindar las consultas necesarias a los estudiantes.
- Entregar en tiempo los resultados de los cortes evaluativos u otros resultados del proceso docente.

- Tener una buena asistencia a la guardia obrera durante el curso.
- Cumplir con lo establecido en el reglamento de seguridad informática.
- Participar en la mayoría de las actividades políticas, culturales, deportivas, productivas y recreativas organizadas por la FEU y UJC del centro, el sindicato, el departamento y el colectivo de año y contribuir activamente a su desarrollo.
- Participación en grupos de investigación – producción.
- Presentar trabajos en los diferentes eventos de Ciencia y Técnica a nivel de facultad o central.
- Participar en los festivales culturales y deportivos, directa o indirectamente apoyando a la representación de la facultad.
- Participar en los matutinos de la facultad.
- Mantener el pago de la MTT y sindicato actualizado.
- Participar como miembro del proyecto científico o productivo cumpliendo todas las tareas que le sean asignadas.

Proceso de captación de alumno ayudante: Este proceso se inicia a partir de la captación de un estudiante para que ejerza como alumno ayudante. Termina cuando la secretaria docente a partir de la propuesta de los estudiantes aprobados para desempeñarse como alumno ayudante procede a elaborar la resolución donde se refleja finalmente los estudiantes que son seleccionados enviándose para conocimiento al jefe de departamento y a los directivos.

Proceso de elaboración de plan de trabajo: La elaboración del plan de trabajo se realiza a los ALUMNO AYUDANTE y los profesores, los jefes de departamentos son los encargados de orientarles a las secretarías una serie de indicaciones para que las mismas lo realicen y darle el visto bueno al final.

Proceso de evaluación de actividades del plan de trabajo: Este proceso se inicia cuando el profesor le entrega al jefe de departamento una autoevaluación y después de un riguroso análisis se asigna una evaluación al mismo.

Proceso de búsqueda de información: Este proceso comienza cuando un interesado necesite saber alguna información de cierto profesor o alumno ayudante y culmina cuando la secretaria le muestra los datos.

Debido al cúmulo de trabajo de los jefes de departamento y la carencia de una fuente centralizada que recopile un informe de todos los profesores de la facultad, se resolvió realizar la implementación de un Sistema de Control y Planificación de las Tareas de Profesores de la Facultad 7, supliendo el trabajo manual por el informatizado y que

ofrezca información referente al profesor en el instante deseado tanto a los propios profesores como a los dirigentes de la facultad permitiendo a los jefes de departamento un alto uso de la información que se recopile en el sistema.

## 2.0.3 Diagrama de Casos de Uso del Negocio

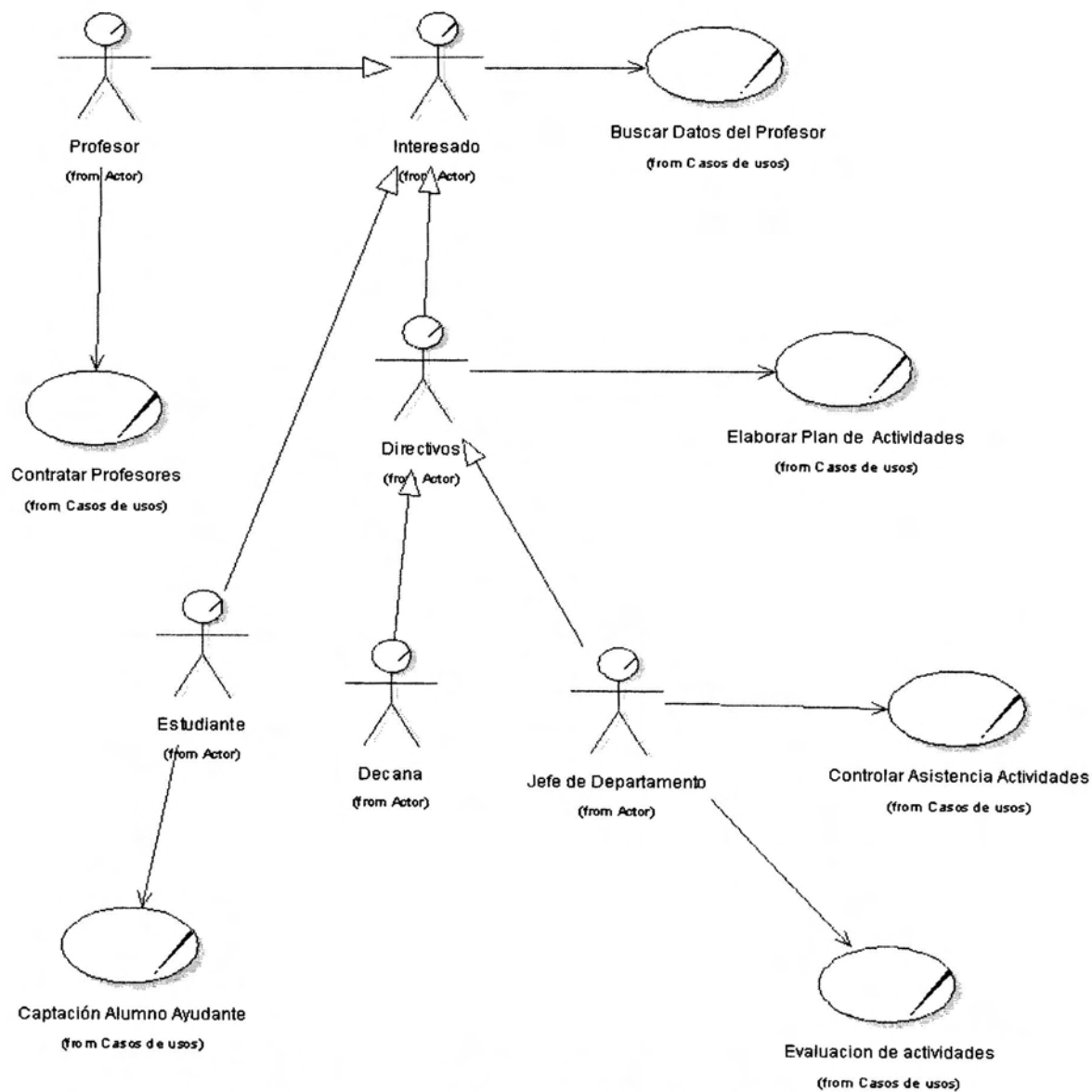


Figura 2 Diagrama de Casos de Uso del Negocio

## 2.0.4 Descripción textual de los Casos de Uso de Negocio

<b>Caso de Uso:</b>	Contratar Profesores	
<b>Actores:</b>	Profesor (inicia)	
<b>Trabajadores:</b>	Secretaria	
<b>Resumen:</b>	El caso de uso se inicia cuando el Profesor llega a la facultad, le dice sus datos a la secretaria, la cual llena la Planilla de Datos del Profesor, finalizando así el caso de uso.	
<b>Precondiciones:</b>		
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>	
1- El profesor se presenta en la facultad. 1.2-Le dice sus datos a la secretaria.	1.1 -La secretaria pide los datos pertinentes.  1.3--La Secretaria registra la información en la Planilla de Datos del Profesor.  1.4-La secretaria archiva los datos.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>	
<b>Poscondiciones</b>	El profesor queda registrado.	

Tabla 3 Descripción textual del Casos de Uso Contratar Profesores

<b>Caso de Uso:</b>	Controlar Asistencia Actividades	
<b>Actores:</b>	Jefe de Departamento (inicia)	
<b>Trabajadores:</b>	Controlador de Asistencia	
<b>Resumen:</b>	El caso de uso se inicia cuando el jefe de departamento le solicita al controlador de asistencia que controle la asistencia a las diferentes actividades que debe realizar el profesor.	
<b>Precondiciones:</b>		
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>	
1. El Jefe de Departamento le orienta al Controlador de Asistencia que tiene que controlar la asistencia de los profesores a las actividades.  1.3 El Jefe de Departamento recibe informe.	1.1-Hace un Listado de Asistencia que incluye la justificación de los ausentes.  1.2 Le informa la asistencia al Jefe de Departamento.	
<b>Poscondiciones</b>	La asistencia a las actividades queda archivada.	

Tabla 4 Descripción textual del Casos de Uso Controlar Asistencia Actividades

<b>Caso de Uso:</b>	Elaborar Plan de Actividades	
<b>Actores:</b>	Directivos (inicia)	
<b>Trabajadores:</b>	Secretaria	
<b>Resumen:</b>	El caso de uso se inicia cuando los Directivos le entregan a la Secretaria las orientaciones de las actividades planificadas para que esta le busque un local y las envíe a los profesores.	
<b>Precondiciones:</b>		
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>	
1. Los Directivos entregan las orientaciones de la planificación de actividades.	1.1 La secretaria recibe las orientaciones de la planificación de las actividades.  1.2 La secretaria elabora el plan de actividades de la facultad.  1.3 Envía plan de actividades a todos los profesores.	
<b>Poscondiciones</b>	Plan de actividades elaborado.	

Tabla 5 Descripción textual del Casos de Uso Elaborar Plan de Actividades

<b>Caso de Uso:</b>	Buscar Datos del Profesor	
<b>Actores:</b>	Usuarios (inicia)	
<b>Trabajadores:</b>	Secretaria	
<b>Resumen:</b>	El caso de uso se inicia cuando el Usuario solicita a la Secretaria buscar de los datos de un profesor.	
<b>Precondiciones:</b>		
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>	
11. El Usuario le solicita a la Secretaria buscar los datos de un profesor.	1.1 La secretaria hace la búsqueda en las planillas de datos de profesores.  1.2 Informa los datos del profesor al usuario.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>	
Actividad 1.3. Si no existe la plantilla del profesor buscado, se lo informa al usuario y este se retira.		
<b>Poscondiciones</b>		

Tabla 6 Descripción textual del Casos de Uso Buscar Datos del Profesor

<b>Caso de Uso:</b>	Captación Alumno Ayudante
<b>Actores:</b>	Alumno Ayudante (inicia)
<b>Trabajadores:</b>	Secretaria
<b>Resumen:</b>	El caso de uso se inicia cuando el estudiante es oficialmente Alumno Ayudante y le da los datos a la secretaria para que llene la Planilla de Datos de Alumno Ayudante.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
1. El Alumno Ayudante se presenta ante la secretaria para que esta lo registre.	1.1 La secretaria registra la información en la planilla de datos del Alumno Ayudante.  1.2 La secretaria archiva los datos.
<b>Flujos Alternos</b>	
<b>Poscondiciones</b>	1.3 Los datos quedan archivados.

**Tabla 7** Descripción textual del Casos de Uso Captación Alumno Ayudante

**2.0.5 Diagrama de actividades**

El diagrama de actividad es un grafo que contiene estados en el que puede hallarse una actividad.



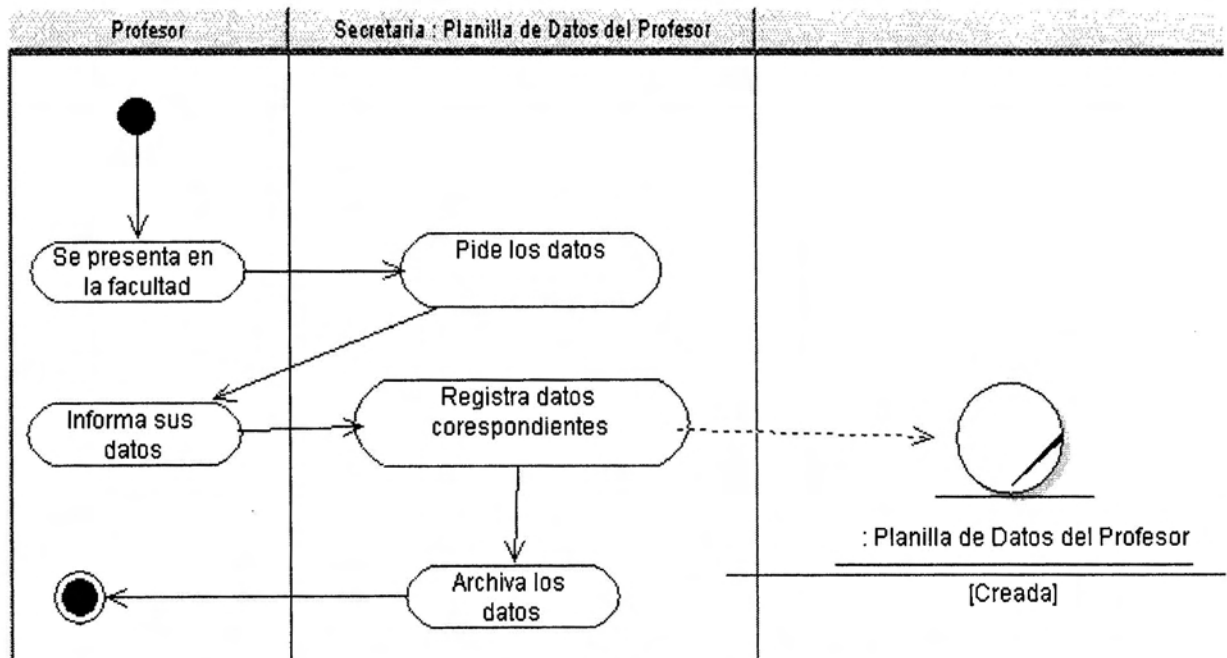


Figura 3 Diagrama de Actividades Caso de Uso: Contratar Profesor

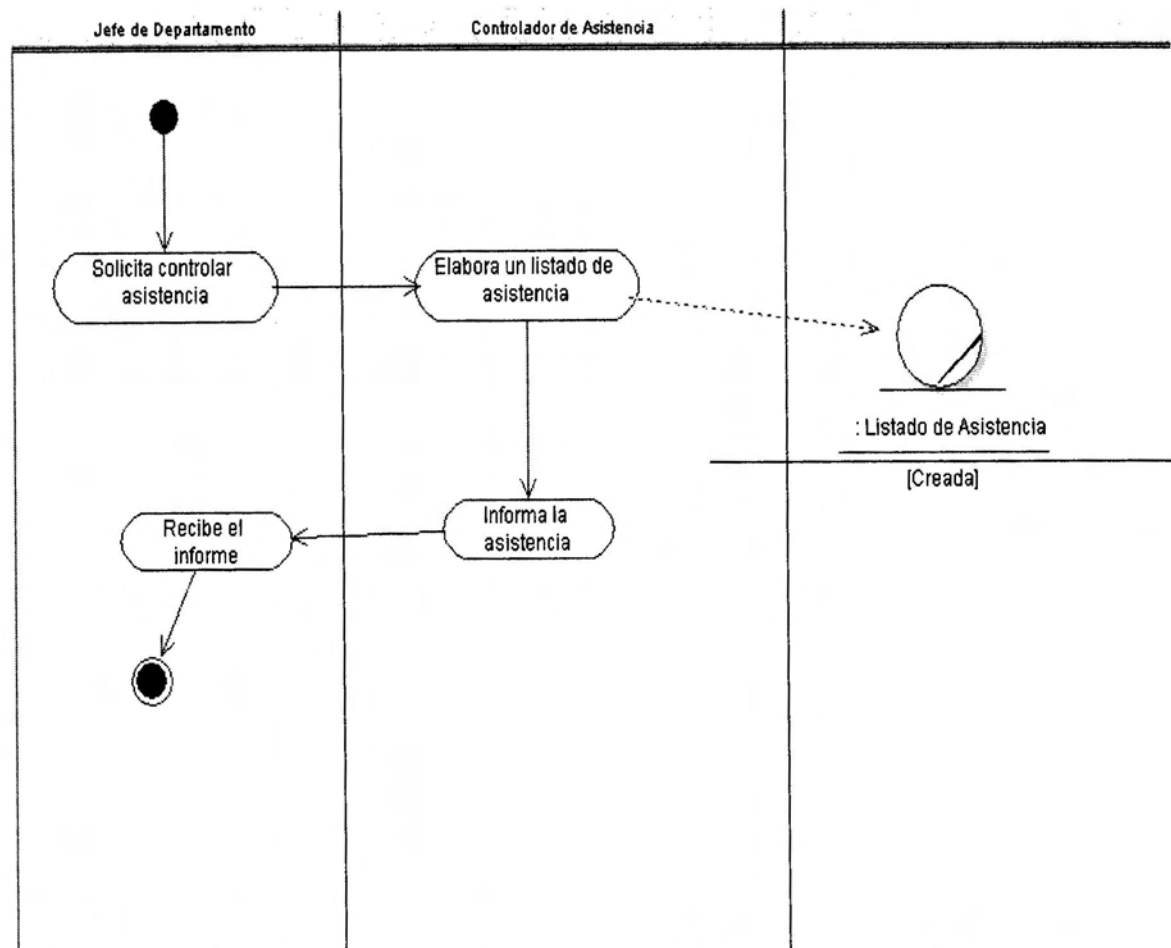


Figura 4 Diagrama de Actividades Caso de Uso: Controlar Asistencia Actividades

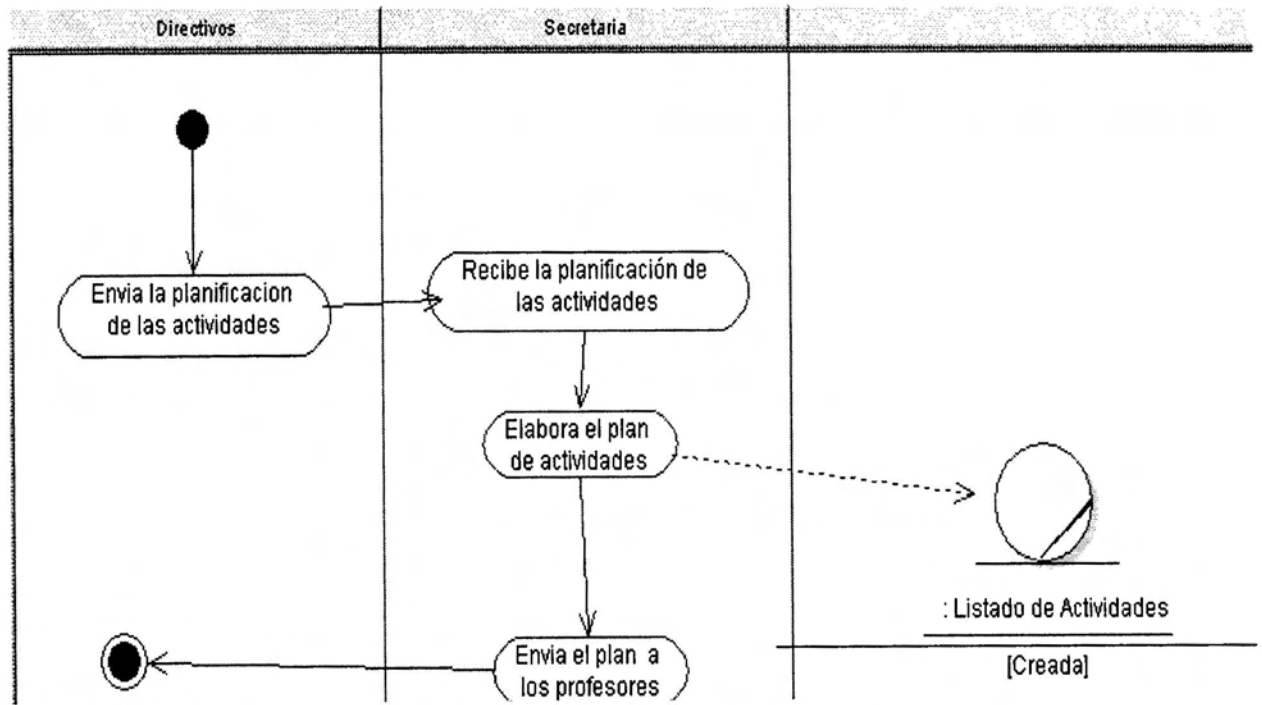


Figura 5 Diagrama de Actividades Caso de Uso: Elaborar Plan de Actividades

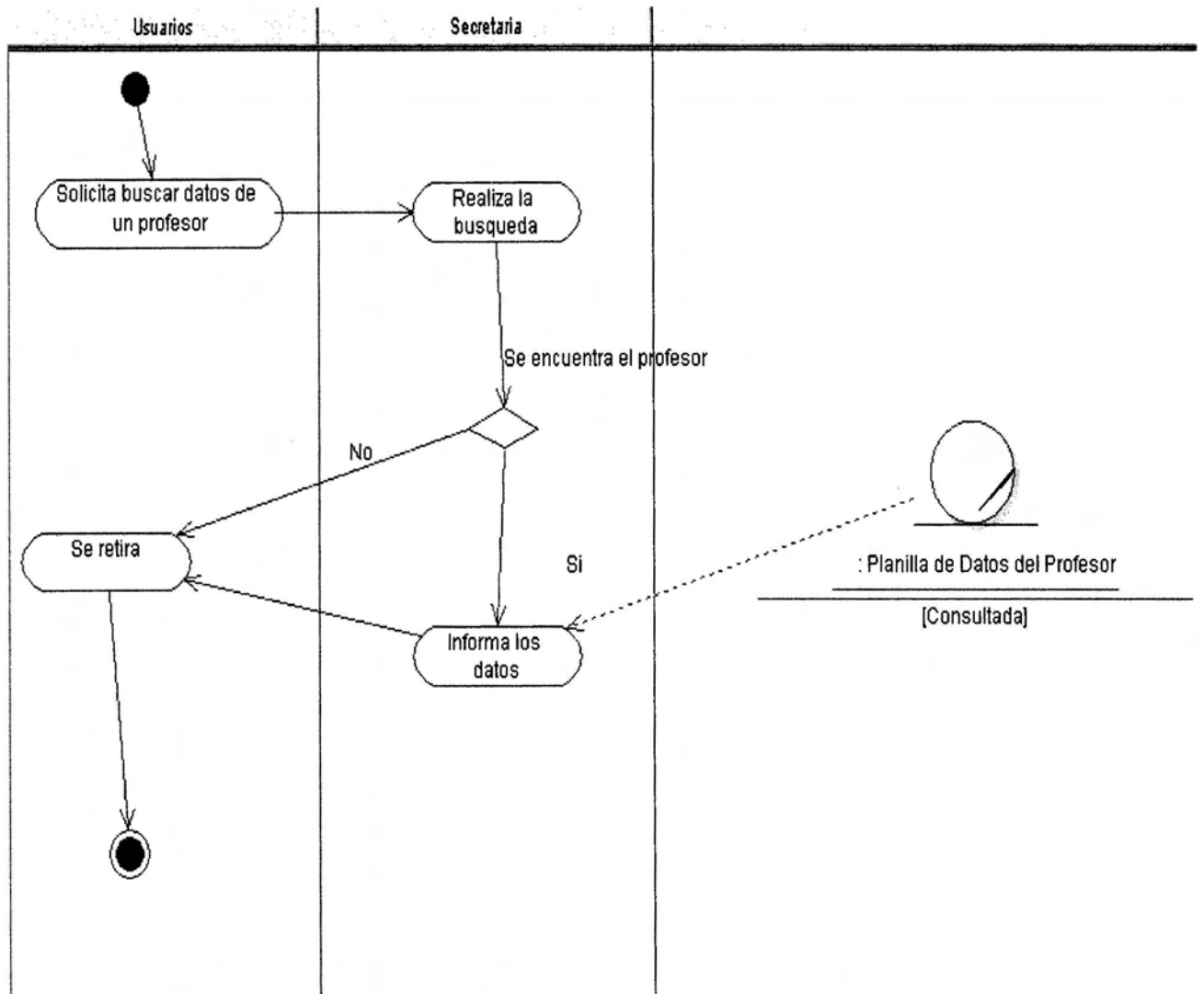
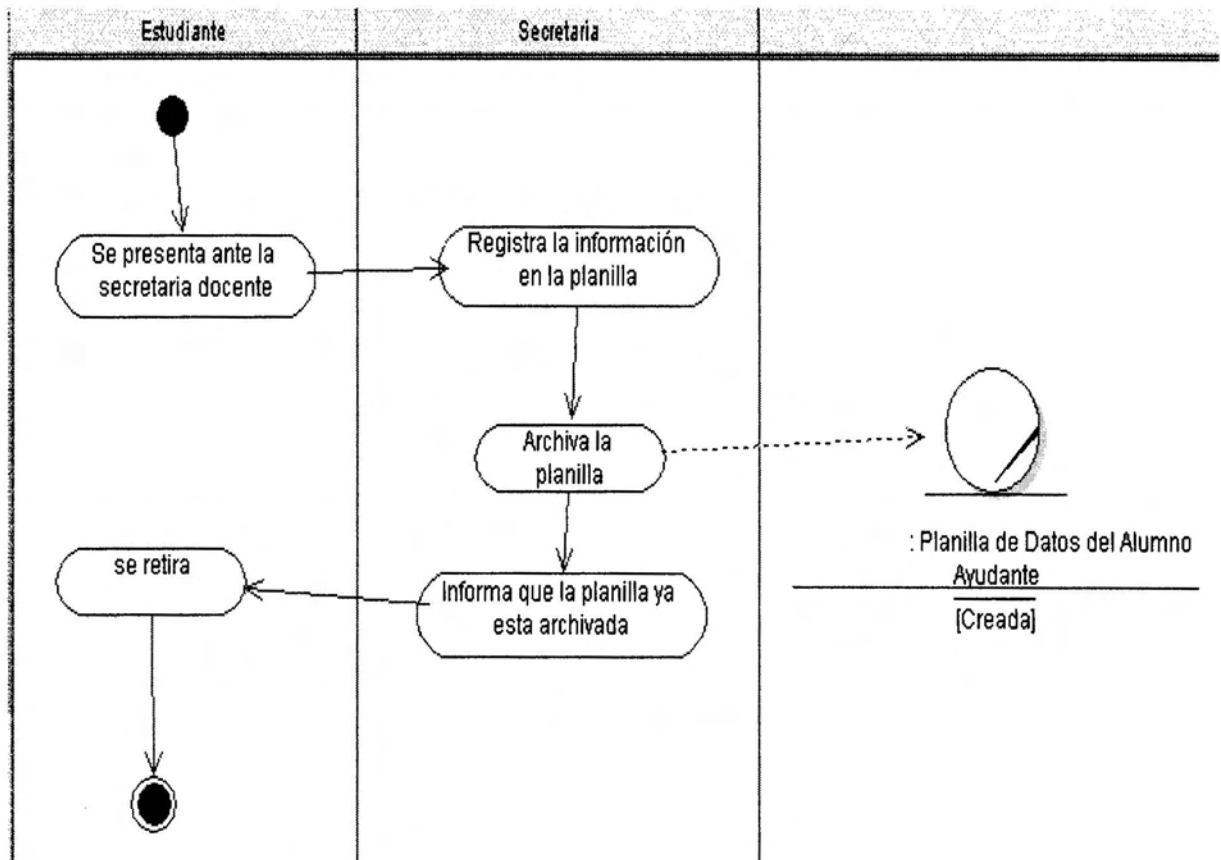


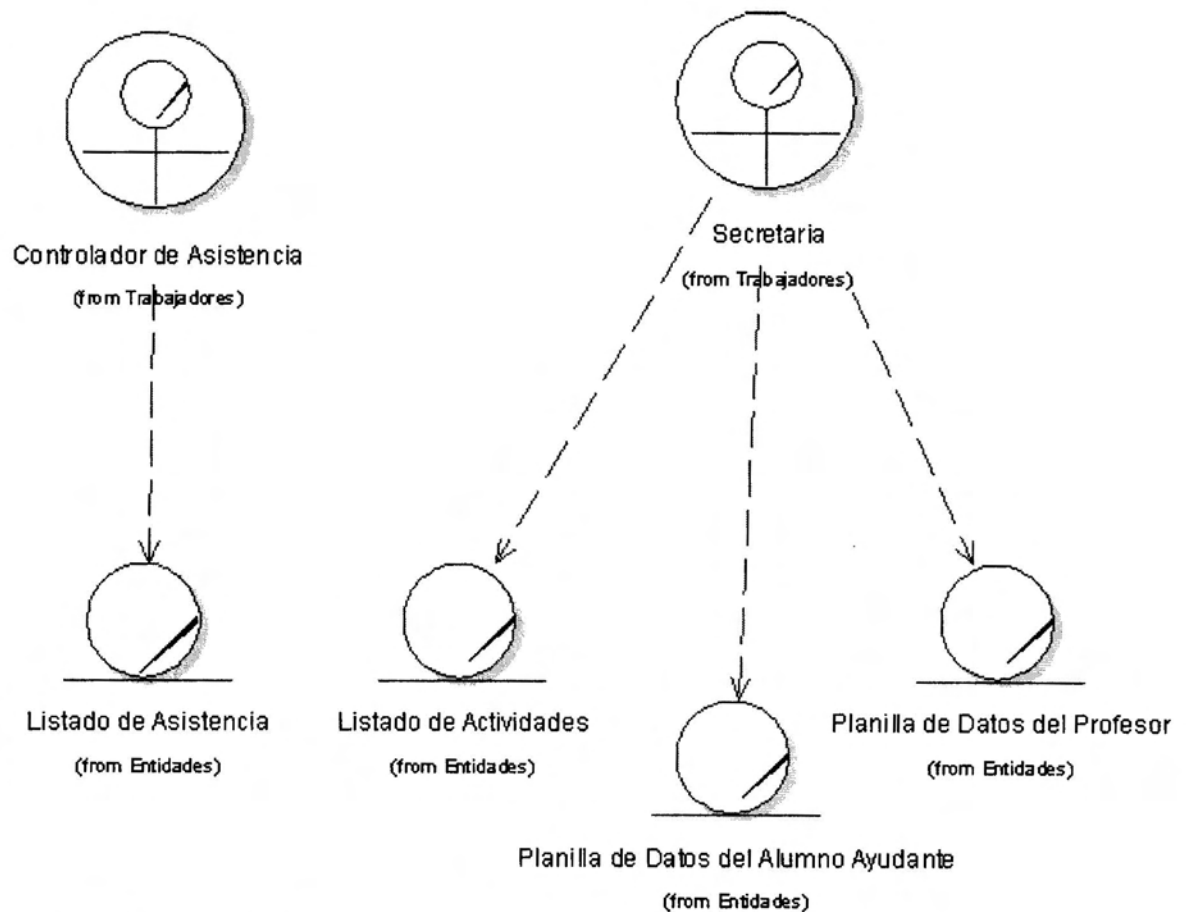
Figura 6 Diagrama de Actividades Caso de Uso: Buscar Datos de los Profesores



**Figura 7** Diagrama de Actividades Caso de Uso: Captación Alumno Ayudante

### 2.0.6 Modelo de Objetos

El modelo de objetos del negocio, muestra la participación de los trabajadores y entidades del negocio y la relación que se establece entre ellos. Describe cómo cada caso de uso del negocio es llevado a cabo por un conjunto de trabajadores que utilizan un conjunto de entidades del negocio.



**Figura 8** Diagrama de Objetos

### 2.1 Requisitos o Requerimientos.

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

#### 2.1.1 Requerimientos Funcionales.

Los nomencladores son los encargados de gestionar una serie de atributos que son imprescindibles cuando se gestiona información de los profesores y alumnos ayudantes, a continuación se mencionan algunos: Cargos, Tipo de contrato, Categoría docente, Títulos universitarios, Grado científico, Provincia, Municipio, Edificio, Apartamento, Grupo docentes, Polo científico, Área temática.

## RF1 Gestionar Datos Profesor.

- 1.1 Registrar Datos Profesor.
- 1.2 Modificar Datos Profesor.
- 1.3 Eliminar Datos Profesor.

## RF2 Gestionar Asistencia Actividades.

- 2.1 Registrar Asistencia Actividades.
- 2.2 Eliminar Asistencia Actividades
- 2.3 Modificar Asistencia Actividades.

## RF3 Gestionar Actividades.

- 3.1 Registrar Actividades.
- 3.2 Modificar Actividades.
- 3.3 Eliminar Actividades.

## RF4 Gestionar Usuario

- 4.1 Registrar Usuario.
- 4.2 Modificar Usuario.
- 4.3 Eliminar Usuario.

## RF5 Buscar Información

- 5.1 Buscar y Mostrar Profesor.
- 5.2 Buscar y Mostrar Actividad.
- 5.3 Buscar y Mostrar Asistencia Actividad.
- 5.4 Buscar y Mostrar Datos Alumno Ayudante.

## RF6 Gestionar Datos Alumno Ayudante.

- 6.1 Registrar Datos Alumno Ayudante.
- 6.2 Modificar Datos Alumno Ayudante.
- 6.3 Eliminar Datos Alumno Ayudante.

## RF7 Autenticar Usuario

## RF8 Crear Reportes

8.1 Reporte de Evaluación General del Departamento.

## RF9 Gestionar Privilegio de Usuario

9.1 Registrar Privilegio de Usuario.

9.2 Modificar Privilegio de Usuario.

9.3 Eliminar Privilegio de Usuario.

## RF10 Gestionar Nomencladores

10.1 Registrar Nomencladores.

10.2 Modificar Nomencladores.

10.3 Eliminar Nomencladores.

## RF11 Gestionar Asistencia

11.1 Registrar Asistencia.

11.2 Eliminar Asistencia.

### **2.1.2 Requerimientos no funcionales de:**

#### **Apariencia o interfaz externa**

- ✓ La interfaz debe ser sencilla, agradable y fácil de usar por los usuarios.
- ✓ Debe mostrar lo más claramente posible todas las opciones con las que cuenta el sistema con el objetivo de agilizar todo el proceso.

#### **Usabilidad**

Para utilizar la aplicación web es necesario estar registrado en el sistema y en dependencia del rol que posea, tendrá acceso o no, a la información solicitada.

#### **Rendimiento**

Se debe garantizar que la respuesta a solicitudes de los usuarios del sistema sea en un tiempo lo más breve posible.

#### **Seguridad**

- ✓ Se les garantizará el acceso a la información a los usuarios autorizados y restringiendo el acceso a los no autorizados. sin que los mecanismos de seguridad oculten o retrasen a los usuarios la obtención de sus datos en el momento deseado
- ✓ Mantener confidencialidad la información.

### **Implementación**

Será implementada haciendo uso de la tecnología .Net y SQL Server 2000 como Gestor de Base de Datos.

### **Software**

El sistema operativo será Windows XP o superior en la computadora cliente, en la servidora web y el servidor SQL Server 2000 será Windows NT 2003.

### **Hardware**

#### *Clientes:*

Pentium de 133 MHz o superior.

128 MB de memoria RAM mínima, 256 MB de memoria RAM

#### *Servidor Base Dato (SQL)*

Dual PIII de 933MHz o superior.

1 GB de memoria RAM.

100 GB de disco duro

#### *Servidor Web*

Dual PIII de 933MHz o superior

512 GB de memoria RAM.

30 GB de disco duro.

Al definir los requisitos del hardware en que el sistema, se deben basar en que será usado por lo menos 100 personas, también en la cantidad de consultas que se podrían realizar al mismo tiempo que no podrían sobrecargar al servidor propuesto, por la parte web el servidor con esa características será capaz de construir las paginas que sean necesarias y dar todos los servicios correspondientes sin ningún problema, para la computadora cliente con la propuesta hecha basta, pues solo se necesita para acceder a la aplicación un navegador web.



## 2.2 Descripción del Sistema Propuesto

### 2.2.1 Descripción de los actores del sistema

Actores del Sistema	Justificación
Secretaria	Este actor maneja un enorme volumen de información gestionando los datos de los profesores (nombre, dirección, cargos número de teléfono, solapín, CI, asignatura entre otros) y alumnos ayudantes (nombre, grupo, solapín, asignatura, grupo etc.).
Controlador de Asistencia	Este actor es el encargado de ir donde se están realizando las actividades y verificar la asistencia y registrar el listado de asistencia con las posibles causas de las ausencias en el sistema.
Administrador	Es el mayor beneficiado pues va a gestionar los usuarios es decir que va a interactuar con la aplicación agregando, eliminando y modificando usuarios.
Usuario	Es la generalización de Secretaria, Controlador de Asistencia, Administrador, Profesores, Directivos, Decana, Vicedecanos.
Profesores	Son los que van a buscar información (datos de los profesores, planificación de actividades).
Directivos	Tienen estar consultando constantemente un volumen de información muy grande sobre profesores, actividades y alumnos ayudantes, por otra parte van a gestionar el plan de actividades modificando, eliminando o insertando una actividad.

Tabla 8 Descripción de los actores del sistema

2.2.2 Diagrama de caso de uso del sistema.

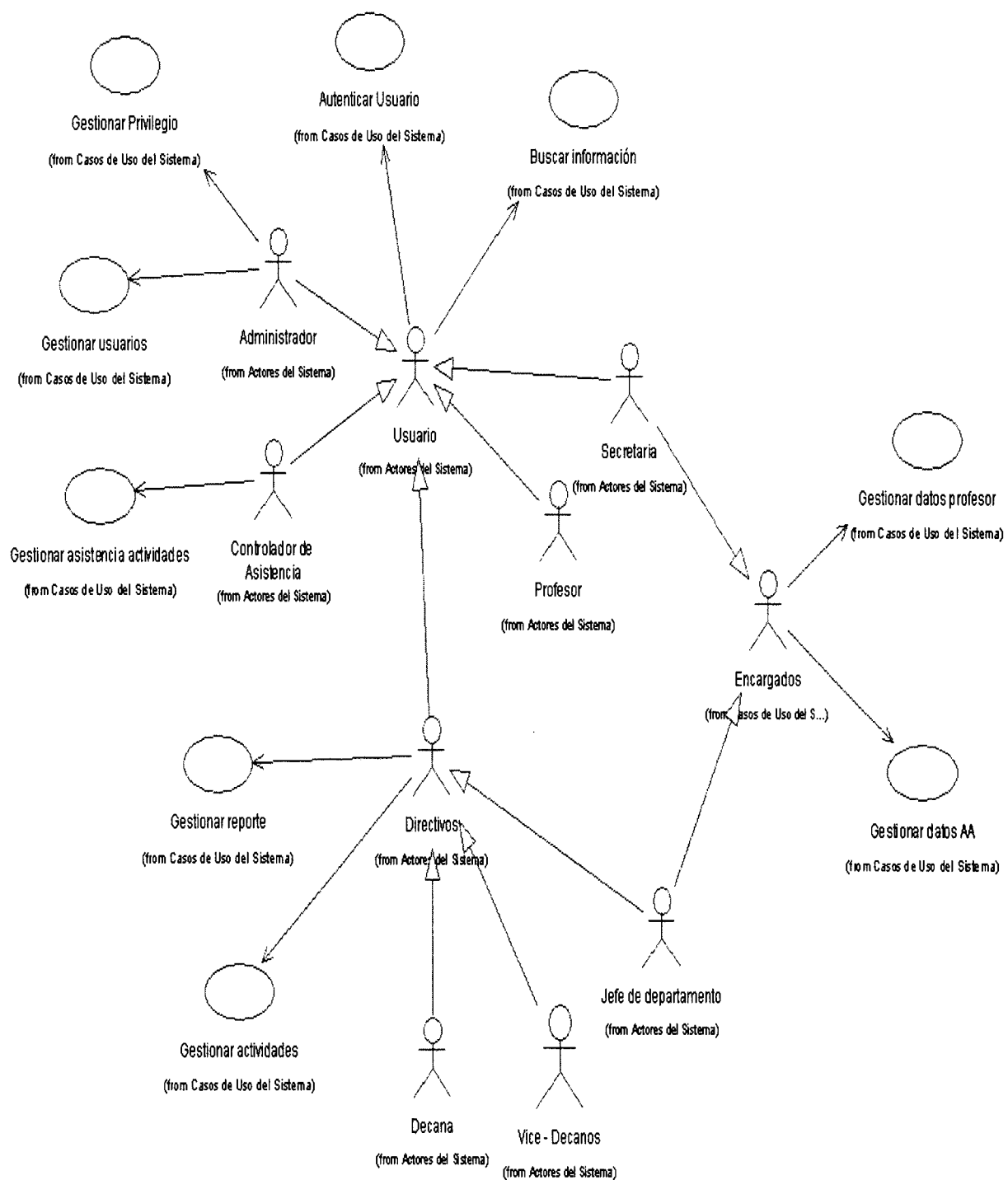


Figura 9 Diagrama de caso de uso del sistema

## 2.2.3 Descripciones de los Casos de Uso del Sistema

<b>Nombre del Caso de Uso</b>	<b>Gestionar Datos del Profesor</b>
<b>Actores</b>	Encargados(inicia)
<b>Propósito</b>	Permitir registrar, modificar o eliminar los datos del profesor.
<b>Resumen</b>	El caso de uso se inicia cuando los Encargados necesitan gestionar los datos de un profesor, para ello realiza las acciones "Registrar Datos del Profesor Adjunto" o " Registrar Datos del Profesor Interno ", "Modificar Datos del Profesor" o "Eliminar Profesor". El CU termina cuando el usuario fue insertado, modificado o eliminado.
<b>Referencias</b>	RF1
<b>Precondiciones</b>	Los encargados deben autenticarse. El profesor debe ser usuario del sistema.
<b>Poscondiciones</b>	Se registra o modifican los datos del profesor.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1- Los encargados solicita registrar o modificar sus datos.	1.1- El sistema ejecuta alguna de las siguientes acciones: a) Si decide registrar los datos, ir a la sección "Registrar Datos del Profesor Adjunto" o "Registrar Datos del Profesor Interno". b) Si decide actualizar alguno de los datos, ir a la sección "Modificar Datos del Profesor". b) Si decide eliminar algún profesor, ir a la sección "Eliminar Profesor".
<b>Sección "Registrar Datos del Profesor"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
2- Los encargados registra los datos del profesor.	2.1- El sistema verifica que todos los campos estén llenos. 2.2 - El sistema verifica que el profesor no exista. 2.3 -Los datos del profesor se almacenan en

	<p>el sistema.</p> <p>2.4 -Se muestra un mensaje informándosele a la secretaria que ya el profesor ha sido registrado.</p>
<b>Sección "Modificar Datos del Profesor"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	3.1- El sistema muestra todos los datos que el profesor tiene almacenado.
4- Los encargados realizan las modificaciones deseadas.	4.1 - Se actualiza la información en la BD del sistema.
<b>Sección "Eliminar Profesor"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1- Los encargados deciden eliminar a un profesor.	1.1-El sistema muestra un listado de todos los profesores registrados.
2- Los encargados escogen el profesor a eliminar.	2.1 -El sistema pide confirmación. 2.2 -El sistema elimina el profesor.
<b>Curso Alterno de los Eventos</b>	
2.1 -Si faltan datos obligatorios se presenta un mensaje que indica que elementos obligatorios le faltan.	
2.2 -Si el profesor ya existe en el sistema se presenta un mensaje.	
<b>Prioridad:</b>	Crítico
<b>Prototipo de Interfaz</b>	RF1

Tabla 9 Descripciones del Caso de Uso del Sistema Gestionar Datos del Profesor

<b>Nombre del Caso de Uso</b>	<b>Gestionar Asistencia Actividades</b>
<b>Actores</b>	Controlador de Asistencia (inicia)
<b>Propósito</b>	Permite registrar o eliminar las asistencias.
<b>Resumen</b>	El caso de uso se inicia cuando el Controlador de Asistencia decide gestionar la asistencia a las actividades, pudiendo registrar o eliminar la asistencia a las actividades.
<b>Referencias</b>	RF2
<b>Precondiciones</b>	El Controlador de Asistencia debe autenticarse.
<b>Poscondiciones</b>	Se registra o elimina la asistencia a una actividad.

<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1- El Controlador de Asistencia necesita registrar.	1.1- El sistema ejecuta alguna de las siguientes acciones: a) Registrar asistencia actividades, ir a la sección "Registrar Asistencia Actividades".
<b>Sección "Registrar Asistencia Actividades"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
2- El Controlador de Asistencia entra la asistencia a las actividades.	2.1- El sistema verifica que todos los campos estén llenos. 2.2- La asistencia de la actividad se almacenan en el sistema. 2.3 -Se muestra un mensaje informando al directivo que ya ha sido registrada la asistencia a la actividad.
<b>Curso Alternativo de los Eventos</b>	
2.1- Si faltan datos obligatorios se presenta un mensaje que indica que elementos obligatorios le faltan.	
<b>Prioridad:</b>	Crítico
<b>Prototipo de Interfaz</b>	RF2

**Tabla 10** Descripciones del Caso de Uso del Sistema Gestionar Asistencia Actividades

<b>Nombre del Caso de Uso</b>	<b>Buscar Información</b>
<b>Actores</b>	Usuario (inicia)
<b>Propósito</b>	Permite buscar cualquier información del profesor.
<b>Resumen</b>	El caso de uso se inicia cuando el usuario decide ver cualquier información, ya sea los datos del profesor, así como la asistencia a las actividades, además de las diferentes actividades planificadas para cada profesor.
<b>Referencias</b>	RF5
<b>Precondiciones</b>	El usuario debe autenticarse.
<b>Poscondiciones</b>	Se muestra la información deseada.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1- El usuario necesita buscar información	1.1- El sistema ejecuta alguna de las siguientes acciones: a) Si decide ver los datos de un profesor, ir a la sección "Buscar Profesor". b) Si decide ver las asistencias a las actividades, ir a la sección "Buscar Asistencia". c) Si decide ver las actividades, ir a la sección "Buscar Actividad".
<b>Sección "Buscar Datos Profesor"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
2- El usuario selecciona los campos por los que desea buscar a un profesor. 2.2- El usuario selecciona el profesor del cual desea ver sus datos.	2.1- El sistema muestra un listado de todos los profesores registrados con dichos datos. 2.3- El sistema muestra un listado de todos los datos de dicho profesor.
<b>Sección "Buscar Asistencia"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
3. El usuario selecciona la actividad de la cual quiere conocer la asistencia.	3.1. El sistema muestra un listado con los profesores que asistieron a dicha actividad.
<b>Sección "Buscar Datos Alumno Ayudante"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>

4- El usuario selecciona los datos por los cuales quiere buscar a un alumno ayudante.	4.1- El sistema muestra un listado de todos los alumnos ayudante registrados con dichos datos.
4.2- El usuario selecciona el alumno ayudante del cual desea ver sus datos.	4.3- El sistema muestra un listado de todos los datos de dicho profesor.
<b>Sección "Buscar Actividad"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
5. El usuario selecciona los datos por los cuales quiere buscar la actividad.	5.1. El sistema muestra un listado de todas las actividades registradas con dichos datos.
<b>Prioridad:</b>	Crítico
<b>Prototipo de Interfaz</b>	RF5

Tabla 11 Descripciones del Caso de Uso del Sistema Buscar Información

<b>Nombre del Caso de Uso</b>	<b>Gestionar Actividad</b>
<b>Actores</b>	Directivos(inicia)
<b>Propósito</b>	Permite registrar, modificar y eliminar actividades.
<b>Resumen</b>	El caso de uso se inicia cuando el Directivo decide gestionar actividad, pudiendo registrar, modificar o eliminar la actividad.
<b>Referencias</b>	RF3
<b>Precondiciones</b>	El directivo debe autenticarse.
<b>Poscondiciones</b>	Se registra, modifica o elimina una actividad.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1- El Directivo necesita registrar, modificar o eliminar una actividad.	1.1- El sistema ejecuta alguna de las siguientes acciones: a) Si decide registrar una actividad, ir a la sección "Registrar Actividad". b) Si decide modificar alguno de los datos, ir a la sección "Modificar Actividad". c) Si decide eliminar alguna actividad, ir a la sección

	"Eliminar Actividad".
<b>Sección "Registrar Actividad"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
2- El Directivo registra la actividad.	2.1 -El sistema verifica que todos los campos estén llenos. 2.2 -Los datos de la actividad se almacenan en el sistema. 2.3 -Se muestra un mensaje informándosele al Directivo que ya se ha registrado la actividad.
<b>Sección "Modificar Actividad"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
3- El Directivo selecciona el departamento al que pertenece la actividad que desea modificar.	3.1- El sistema muestra todas las actividades que están registradas en ese departamento.
4- El Directivo realiza las modificaciones deseadas.	4.1- El sistema le muestra un mensaje para confirmar si desea modificar la actividad.
5-El Directivo confirma que desea modificar la actividad.	5.1 - Se actualiza la información en la BD del sistema.
<b>Sección "Eliminar Actividad"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
6- El Directivo selecciona el departamento al que pertenece la actividad que desea eliminar.	6.1-El sistema muestra todas las actividades que tiene almacenada de ese departamento.
7-El Directivo selecciona la actividad deseada.	7.1- El sistema pide confirmación. 7.2 -El sistema elimina la actividad. 7.3 -Se actualiza la información en la BD del sistema.
<b>Curso Alternativo de los Eventos</b>	
2.1- Si faltan datos obligatorios se presenta un mensaje que indica que elementos obligatorios le faltan.	



<b>Prioridad:</b>	Crítico
<b>Prototipo de Interfaz</b>	RF3

Tabla 12 Descripciones del Caso de Uso del Sistema Gestionar Actividad

<b>Nombre del CU</b>	<b>Autenticarse</b>	
<b>Actores</b>	Usuario (inicia)	
<b>Propósito</b>	Permitir autenticarse.	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario introduce los datos que se le piden para acceder a la aplicación, estos se verifican y finaliza dándole acceso a la información que le corresponde.	
<b>Referencias</b>	RF7	
<b>Precondiciones</b>		
<b>Poscondiciones</b>	Se habilitan las funcionalidades según sus privilegios.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1. El usuario entra usuario y contraseña.	1.1 El sistema encripta la contraseña. 1.2 Busca el usuario y compara la contraseña. 1.3 Se le asignan los permisos.	
<b>Flujos alternativos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
	1.3 En caso de que no exista se le envía un aviso.	
<b>Prioridad:</b>	Crítico	
<b>Prototipo de Interfaz</b>	RF7	

Tabla 13 Descripciones del Caso de Uso del Sistema Autenticarse

Tabla 14 Descripciones del Caso de Uso del Sistema Gestionar Asistencia

<b>Nombre del Caso de Uso</b>	<b>Gestionar Privilegio de Usuario</b>
<b>Actores</b>	Administrador(inicia)
<b>Propósito</b>	Insertar un nuevo privilegio para que el usuario tenga diferentes accesos al sistema en dependencia del rol que posea el mismo, modificar privilegio o

	eliminar privilegio en caso de ser necesario.
<b>Resumen</b>	El caso de uso se inicia cuando el administrador accede a la interfaz de administrar sistema y selecciona la acción que desea realizar, ya sea insertar, modificar o eliminar un privilegio y termina el caso de uso con la ejecución de dicha acción.
<b>Precondiciones</b>	En caso de que se desea modificar o eliminar un privilegio determinado, es necesario que este se encuentre registrado en el sistema
<b>Poscondiciones</b>	Se inserta, modifica o elimina un privilegio.
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
<p>1- El administrador accede a la interfaz de administración.</p> <p>2-El administrador elige la acción a realizar.</p>	<p>1.1- El sistema muestra una serie de acciones a realizar.</p> <p>2.1-Si elige.</p> <p>a) Insertar un nuevo privilegio ir a la sección "Insertar Privilegio"</p> <p>b) Modificar un privilegio ir a la sección "Modificar Privilegio"</p> <p>c) Eliminar un privilegio ir a la sección "Eliminar Privilegio".</p>
<b>Sección "Insertar Privilegio"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
2- El administrador introduce los datos (nombre del privilegio, páginas de acceso).	<p>1- El sistema muestra la interfaz de insertar privilegio, mostrando un formulario con los campos generales que se deben introducir.</p> <p>2.1- El sistema verifica que no exista ese privilegio en el sistema.</p> <p>2.2-El sistema valida los datos introducidos.</p>

	2.3- Inserta el nuevo privilegio en el sistema
<b>Sección "Modificar Privilegio"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
<p>2- El administrador selecciona el privilegio que desea modificar.</p> <p>3-El administrador realiza los cambios pertinentes.</p>	<p>1.2-El sistema muestra el listado de los privilegios existentes en la base de datos.</p> <p>2.1- El sistema muestra la información del privilegio seleccionado.</p> <p>3.1 El sistema verifica que los campos obligatorios estén llenos.</p> <p>3.2- El sistema guarda los datos modificados en la base de datos y los actualiza.</p>
<b>Sección "Eliminar Privilegio"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
<p>2- El administrador elige el privilegio a eliminar.</p> <p>3- El administrador confirma la acción.</p>	<p>1. El sistema muestra el listado de los privilegios existentes.</p> <p>2.1- El sistema pide confirmación de que desea eliminar el privilegio.</p> <p>3.1- El sistema elimina el privilegio seleccionado.</p>
<b>Poscondiciones</b>	
Se registra un privilegio, se actualizan los datos.	
<b>Prioridad:</b>	Crítico
<b>Prototipo de Interfaz</b>	RF9

**Tabla 15** Descripciones del Caso de Uso del Sistema Gestionar Privilegio de Usuario

## **Conclusiones**

En este capítulo se realizó la modelación del negocio determinándose las características y actividades que se desea automatizar determinando las mejoras que puede aportar, al negocio del sistema a desarrollar.

Como resultado de esto se obtuvo, un diagrama de caso de uso del sistema, descripciones de los casos de uso del negocio y del sistema respectivamente, así como el modelo de objetos del negocio, y los diagramas de actividades por cada caso de uso.

Además se realizó la captura de los requerimientos funcionales y no funcionales a tener en cuenta para el desarrollo del sistema dejando plasmado una breve descripción de los requisitos funcionales críticas del sistema.

## CAPÍTULO 3 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

En este capítulo se muestran los resultados obtenidos en el desarrollo de los procesos de análisis y diseño del sistema, así como los diagramas que fueron necesarios utilizar para obtener una mayor claridad a la hora de elaborar la solución que se propone y para lograr una implementación acorde a las necesidades de los usuarios.

### 3.0 Diagramas de colaboración

Un Diagrama de Colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere). A diferencia de los Diagramas de Secuencia, los Diagramas de Colaboración muestran las relaciones entre los roles de los objetos. La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia.

También muestra cómo las instancias específicas de las clases trabajan juntas para conseguir un objetivo común. Implementa las asociaciones del diagrama de clases mediante el paso de mensajes de un objeto a otro. Dicha implementación es llamada "enlace". [Ver anexo](#)

### 3.1 Diagramas de clases de Diseño

El modelo de clases está en el núcleo del desarrollo y del diseño orientado a objetos; expresa el estado persistente y el comportamiento del sistema. Una clase encapsula el estado (los atributos) y ofrece los servicios para manipularlo (el comportamiento). Un buen diseño orientado a objetos limita el acceso directo a los atributos de la clase y ofrece los servicios que manipulan a deferencia del solicitante.

Este ocultamiento de los datos y exposición de los servicios asegura que las modificaciones de los datos se realizan sólo en un lugar y de acuerdo con reglas específicas; para grandes sistemas la cantidad de código que tiene acceso directo a los elementos de datos en muchos sitios es extremadamente alto.

**3.1.1 Subsistemas y Clases de Diseño**

Los subsistemas de diseño son un mecanismo de organización de los artefactos del modelo de diseño en piezas más manejables .pueden contra de clases del diseño, realización de caso de uso, interfaces y otros subsistemas. Deben ser:

- Cohesivos: sus contenidos deben estar fuertemente relacionados
- Débilmente acoplados: minimizar las dependencias entre subsistemas

Las clase de diseño deben ser la encapsulación completa de los atributos y métodos que se pueden esperar, en forma razonable, que existan para la clase, es decir que pueden contener los métodos que sean suficientes para lograr el objetivo de la clase.

**3.1.2 Elementos específicos de la arquitectura de un entorno web.**

Un fichero *script* interpretable con extensión ASP.NET contiene código que se ejecuta en el servidor y código que se ejecuta en el fichero de la máquina cliente. Jim Conallen estableció como parte de su extensión, que cada fichero script interpretable podría modelarse a partir de varias clases de UML, una clase representaría el código puramente servidor, una clase representaría el código puramente cliente, y una clase representaría los formularios presentes en el código cliente.

Así, su extensión presenta como elementos más significativos 3 clases de UML estereotipadas con los siguientes estereotipos "Server Page", "Client Page","Form" empleados para el código servidor, código cliente y formulario respectivamente.

Las relaciones posibles relaciones entre los tres elementos claves son:

Hasta Desde	Client Page	Form	Server Page
Client Page	<<Link>>, <<redirect>>	Contiene	<<Link>>, <<redirect>>
Form	Agregado por.	---	<<Submit>>
Server Page	<<Build>>, <<redirect>>		<<Redirect>>

Figura 10 Relaciones entre las clases principales que conforman la extensión de UML para Web

Las páginas servidoras se encargan de construir o generar el resultado HTML que conforma el código cliente (*build*). Los formularios envían sus datos a las paginas servidoras para procesar los pedidos (*submit*), y además forman parte del código cliente o resultado HTML; es por esto que la relación entre las clase empleada para el código cliente y la clase empleada para el formulario es de agregación. Entre páginas clientes pueden existir vínculos (*link*). Una página cliente podría contener varios formularios e incluso estos pueden enviar sus datos a distintas páginas servidoras encargadas de procesarlos.

# CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

En caso de la existencia de un fichero con código servidor, que contenga funciones para procesamiento, que no genera resultado HTML directo, o sea la página construye resultado HTML; entonces se modelaría una sola clase UML, la cual es la representación del fichero script interpretable final. [25]

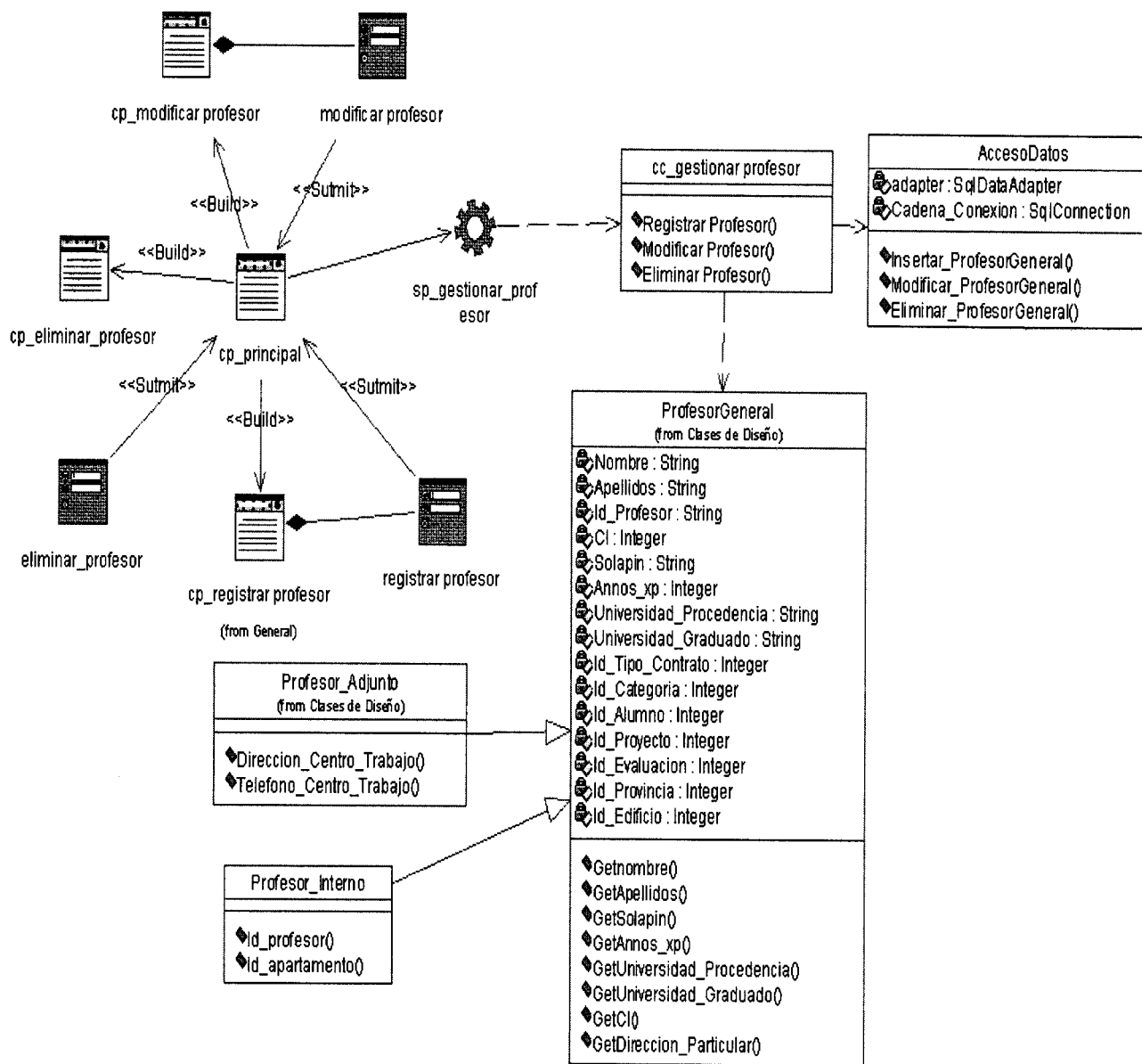


Figura 11 Diagrama de clase de diseño Gestionar Profesor

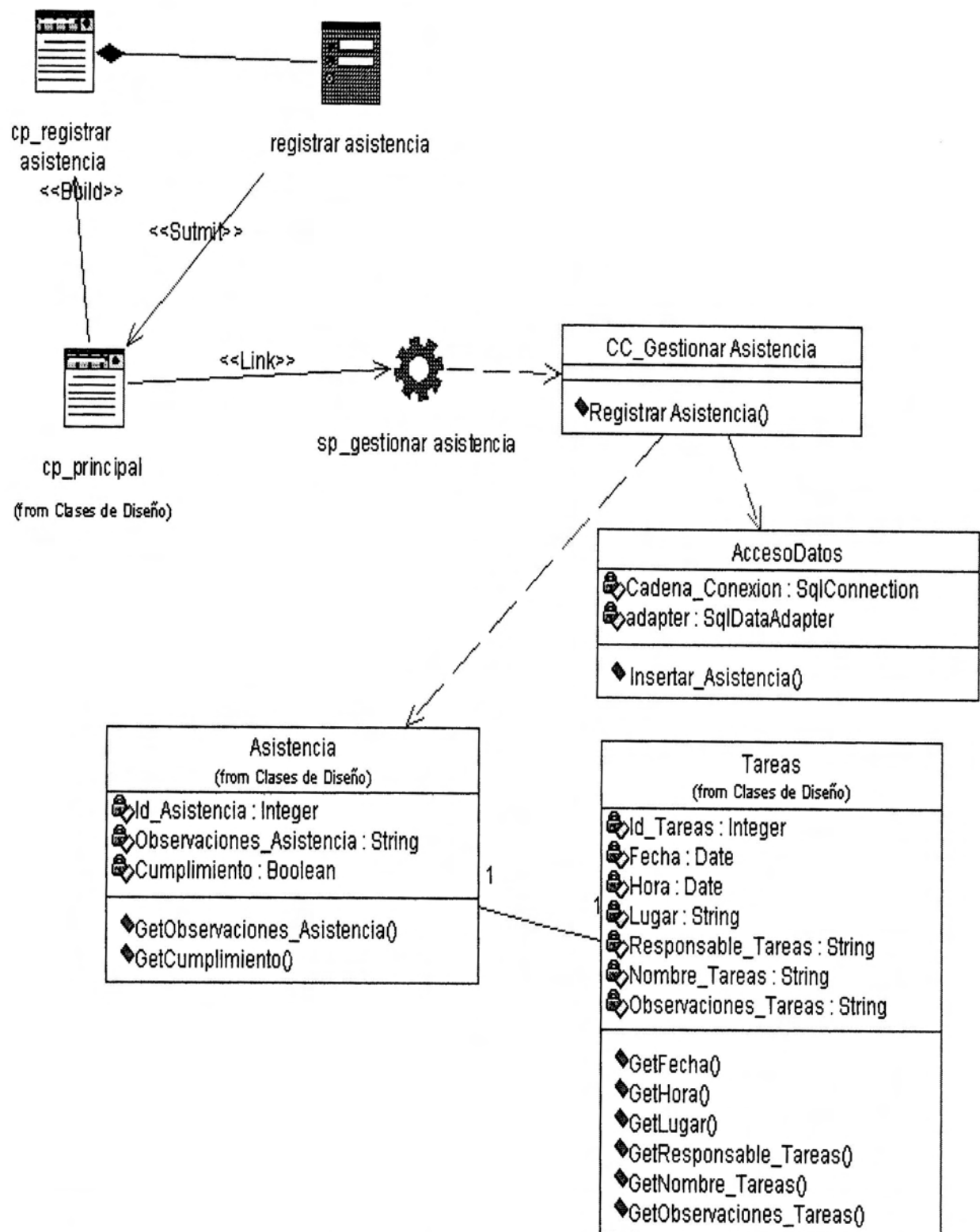


Figura 12 Diagrama de clase de diseño Registrar Asistencia



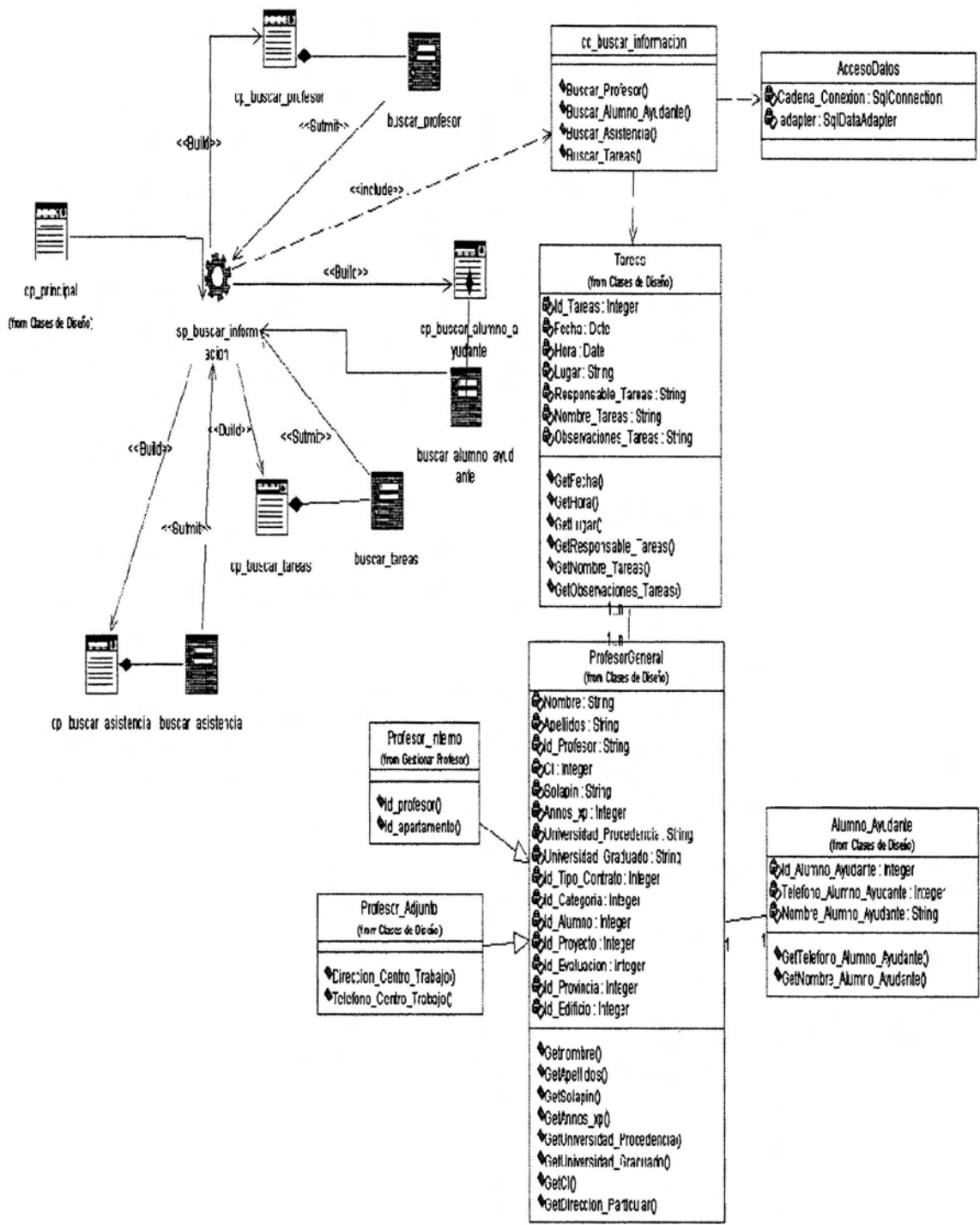


Figura 13 Diagrama de clase de diseño Buscar

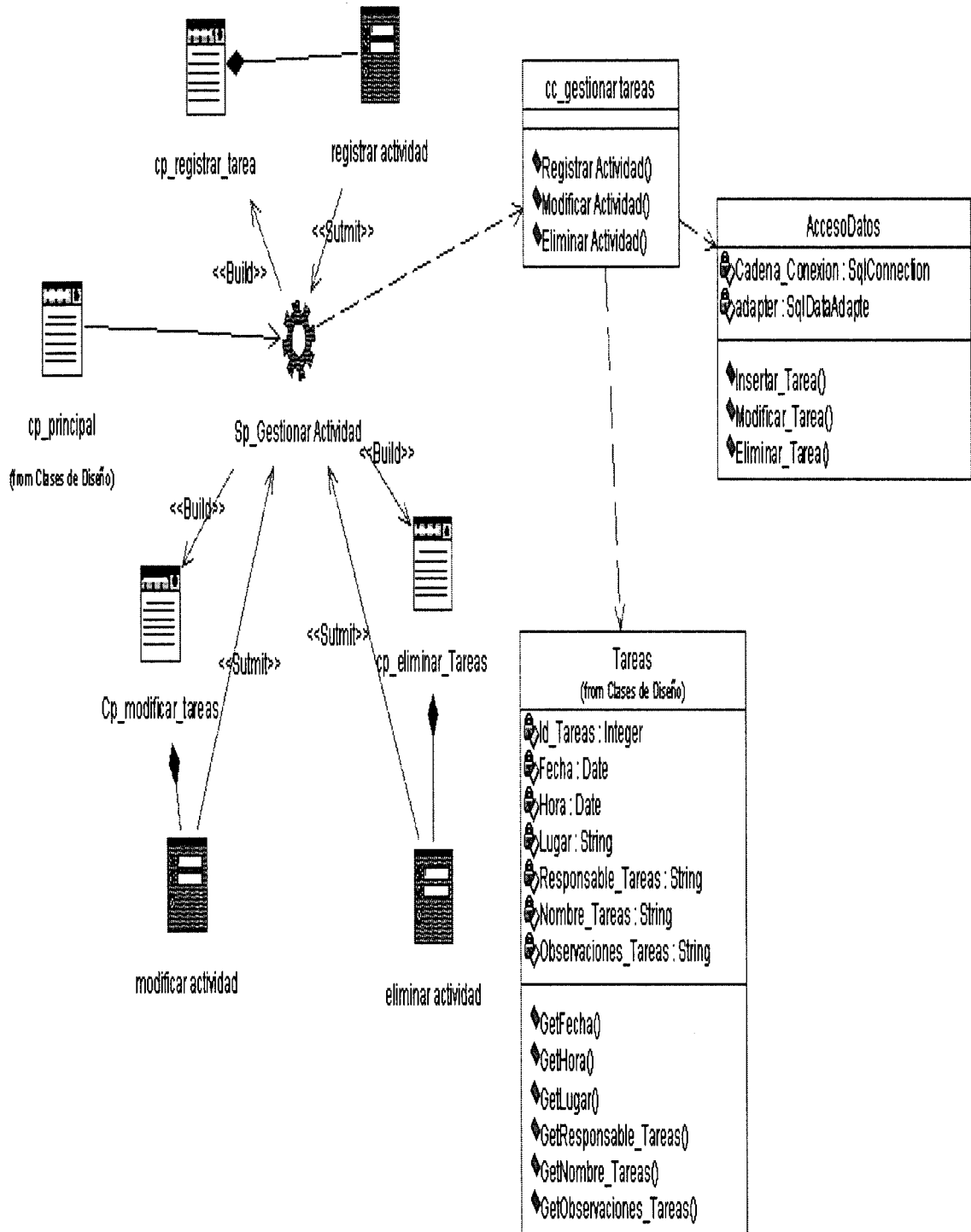


Figura 14 Diagrama de clase de diseño Gestionar Tareas

## 3.2 Descripción de las clases persistentes

Para facilitar el análisis, se clasifican las clases en:

### Clase de control

- Representan coordinación, secuencia, transacciones, y control de otros objetos.
- Se usan mucho para encapsular el control de un caso de uso concreto.
- También derivaciones y cálculos complejos.
- Manejan y coordinan las acciones y los flujos de control principales, y delegan trabajo a otros objetos (de interfaz y de entidad).

### Clase de entidad

- Modelan información que posee una larga vida.
- Modelan la información y el comportamiento asociado a algún fenómeno o concepto, como una persona, un objeto del mundo real, o un suceso del mundo real.
- Derivan normalmente de una clase de entidad del negocio.
- Puede tener comportamiento complejo.
- Aísla los cambios en la información que representa.

### Clase de interfaz

- Modelan la interacción entre el sistema y sus actores (usuarios y sistemas externos).
- Reciben y presentan información y peticiones de y hacia los actores.
- Reúnen los requisitos en los límites del sistema.
- Suelen ser abstracciones de ventanas, formularios, interfaces de impresoras, sensores, terminales.
- Describan lo que se obtiene con la interacción, no el proceso físico de cómo se ejecuta.
- Una clase de interfaz por cada actor, y viceversa.

## CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

A continuación aparecen descritas las principales clases persistentes teniendo en cuenta sus atributos y métodos las cuales permiten tener una visión general de los atributos y métodos de la aplicación propuesta.

### Principales Clases entidades

<b>Nombre: ProfesorGeneral</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
id_profesor	int
solapin	int
annos_xp	int
nombre	string
apellidos	string
universidad_procedencia	string
universidad_graduado	string
id_proyecto	int
direccion	string
id_tipo_contrato	int
id_categoria	int
ci	Int64
id_evaluacion	int
id_municipio	int
id_cargo	int
id_grupodocente	int
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	ProfesorGeneral
<b>Descripción:</b>	Constructor de la clase.
<b>Nombre:</b>	Id_cargo
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_cargo.
<b>Nombre:</b>	Id_evaluacion
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_evaluacion.
<b>Nombre:</b>	Apellidos
<b>Descripción:</b>	Propiedad para mostrar y modificar los apellidos.
<b>Nombre:</b>	Direccion
<b>Descripción:</b>	Propiedad para mostrar y modificar el direccion.
<b>Nombre:</b>	Universidad graduado
<b>Descripción:</b>	Propiedad para mostrar y modificar el universidad_graduado.
<b>Nombre:</b>	Id_profesor
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_profesor.
<b>Nombre:</b>	Solapin
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_profesor.
<b>Nombre:</b>	Annos_xp

# CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

<b>Descripción:</b>	Propiedad para mostrar y modificar el id_profesor.
<b>Nombre:</b>	Nombre
<b>Descripción:</b>	Propiedad para mostrar y modificar el nombre.
<b>Nombre:</b>	Universidad_procedencia
<b>Descripción:</b>	Propiedad para mostrar y modificar el universidad_procedencia.
<b>Nombre:</b>	Id_categoria
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_categoria.
<b>Nombre:</b>	Id_proyecto
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_proyecto.
<b>Nombre:</b>	Id_tipo_contrato
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_tipo_contrato.
<b>Nombre:</b>	Ci
<b>Descripción:</b>	Propiedad para mostrar y modificar el ci.
<b>Nombre:</b>	Id_municipio
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_municipio.
<b>Nombre:</b>	Id_grupodocente
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_grupodocente.

Tabla 16 Tabla clase entidad profesor general.

<b>Nombre:</b> Alumnos ayudantes	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id_alumno	int
id_asignatura	int
nombre_alumno	string
id_grupo_docente	int
telefono	string
id_apartamento	int
id_profesor	int
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	Alumnos ayudantes
<b>Descripción:</b>	Constructor de la clase.
<b>Nombre:</b>	Telefono
<b>Descripción:</b>	Propiedad para mostrar y modificar el telefono.
<b>Nombre:</b>	Id_grupo_docente
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_grupo_docente.
<b>Nombre:</b>	Nombre_alumno
<b>Descripción:</b>	Propiedad para mostrar y modificar el nombre_alumno.
<b>Nombre:</b>	Id_asignatura
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_asignatura.
<b>Nombre:</b>	Id_alumno
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_alumno.
<b>Nombre:</b>	Id_apartamento
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_apartamento.
<b>Nombre:</b>	Id_profesor
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_profesor.

Tabla 17 Tabla clase entidad alumno ayudante.

<b>Nombre:</b> Asistencia
---------------------------

<b>Tipo de clase:</b> Entidad	
Atributo	Tipo
id_tarea	int
id_profesor	int
id_asistencia	int
cumplimiento	bool
observaciones_asistencia	string
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	Asistencia
<b>Descripción:</b>	Constructor de la clase.
<b>Nombre:</b>	Id_asistencia
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_asistencia.
<b>Nombre:</b>	Cumplimiento
<b>Descripción:</b>	Propiedad para mostrar y modificar el cumplimiento.
<b>Nombre:</b>	Id_tarea
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_tarea.
<b>Nombre:</b>	Id_profesor
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_profesor.
<b>Nombre:</b>	Observaciones_Asistencia
<b>Descripción:</b>	Propiedad para mostrar y modificar el observaciones_Asistencia.

Tabla 18 Tabla clase entidad asistencia.

<b>Nombre:</b> Tarea	
<b>Tipo de clase:</b> Entidad	
Atributo	Tipo
id_tarea	int
fecha	string
hora	string
lugar	string
responsable	string
nombre_tarea	string
observaciones	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	Tarea
<b>Descripción:</b>	Constructor de la clase.
<b>Nombre:</b>	Id_tarea
<b>Descripción:</b>	Propiedad para mostrar y modificar el id_tarea.
<b>Nombre:</b>	Observaciones
<b>Descripción:</b>	Propiedad para mostrar y modificar las observaciones.
<b>Nombre:</b>	Nombre_tarea
<b>Descripción:</b>	Propiedad para mostrar y modificar el nombre_tarea.
<b>Nombre:</b>	Responsable
<b>Descripción:</b>	Propiedad para mostrar y modificar el responsable.
<b>Nombre:</b>	Lugar
<b>Descripción:</b>	Propiedad para mostrar y modificar el lugar.
<b>Nombre:</b>	Fecha
<b>Descripción:</b>	Propiedad para mostrar y modificar lal fecha.
<b>Nombre:</b>	Hora
<b>Descripción:</b>	Propiedad para mostrar y modificar la hora.

Tabla 19 Tabla clase entidad tarea.

<b>Nombre:</b> Jefe_Departamento_GestionarDatosAAyudante	
<b>Tipo de clase:</b> Interfaz	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	ImageButton1_Click
<b>Descripción:</b>	Evento que se ejecuta al hacer click sobre el botón.En el se crea un alumno ayudante y se muestra en el GridView1.
<b>Nombre:</b>	GridView1_SelectedIndexChanged
<b>Descripción:</b>	Evento que se ejecuta cuando se realiza una selección en el GridView de un alumno ayudante.
<b>Nombre:</b>	ImageButton2_Click
<b>Descripción:</b>	Evento que se ejecuta al hacer click sobre el botón.En el se modifica un alumno ayudante.
<b>Nombre:</b>	botón3_Click
<b>Descripción:</b>	Evento que se ejecuta al hacer click sobre el botón.En el se elimina un alumno ayudante.

Tabla 20 Tabla clase interfaz Jefe\_Departamento\_GestionarDatosAAyudante.

<b>Nombre:</b> Jefe_Departamento_GestionarDatosProfesor	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
ad	AccesoDatos
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	Page_Load
<b>Descripción:</b>	Evento que se ejecuta cuando se carga la página. En el se comprueba si existen profesores en el GridView1..
<b>Nombre:</b>	ImageButton3_Click
<b>Descripción:</b>	Evento que se ejecuta al hacer click sobre el botón.En el se modifica un profesor.
<b>Nombre:</b>	cb_tipocontrat_DataBound
<b>Descripción:</b>	Se inserta en la primera posicion del DropDownList la palabra seleccionar.
<b>Nombre:</b>	DropDownList1_DataBound
<b>Descripción:</b>	Se inserta en la primera posicion del DropDownList la palabra seleccionar.
<b>Nombre:</b>	cb_grupo_DataBound
<b>Descripción:</b>	Se inserta en la primera posicion del DropDownList la palabra seleccionar.
<b>Nombre:</b>	cb_cargo_DataBound
<b>Descripción:</b>	Se inserta en la primera posicion del DropDownList la palabra seleccionar.
<b>Nombre:</b>	cb_provincia_DataBound
<b>Descripción:</b>	Se inserta en la primera posicion del DropDownList la palabra seleccionar.
<b>Nombre:</b>	DropDownList2_DataBound
<b>Descripción:</b>	Se inserta en la primera posicion del DropDownList la palabra seleccionar.
<b>Nombre:</b>	cb_proyecto_DataBound
<b>Descripción:</b>	Se inserta en la primera posicion del DropDownList la palabra seleccionar.
<b>Nombre:</b>	ImageButton2_Click1
<b>Descripción:</b>	Muestra todos los atributos del profesor en sus respectivos campos para que estos sean modificados.
<b>Nombre:</b>	ImageButton1_Click1

<b>Descripción:</b>	Evento que se ejecuta al hacer click sobre el botón.En el se elimina un profesor.
---------------------	---

Tabla 21 Tabla clase interfaz Jefe\_Departamento\_GestionarDatosProfesor.

<b>Nombre:</b> Directivos_GestionarActividades	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
VsLista	List<ProfesorGeneral>
VsListaNombre	List<string>
VsListaNombre1	List<string>
VsListaID	List<string>
VsListaGeneral	List<ProfesorGeneral>
VsListaTar	List<Tarea>
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	Button1_Click
<b>Descripción:</b>	Evento que se ejecuta al hacer click sobre el botón.En el se inserta una tarea.
<b>Nombre:</b>	ImageButton1_Click
<b>Descripción:</b>	Evento que se ejecuta al hacer click sobre el botón.En el se define a los profesores que se les va a incluir en la asitencia.
<b>Nombre:</b>	ImageButton2_Click
<b>Descripción:</b>	Evento que se ejecuta al hacer click sobre el botón.En el se elimina a los profesores que por algún error se les iba a incluir en la tarea.
<b>Nombre:</b>	Button1_Click
<b>Descripción:</b>	Evento que se ejecuta al hacer click sobre el botón.En el se inserta la tarea..

Tabla 22 Tabla clase interfaz Directivos\_GestionarActividades.

<b>Nombre:</b> Controlador_GestionarAsistencia	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
VsLista	List<ProfesorGeneral>
VslistaAsist	List<Asistencia>
VsListaGeneral	List<ProfesorGeneral>
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	DropDownList3_SelectedIndexChanged
<b>Descripción:</b>	Evento que se ejecuta al insertar o eliminar algún profesor del DropDownList3 actualizandose el mismo.
<b>Nombre:</b>	ImageButton2_Click
<b>Descripción:</b>	Evento que se ejecuta al hacer click sobre el botón.En el se elimina un profesor de la lista del ListBox que erroneamente se le iba asignar la asistencia.
<b>Nombre:</b>	ImageButton1_Click
<b>Descripción:</b>	Evento que se ejecuta al hacer click sobre el botón.En el se inserta un profesor en la lista del ListBox al cual se le va a asignar la asistencia.
<b>Nombre:</b>	ImageButton3_Click
<b>Descripción:</b>	Evento que se ejecuta al hacer click sobre el botón.En el se asigna la asistencia..
<b>Nombre:</b>	RefrescarProfesoresConvocados
<b>Descripción:</b>	Metodo en el que se actualiza el ListBox.
<b>Nombre:</b>	RefrescarProfesoresAsignados
<b>Descripción:</b>	Metodo en el que se actualiza el ListBox.

Tabla 23 Tabla clase interfaz Directivos\_GestionarActividades.



<b>Nombre:</b> AccesoDatos	
<b>Tipo de clase:</b>	
<b>Atributo</b>	<b>Tipo</b>
Cadena_Conexion	SqlConnection
adapter	SqlDataAdapter
insertar	SqlCommand
eliminar	SqlCommand
modificar	SqlCommand
seleccionar	SqlCommand
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	AccesoDatos
<b>Descripción:</b>	Constructor de la clase.
<b>Nombre:</b>	AbrirConexion
<b>Descripción:</b>	Abre la conexión a la base de datos.
<b>Nombre:</b>	CerrarConexion
<b>Descripción:</b>	Cierra la conexión a la base de datos.
<b>Nombre:</b>	Insertar_AlumnoAyudante
<b>Descripción:</b>	Inserta un alumno ayudante en la base de datos.
<b>Nombre:</b>	Insertar_Apartamento
<b>Descripción:</b>	Inserta un apartamento en la base de datos.
<b>Nombre:</b>	Insertar_Departamento
<b>Descripción:</b>	Inserta un departamento en la base de datos.
<b>Nombre:</b>	Insertar_AreaTematica
<b>Descripción:</b>	Inserta un área temática en la base de datos.
<b>Nombre:</b>	Insertar_Asignatura
<b>Descripción:</b>	Inserta una asignatura en la base de datos.
<b>Nombre:</b>	Insertar_Asistencia
<b>Descripción:</b>	Inserta una asistencia en la base de datos.
<b>Nombre:</b>	Insertar_Cargo
<b>Descripción:</b>	Inserta un cargo en la base de datos.
<b>Nombre:</b>	Insertar_CategoriaDocente
<b>Descripción:</b>	Inserta una categoría docente en la base de datos.
<b>Nombre:</b>	Insertar_Disciplina
<b>Descripción:</b>	Inserta una disciplina en la base de datos.
<b>Nombre:</b>	Insertar_Edificio
<b>Descripción:</b>	Inserta un edificio a en la base de datos.
<b>Nombre:</b>	Insertar_ProfesorGeneral
<b>Descripción:</b>	Inserta un Profesor en la base de datos.
<b>Nombre:</b>	Insertar_Profesor_Interno
<b>Descripción:</b>	Inserta un profesor interno en la base de datos.
<b>Nombre:</b>	Insertar_ProfesorAdjunto
<b>Descripción:</b>	Inserta un profesor adjunto en la base de datos.
<b>Nombre:</b>	Modificar_AlumnoAyudante
<b>Descripción:</b>	Modifica un alumno ayudante en la base de datos.
<b>Nombre:</b>	Modificar_Apartamento
<b>Descripción:</b>	Modifica un apartamento en la base de datos.

## CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

<b>Nombre:</b>	Modificar_Departamento
<b>Descripción:</b>	Modifica un departamento en la base de datos.
<b>Nombre:</b>	Modificar_AreaTematica
<b>Descripción:</b>	Modifica un área temática en la base de datos.
<b>Nombre:</b>	Modificar_Asignatura
<b>Descripción:</b>	Modifica una asignatura en la base de datos.
<b>Nombre:</b>	Modificar_Asistencia
<b>Descripción:</b>	Modifica una asistencia en la base de datos.
<b>Nombre:</b>	Modificar_Cargo
<b>Descripción:</b>	Modifica un cargo en la base de datos.
<b>Nombre:</b>	Modificar_CategoriaDocente
<b>Descripción:</b>	Modifica una categoría docente en la base de datos.
<b>Nombre:</b>	Modificar_Disciplina
<b>Descripción:</b>	Modifica una disciplina en la base de datos.
<b>Nombre:</b>	Modificar_Edificio
<b>Descripción:</b>	Modifica un edificio a en la base de datos.
<b>Nombre:</b>	Modificar_ProfesorGeneral
<b>Descripción:</b>	Modifica un Profesor en la base de datos.
<b>Nombre:</b>	Modificar_ProfesorInterno
<b>Descripción:</b>	Modifica un profesor interno en la base de datos.
<b>Nombre:</b>	Modificar_ProfesorAdjunto
<b>Descripción:</b>	Modifica un profesor adjunto en la base de datos.
<b>Nombre:</b>	Eliminar_AlumnoAyudante
<b>Descripción:</b>	Elimina un alumno ayudante en la base de datos.
<b>Nombre:</b>	Eliminar_Apartamento
<b>Descripción:</b>	Elimina un apartamento en la base de datos.
<b>Nombre:</b>	Eliminar_Departamento
<b>Descripción:</b>	Elimina un departamento en la base de datos.
<b>Nombre:</b>	Eliminar_AreaTematica
<b>Descripción:</b>	Elimina un área temática en la base de datos.
<b>Nombre:</b>	Eliminar_Asignatura
<b>Descripción:</b>	Elimina una asignatura en la base de datos.
<b>Nombre:</b>	Eliminar_Asistencia
<b>Descripción:</b>	Elimina una asistencia en la base de datos.
<b>Nombre:</b>	Eliminar_Cargo
<b>Descripción:</b>	Elimina un cargo en la base de datos.
<b>Nombre:</b>	Eliminar_Categoria
<b>Descripción:</b>	Elimina una categoría docente en la base de datos.
<b>Nombre:</b>	Eliminar_Disciplina
<b>Descripción:</b>	Elimina una disciplina en la base de datos.
<b>Nombre:</b>	Eliminar_Edificio
<b>Descripción:</b>	Elimina un edificio a en la base de datos.
<b>Nombre:</b>	Eliminar_ProfesorGeneral
<b>Descripción:</b>	Elimina un Profesor en la base de datos.
<b>Nombre:</b>	Eliminar_ProfesorAdjunto
<b>Descripción:</b>	Elimina un profesor adjunto en la base de datos.
<b>Nombre:</b>	Seleccionar_Profesores_Tarea
<b>Descripción:</b>	Selecciona una lista de profesores dada una tarea.
<b>Nombre:</b>	Seleccionar_Profesores_Asistencia
<b>Descripción:</b>	Selecciona una lista de profesores dada una asistencia.
<b>Nombre:</b>	Seleccionar_Profesores
<b>Descripción:</b>	Selecciona una lista de todos los profesores.
<b>Nombre:</b>	Seleccionar_Tareas_ID

## CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

<b>Descripción:</b>	Selecciona una lista de todas las tareas.
<b>Nombre:</b>	Seleccionar_Asistencia
<b>Descripción:</b>	Selecciona una lista de asistencia dado un profesor.
<b>Nombre:</b>	Seleccionar_Profesor_ID
<b>Descripción:</b>	Selecciona una lista de todos los profesores.
<b>Nombre:</b>	Seleccionar_IDProfesor_Solapin
<b>Descripción:</b>	Selecciona un profesor dado un solapin.

Tabla 24 Tabla clase AccesoDatos.

<b>Nombre:</b> cc_gestionartareas	
<b>Tipo de clase:</b> controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	Registrar_actividad
<b>Descripción:</b>	Inserta una actividad en la base de datos.
<b>Nombre:</b>	Modificar_actividad
<b>Descripción:</b>	Modifica una actividad en la base de datos.
<b>Nombre:</b>	Eliminar_actividad
<b>Descripción:</b>	Modifica una actividad en la base de datos.

Tabla 25 Tabla controladora cc\_gestionartareas.

<b>Nombre:</b> cc_buscar_informacion	
<b>Tipo de clase:</b> controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	Buscar_profesor
<b>Descripción:</b>	Busca un profesor en la base de datos.
<b>Nombre:</b>	Buscar_alumno ayudante
<b>Descripción:</b>	Busca un alumno ayudante en la base de datos.
<b>Nombre:</b>	Buscar_tareas
<b>Descripción:</b>	Busca una tarea en la base de datos.

Tabla 26 Tabla controladora cc\_buscar\_informacion.

## 3.3 Diseño de la Base de Datos

La forma física como estén almacenados los datos es independiente del concepto que se tenga de ellos. Son el conjunto de programas que saben cómo traer, unir y mostrar los datos, así como aquellos encargados de almacenarlos, los que le dan coherencia al concepto base de datos, del diseño de la estructura lógica depende toda la funcionalidad del sistema.

Son muchas las consideraciones a tomar en cuenta al momento de hacer el diseño de la base de datos, quizá las más fuertes sean: [26]

- La velocidad de acceso.
- El tamaño de la información.
- El tipo de la información.
- Facilidad de acceso a la información.
- Facilidad para extraer la información requerida.
- El comportamiento del manejador de bases de datos con cada tipo de información.

### 3.3.1 Diagrama de clases persistentes.

A partir del diagrama de clases persistentes se realizaron una serie de transformaciones, derivadas de los procesos de normalización del diseño de la base de datos, y de la selección de una serie de tablas para la configuración para la conformación de un diagrama entidad relación homogéneo. También como parte de las relaciones que se presentan en una base de datos y que por lo general difieren de lo que se puede modelar a partir de clases, como el caso de una relación de mucho a mucho genera una tabla intermedia a nivel de base de datos y esto no se modela así en un diagrama de clases.

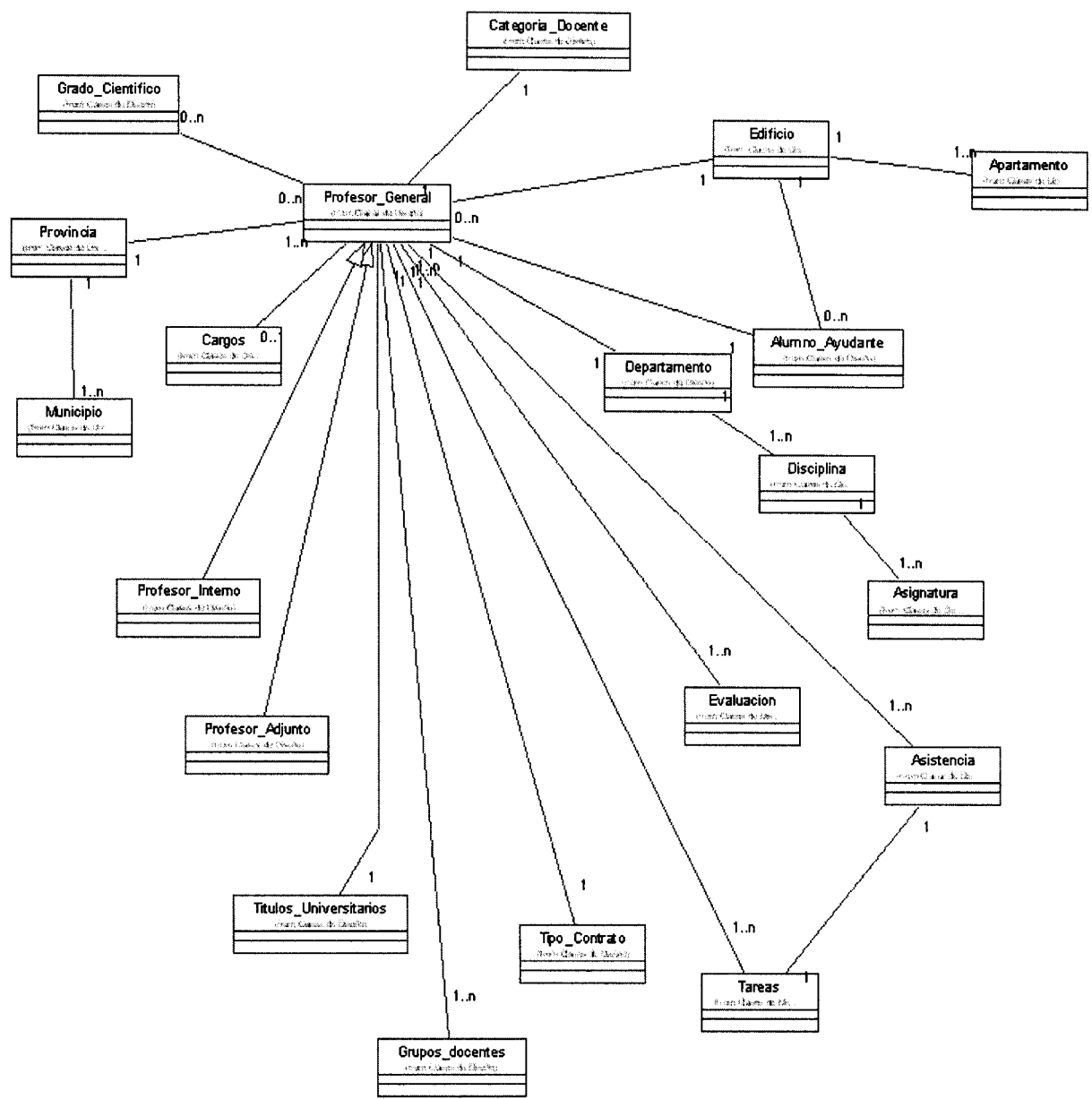


Figura 15 Diagrama de clases persistentes

3.3.2 Modelo de Datos

El modelo de datos es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia. [27]



Nombre: Profesor general		
Tipo de clase: Entidad		
Atributo	Tipo	Descripción
Id_profesor	Integer	Es el identificador de la tabla.
nombre	String	Es el nombre del profesor
apellidos	String	Son los apellidos del profesor.
Dirección_particular	String	Responde a la dirección particular del profesor.
CI	Integer	Es el carné de identidad del profesor.
universidad_procedencia	String	Es la última universidad donde trabajo dicho profesor.
universidad_graduado	String	Universidad donde se graduó dicho profesor.
Id_tipo_contrato	Integer	Identificador del tipo de contrato.
Id_categoria	Integer	Identificador de la categoría docente.
Id_alumno	Integer	Identificador del alumno ayudante al cual el profesor va a apadrinar.
Id_proyecto	Integer	Identificador del proyecto al cual pertenece dicho profesor.
Id_evaluación	Integer	Identificador de la evaluación que de dicho profesor.
Id_municipio	Integer	Identificador del municipio de donde proviene este profesor.
Id_apartamento	Integer	Identificador del apartamento donde reside el profesor.
Id_Cargos		Identificador del cargo que tiene este profesor.

Tabla 27 Profesor general

<b>Nombre: Proyecto</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Id_proyecto</b>	<b>Integer</b>	Identificador del proyecto en cuestión.
<b>Nombre_proyecto</b>	<b>String</b>	Nombre que tiene el proyecto.
<b>Responsable_proyecto</b>	<b>String</b>	Nombre y apellidos del profesor responsable del proyecto.
<b>Lab_proyecto</b>	<b>String</b>	Laboratorio donde se encuentra el proyecto.
<b>Telefono_proyecto</b>	<b>Integer</b>	Teléfono que tiene el proyecto.
<b>Id_area</b>	<b>Integer</b>	Identificador del área temática al cual pertenece el proyecto.

Tabla 28 Proyecto

<b>Nombre: Área temática</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Id_area</b>	<b>Integer</b>	Identificador del área temática.
<b>Nombre _ área</b>	<b>String</b>	Nombre que lleva el área temática.
<b>Responsable_ área_ temática</b>	<b>String</b>	Nombre y apellidos del responsable del área temática.
<b>Teléfono_ área_ temática</b>	<b>String</b>	Teléfono que tiene la oficina del área temática.
<b>Id_polo</b>	<b>Integer</b>	Identificador del polo científico al cual pertenece el área temática.

Tabla 29 Área temática



<b>Nombre: Polo _ científico</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Nombre_polo	String	Nombre del polo científico.
Responsable	String	Nombre y apellidos del responsable del polo científico.
Teléfono _ polo	Integer	Teléfono que tiene la oficina del área temática.
Id_polo	Integer	Identificador del polo científico.

Tabla 30 Polo científico

<b>Nombre: Cargos</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Nombre_cargo	String	Nombre del cargo.
Id_cargo	Integer	Identificador del cargo.

Tabla 31 Cargos

<b>Nombre: Tipo _ contrato</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Nombre_contrato	String	Nombre del contrato.
Id_tipo_contrato	Integer	Identificador del contrato.

Tabla 32 Tipo contrato

<b>Nombre: Categoría docente</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Nombre_categoria	String	Nombre de la categoría docente.
Id_categoria	Integer	Identificador de la categoría docente.

Tabla 33 Categoría docente

<b>Nombre: Evaluación</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Tipo_ evaluación	String	Nombre del tipo de la evaluación.
Id_ evaluación	Integer	Identificador de la evaluación.

Tabla 34 Evaluación

<b>Nombre: Titulos_ universitarios</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Nombre_ titulo	String	Nombre del título universitario.
Id_ titulo	Integer	Identificador del título universitario.

Tabla 35 Títulos universitarios

<b>Nombre: Grado_ Científico</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Nombre_ grado	String	Nombre del grado científico.
Id_ grado	Integer	Identificador grado científico.

Tabla 36 Grado Científico

<b>Nombre: Provincia</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Nombre_ provincia	String	Nombre de la provincia.
Id_ provincia	Integer	Identificador de la provincia.

Tabla 37 Provincia

<b>Nombre: Municipio</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Nombre_ municipio	String	Nombre del municipio.
Id_ municipio	Integer	Identificador del municipio.

Tabla 38 Municipio

<b>Nombre: Profesor_Adjunto</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Dirección_centro_trabajo	String	Dirección del centro de trabajo de dicho profesor.
Teléfono_centro_trabajo	Integer	Teléfono del centro de trabajo de dicho profesor.
Id_profesor	Integer	Identificador del profesor.

Tabla 39 Profesor Adjunto

<b>Nombre: Profesor_Interno</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_profesor	Integer	Identificador del profesor.
Id_apartamento	Integer	Identificador del apartamento

Tabla 40 Profesor Interno

<b>Nombre: Profesor_titulosuniversitarios</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_profesor	Integer	Identificador del profesor.
Id_titulo	Integer	Identificador del título universitario.

Tabla 41 Profesor títulos universitarios

<b>Nombre: Profesor_asignatura</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_profesor	Integer	Identificador del profesor.
Id_asignatura	Integer	Identificador de la asignatura.

Tabla 42 Profesor asignatura

<b>Nombre: Asignatura</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Id_disciplina</b>	<b>Integer</b>	Identificador de la disciplina al cual pertenece la asignatura.
<b>Id_asignatura</b>	<b>Integer</b>	Identificador de la asignatura.
<b>Nombre_asignatura</b>	<b>String</b>	Nombre de la asignatura.
<b>Responsable_asignatura</b>	<b>String</b>	Nombre y apellidos del responsable de la asignatura.

Tabla 43 Asignatura

<b>Nombre: Disciplina</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Id_disciplina</b>	<b>Integer</b>	Identificador de la disciplina.
<b>Id_departamento</b>	<b>Integer</b>	Identificador del departamento al cual pertenece la disciplina.
<b>Nombre_disciplina</b>	<b>String</b>	Nombre de la disciplina.
<b>jefe_disciplina</b>	<b>String</b>	Nombre y apellidos del responsable de la disciplina.

Tabla 44 Disciplina

<b>Nombre: Departamento</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Id_departamento</b>	<b>Integer</b>	Identificador del departamento.
<b>Nombre_departamento</b>	<b>String</b>	Nombre del departamento.
<b>Jefe_departamento</b>	<b>String</b>	Nombre y apellidos del responsable del departamento.

Tabla 45 Departamento

Nombre: Alumno _Ayudante		
Tipo de clase: Entidad		
Atributo	Tipo	Descripción
Id_alumno	Integer	Identificador del alumno ayudante.
telefono	Integer	Teléfono del alumno ayudante.
Nombre_alumno	String	Nombre y apellidos del alumno ayudante.
Id_asignatura	Integer	Identificador de la asignatura que imparte el alumno.
Id_grupo_docente	Integer	Identificador del grupo docente al cual el alumno imparte clases.

Tabla 46 Alumno Ayudante

Nombre: Tarea		
Tipo de clase: Entidad		
Atributo	Tipo	Descripción
Id_tarea	Integer	Identificador de la tarea.
hora	Date	Hora en la que se va a realizar la tarea.
Responsable	String	Nombre y apellidos del responsable de la tarea.
Nombre	String	Nombre que recibe la tarea en cuestión.
Observaciones	String	Comentarios que puedan poner los controladores.

Tabla 47 Tarea

<b>Nombre: Asistencia</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Id_profesor</b>	<b>Integer</b>	Identificador del profesor el cual está cumpliendo la tarea.
<b>Id_tarea</b>	<b>Integer</b>	Identificador de la tarea.
<b>Observaciones</b>	<b>String</b>	Comentario de la realización de la tarea.
<b>Cumplimiento</b>	<b>bool</b>	Define si se cumple o no la tarea.

Tabla 48 Asistencia

<b>Nombre: Usuarios</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Pregunta_seguridad</b>	<b>String</b>	Es una pregunta de seguridad al usuario.
<b>Respuesta_seguridad</b>	<b>String</b>	El usuario realiza la respuesta de seguridad.
<b>Nombre</b>	<b>String</b>	Nombre del usuario.
<b>Password</b>	<b>String</b>	Contraseña.

Tabla 49 Usuarios

<b>Nombre: Privilegio</b>		
<b>Tipo de clase: Entidad</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Descripción</b>	<b>String</b>	Hace una breve descripción del privilegio.
<b>Nombre_privilegio</b>	<b>String</b>	Nombre del privilegio.
<b>Id_privilegio</b>	<b>Integer</b>	Identificador del privilegio.

Tabla 50 Privilegio

### 3.4 Principios de diseño

#### Estándares de la interfaz de la aplicación

Existen factores de calidad de una aplicación que influyen en la satisfacción del usuario, se puede clasificar en aquellos relacionados con la calidad de los contenidos, y la calidad del diseño de la aplicación. La importancia del diseño de la aplicación se basa en que éste será el

que modele la interacción entre los usuarios y la aplicación, y por tanto posibilitará o no la consecución de los objetivos perseguidos por el usuario que no es más que encontrar información, registrar datos entre otros.

Para el diseño de la interfaz del sistema se tuvo en cuenta varios aspectos, como las operaciones que el usuario puede realizar, para ello se ubicaron las mismas en la parte izquierda y superior de la aplicación logrando que estén accesibles. Se trabaja con plantilla para lograr uniformidad entre todas las páginas de la aplicación. Las páginas son formularios lo más entendibles.

### **3.5 Generalidades de la Implementación.**

#### **3.5.1 Modelo de Despliegue.**

Un diagrama de despliegue muestra las relaciones físicas entre los componentes *hardware* y *software* en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes *software* (procesos y objetos que se ejecutan en ellos). Estarán formados por instancias de los componentes *software* que representan manifestaciones del código en tiempo de ejecución (los componentes que sólo sean utilizados en tiempo de compilación deben mostrarse en el diagrama de componentes).

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes *software*, objetos, procesos (caso particular de un objeto). En general un nodo será una unidad de computación de algún tipo, desde un sensor a un *mainframe*. Las instancias de componentes *software* pueden estar unidas por relaciones de dependencia, posiblemente a interfaces (ya que un componente puede tener más de una interfaz).[28]

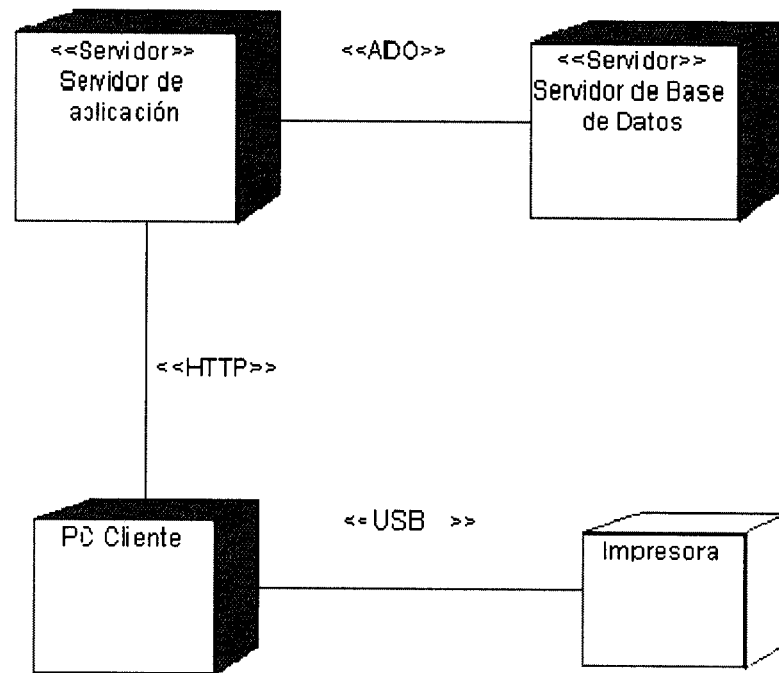


Figura 17 Diagrama de Despliegue

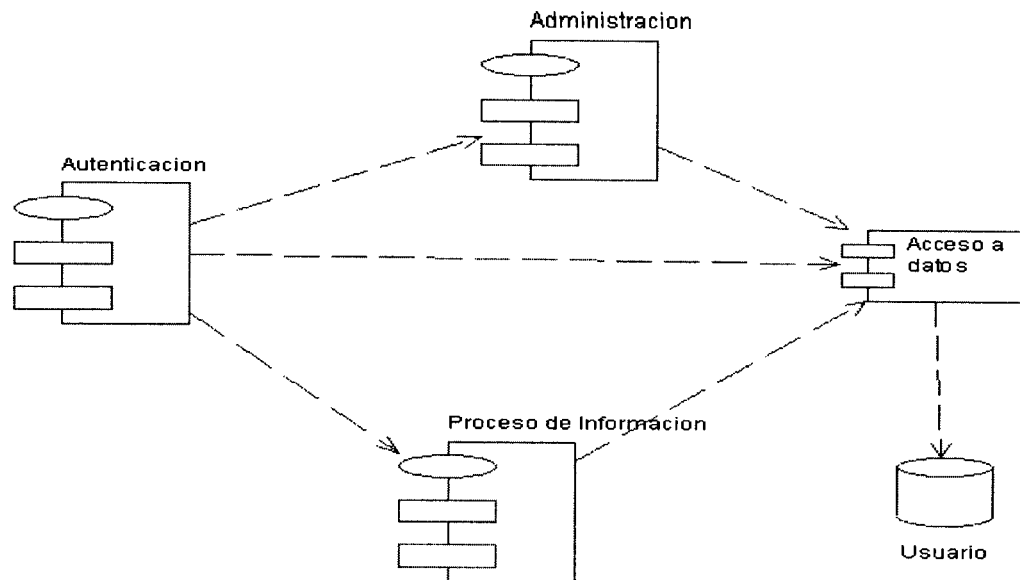
### 3.5.2 Modelo de Implementación.

Lo que distingue a un diagrama de componentes de otros tipos de diagramas es su contenido. Normalmente contienen componentes, interfaces y relaciones entre ellos. Y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo.

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes *software*, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del *software*, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. En cuanto a los componentes, sólo aparecen tipos de componentes, ya que las instancias específicas de cada tipo se encuentran en el diagrama de despliegue.

Un paquete en un diagrama de componentes representa una división física del sistema. Los paquetes se organizan en una jerarquía de capas donde cada capa tiene una interfaz bien definida.[29]





**Figura 18** Diagrama de Componentes

### 3.6 Prueba del sistema propuesto

#### Prueba de caja blanca

Con la prueba de la caja blanca del software se comprueba los caminos lógicos del software proponiendo casos de prueba que se ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coinciden con el esperado o mencionado.

La complejidad ciclomática es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa. La métrica se basa en la representación gráfica de flujo de control del programa. De dicho análisis se desprende una medida cuantitativa de la dificultad de prueba y una indicación de la fiabilidad final. Cuando se utiliza en el contexto del método de prueba del camino básico, el valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y da el límite superior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez. Es una de las métricas de software más ampliamente aceptada, ya que ha sido concebida para ser independiente del lenguaje. A continuación se le calculará la complejidad ciclomática al fragmento de código del sistema propuesto que se muestra debajo.

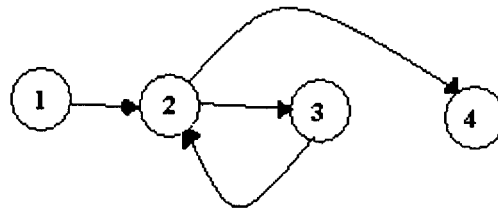
[30]

```

CerrarConexion();
for (int i = 0; i < dataset.Tables[0].Rows.Count; i++)
{
    profesor = new ProfesorGeneral();
    profesor.Id_profesor =
int.Parse(dataset.Tables[0].Rows[i].ItemArray[0].ToString());
    profesor.Nombre = dataset.Tables[0].Rows[i].ItemArray[6].ToString();

    lista.Add(profesor);
}
    
```

Es necesario primero introducir una sencilla notación para la representación del flujo de control, denominada Grafos de Flujo de Control de un programa. El grafo de flujo representa el flujo de control lógico mediante la notación que se ilustra en la siguiente figura:



**Figura 19** Complejidad Ciclomática

Complejidad ciclomática  $V(G) = \text{Cantidad de Aristas } [A] - \text{Cantidad de nodos } [N] + 2$ .

$$V(G) = A - N + 2$$

$$V(G) = 4 - 4 + 2$$

$$V(G) = 2$$

Complejidad Ciclomática	Evaluación del Riesgo
1-10	Programa Simple, sin mucho riesgo
11-20	Más complejo, riesgo moderado
21-50	Complejo, Programa de alto riesgo
50	Programa no testeable, Muy alto riesgo

**Tabla 51** Tabla de comparación de la complejidad ciclomática.

Ya calculada la complejidad y comparando el resultado con la tabla de valores se llega a la conclusión que el programa no tendrá grandes riesgos.

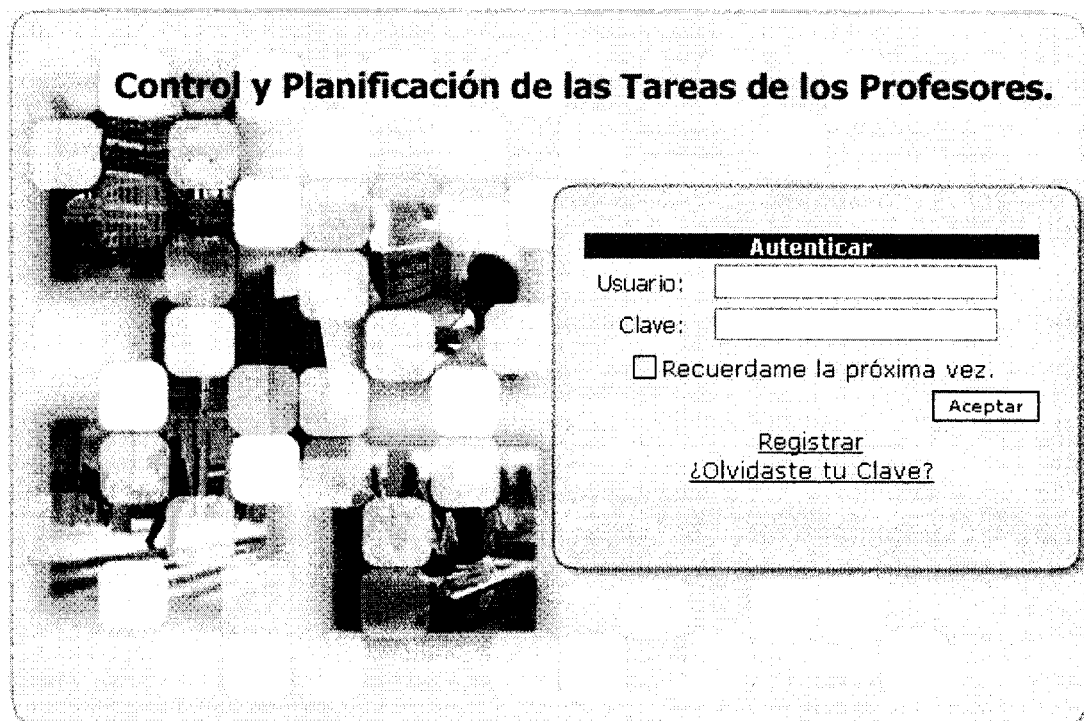
## Prueba de caja negra

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

### Caso de prueba: Autenticar

- 1- Usuario y contraseña que no existan en la base de datos.
- 2- Usuario que no exista en la base de datos y contraseña que si exista en la base de datos.
- 3- Usuario que exista en la base de datos y contraseña que no exista en la base de datos.
- 4- Usuario y contraseña bien.

La figura que se muestra a continuación representa la interfaz donde se van a realizar estas pruebas que no es más que la pagina donde se va loguear el usuario.



**Figura 20** Prueba de caja negra

A continuación se van a describir los diferentes casos de prueba de autenticación. Donde se va a demostrar que un usuario que no se encuentre registrado no podrá acceder a la aplicación, la seguridad es imprescindible en la aplicación propuesta pues la información referente a los datos de profesores es delicada y no puede ser manejada por personal no autorizado.

Caso de Uso	Autenticar Directivo
Caso de prueba	1
Entrada	Nombre de usuario="jejejeje" Contraseña="jejejeje"
Condiciones de ejecución	Como el usuario que existe en la base de datos es Sidius y la contraseña fu70, los datos entrados no coinciden.
Resultado	El sistema muestra un mensaje informando al usuario que los datos introducidos no son correctos.

Tabla 52 Prueba de caja negra caso de prueba 1

Caso de Uso	Autenticar Directivo
Caso de prueba	2
Entrada	Nombre de usuario="jejejeje" Contraseña="fu70"
Condiciones de ejecución	Como el usuario que existe en la base de datos es Sidius y la contraseña fu70, el nombre entrado no coincide.
Resultado	El sistema muestra un mensaje informando al usuario que los datos introducidos no son correctos.

Tabla 53 Prueba de caja negra caso de prueba 2

Caso de Uso	Autenticar Directivo
Caso de prueba	3
Entrada	Nombre de usuario="Sidius" Contraseña="serg"
Condiciones de ejecución	Como el usuario que existe en la base de datos es Sidius y la contraseña fu70, la contraseña de entrada no coinciden.
Resultado	El sistema muestra un mensaje informando al usuario que los datos introducidos no son correctos.

Tabla 54 Prueba de caja negra caso de prueba 3

Caso de Uso	Autenticar Directivo
Caso de prueba	4
Entrada	Nombre de usuario="Sidius" Contraseña="fu70"
Condiciones de ejecución	Como el usuario que existe en la base de datos es Sidius y la contraseña fu70, los datos entrados son válidos.
Resultado	El sistema permite su acceso al sistema.

Tabla 55 Prueba de caja negra caso de prueba 4

### Conclusiones

Como resultado del estudio realizado en este capítulo, correspondiente a la etapa de diseño e implementación del sistema, se modelaron los diagramas de clases para cada uno de los casos de uso críticos del sistema. Se obtuvo el diagrama de clases persistentes y el modelo de datos. Se establecieron las pautas para el diseño de la interfaz, finalmente se muestra el diagrama de despliegue, todo esto antes mencionado se logró con el correcto uso de los artefactos correspondientes que brinda el Rational Rose.

# CONCLUSIONES

Al concluir el trabajo se han cumplido con el objetivo y las tareas planteadas, obteniendo los siguientes resultados:

- Se diseñó un sistema informático utilizando la metodología de ingeniería del software propuesta.
- Se implementó un sistema informático que facilita la gestión de la información referente de los profesores en el departamento ingeniería de software y práctica profesional de la facultad 7.

## RECOMENDACIONES

- Llevar este sistema a todas las facultades de la Universidad de las Ciencias Informáticas.
- Mantener sobre el sistema un estricto cumplimiento del proceso de mantenimiento y actualización periódica, logrando así que se mantenga la fiabilidad y funcionamiento óptimo del sistema y de la información que se gestiona en el mismo.
- Elaborar un manual del usuario para el sistema y hacerlo extensivo a todos los relacionados con el sistema.
- Continuar con la investigación para garantizar mejoras en futuras versiones del sistema.
- Analizar las sugerencias de los profesores y directivos en cuanto al sistema.
- Migrar el sistema a software libre.

## REFERENCIAS BIBLIOGRÁFICAS

1. Santos., A.C., *Tecnología de Gestión de Recursos Humanos.* . 1999: La Habana: Academia ed.
2. *recursos5.* [Recuperado 2008 marzo 11]; Disponible en: <http://www.gestiopolis.com/recursos5/docs/eco/apligesti.htm>
3. *Control.* . [Recuperado 2008 marzo 20]; Disponible en: [http://es.wikipedia.org/wiki/Control\\_de\\_gesti%C3%B3n](http://es.wikipedia.org/wiki/Control_de_gesti%C3%B3n).
4. GONZÁLEZ, O.S. [Recuperado el 2008 junio 5]; Disponible en: [http://www.elprisma.com/apuntes/administracion\\_de\\_empresas/control estrategico/](http://www.elprisma.com/apuntes/administracion_de_empresas/control estrategico/).
5. [Recuperado 2008 junio 11]; Disponible en: <http://www.ikertia.net/proyectos.php?grupo=8>.
6. Pereira, J.E. *Aplicaciones Web y ASPs.* . [Recuperado el 2008 enero 6]; Disponible en: <http://www.mercadeo.com/57 ASPs.htm>.
7. *Desarrollo de aplicaciones por capas. Introducción a la programación en el servidor. Plataformas de desarrollo.* 2004-2005.
8. *Introducción a ASP.NET: El entorno de Trabajo de .NET.* [Recuperado el 2008 mayo 23]; Disponible en: [http://www.mctekk.com/mcimages/287\\_contenidopdf.pdf](http://www.mctekk.com/mcimages/287_contenidopdf.pdf).
9. *Microsoft Visual Studio .NET 2008* [Recuperado el 2008 mayo 14]; Disponible en: <http://www.microsoft.com/products/info/product.aspx?view=44&pcid=0a05620b-d256-487f-88d7-ceaa334cf95a&type=ovr>.
10. DONALD, M.M., *ASP.Net Manual de Referencia.* 2002, Madrid: Mc Graw Hill.
11. *C Sharp.* [Recuperado el 2008 marzo 16]; Disponible en: [http://es.wikipedia.org/wiki/C\\_Sharp](http://es.wikipedia.org/wiki/C_Sharp).
12. Seco, J.A.G. *El lenguaje de programación C#.* 2001 [Recuperado el 2008 abril 24]; Disponible en: <http://www.programacion.com/tutorial/csharp/3/>.



13. *Aplicaciones distribuidas en tres capas*. . 2007 [Recuperado el 2008 enero 25]; Disponible en:  
[http://www.upo.es/cic/proyectos/tecnicos/tres\\_capas/descripcion/index.jsp](http://www.upo.es/cic/proyectos/tecnicos/tres_capas/descripcion/index.jsp).
14. *Desarrollo de aplicaciones por capas* 2005.
15. VALLE, J.G. *Definición arquitectura cliente servidor*. 2005 [Recuperado 2008 mayo 19]; Disponible en: <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml#gcliente>.
16. *Arquitectura Cliente/Servidor*. [Recuperado 2008 marzo 21]; Disponible en:  
<http://www.csi.map.es/csi/silice/Global71.html>.
17. *¿Qué es un servidor? - Definición de servidor*. 2008 [Recuperado 2008 enero 10]; Disponible en: <http://www.masadelante.com/faq-servidor.htm>.
18. *Ventajas del Modelo C/S a las Organizaciones*. [Recuperado 2008 marzo 24]; Disponible en: <http://www.inei.gob.pe/biblioineipub/bancopub/inf/Lib5038/ven2.htm>.
19. *Base\_de\_datos*. . [Recuperado 2008 marzo 14]; Disponible en:  
[http://es.wikipedia.org/wiki/Base\\_de\\_datos](http://es.wikipedia.org/wiki/Base_de_datos).
20. *¿ Qué es un Sistema Gestor de Bases de Datos o SGBD ?* [Recuperado 2008 mayo 11]; Disponible en: <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
21. Corporation, M. *Microsoft SQL Server 2000 Enterprise Edition*. 2002 [Recuperado 2008 abril 12]; Disponible en:  
[http://thesource.ofallevil.com/venezuela/socios/local/telcel-saint/downs/RV\\_SQLBrochure.pdf](http://thesource.ofallevil.com/venezuela/socios/local/telcel-saint/downs/RV_SQLBrochure.pdf).
22. Danysoft. *Modelado de bases de datos*. 2006 [Recuperado 2008 febrero 16]; Disponible en: <http://www.danysoft.info/free/model2.pdf>.
23. *RUP*. [Recuperado 2008 mayo 12]; Disponible en: :  
<http://es.wikipedia.org/wiki/RUP>.
24. *UML*. . [Recuperado 2008 mayo 6]; Disponible en:  
<http://es.wikipedia.org/wiki/UML>.

25. CONALLEN, J., *Construyendo Aplicaciones Web con UML* segunda edición ed.
26. Vargas, G.d.J.S. [Recuperado 2008 junio 21]; Disponible en: <http://www.monografias.com/trabajos30/base-datos/base-datos.shtml>
27. *Modelos de datos*. [Recuperado 2008 mayo 22]; Disponible en: [http://es.wrs.yahoo.com/\\_ylt=A0geunBJFm1IXoYBIDGT.Qt.;\\_ylu=X3oDMTE0bnFvbHYxBHNIYwNzcgRwb3MDMQRjb2xvA2FjMgR2dGIkA0VTQzAxMF8xMA--/SIG=12tq08r2g/EXP=1215195081/\\*\\*http%3A/atenea.udistrital.edu.co/profesores/jdimate/basedatos1/tema1\\_4.htm](http://es.wrs.yahoo.com/_ylt=A0geunBJFm1IXoYBIDGT.Qt.;_ylu=X3oDMTE0bnFvbHYxBHNIYwNzcgRwb3MDMQRjb2xvA2FjMgR2dGIkA0VTQzAxMF8xMA--/SIG=12tq08r2g/EXP=1215195081/**http%3A/atenea.udistrital.edu.co/profesores/jdimate/basedatos1/tema1_4.htm).
28. Vilas, A. [Recuperado 2008 mayo 17]; Disponible en: <http://www-gris.det.uvigo.es/~avilas/UML/node49.html>.
29. [Recuperado 2008 mayo 17]; Disponible en: <http://www-gris.det.uvigo.es/~avilas/UML/node50.html>.
30. Rizzi, I.F. [Recuperado 2008 junio 5]; Disponible en: <http://www.itba.edu.ar/capis/rtis/articulosdeloscuadernosetapaprevia/RIZZI-COMPLEJIDAD.pdf>

# BIBLIOGRAFÍA

*Aplicaciones distribuidas en tres capas.* . 2007 [Recuperado el 2008 enero 25];

Disponible en:

[http://www.upo.es/cic/proyectos/tecnicos/tres\\_capas/descripcion/index.jsp](http://www.upo.es/cic/proyectos/tecnicos/tres_capas/descripcion/index.jsp).

*Arquitectura Cliente/Servidor.* [Recuperado 2008 marzo 21]; Disponible en:

<http://www.csi.map.es/csi/silice/Global71.html>.

*Base\_de\_datos.* . [Recuperado 2008 marzo 14]; Disponible en:

[http://es.wikipedia.org/wiki/Base de datos](http://es.wikipedia.org/wiki/Base_de_datos).

CONALLEN, J., *Construyendo Aplicaciones Web con UML* segunda edición ed.

*Control.* . [Recuperado 2008 marzo 20]; Disponible en:

[http://es.wikipedia.org/wiki/Control de gesti%C3%B3n](http://es.wikipedia.org/wiki/Control_de_gesti%C3%B3n).

Corporation, M. *Microsoft SQL Server 2000 Enterprise Edition.* 2002 [Recuperado

2008 abril 12]; Disponible en:

[http://thesource.ofallevil.com/venezuela/socios/local/telcel-saint/downs/RV\\_SQLBrochure.pdf](http://thesource.ofallevil.com/venezuela/socios/local/telcel-saint/downs/RV_SQLBrochure.pdf).

*C Sharp.* [Recuperado el 2008 marzo 16]; Disponible en:

[http://es.wikipedia.org/wiki/C\\_Sharp](http://es.wikipedia.org/wiki/C_Sharp).

*Desarrollo de aplicaciones por capas. Introducción a la programación en el servidor. Plataformas de desarrollo.* 2004-2005.

DONALD, M.M., *ASP.Net Manual de Referencia.* 2002, Madrid: Mc Graw Hill.

GONZÁLEZ, O.S. [Recuperado el 2008 junio 5]; Disponible en:

[http://www.elprisma.com/apuntes/administracion de empresas/controlestrategico/](http://www.elprisma.com/apuntes/administracion_de_empresas/controlestrategico/).

[Recuperado 2008 mayo 17]; Disponible en: <http://www->

[gris.det.uvigo.es/~avilas/UML/node50.html](http://gris.det.uvigo.es/~avilas/UML/node50.html).

[Recuperado 2008 junio 11]; Disponible en:

<http://www.ikertia.net/proyectos.php?grupo=8>

*Modelado de bases de datos*. 2006 [Recuperado 2008 febrero 16]; Disponible en:

<http://www.danysoft.info/free/model2.pdf>.

*Introducción a ASP.NET: El entorno de Trabajo de .NET*. [Recuperado el 2008 mayo

23]; Disponible en: [http://www.mctekk.com/mcimages/287\\_contenido.pdf](http://www.mctekk.com/mcimages/287_contenido.pdf).

*Microsoft Visual Studio .NET 2008* [Recuperado el 2008 mayo 14]; Disponible en:

<http://www.microsoft.com/products/info/product.aspx?view=44&pcid=0a05620b-d256-487f-88d7-ceaa334cf95a&type=ovr>.

*Modelos de datos*. [Recuperado 2008 mayo 22]; Disponible en:

[http://es.wrs.yahoo.com/\\_ylt=A0geunBJFm1IXoYBIDGT.Qt.;\\_ylu=X3oDMTE0bnFvbHYxBHNIYwNzcgRwb3MDMQRib2xvA2FjMgR2dGIkA0VTQzAxMF8xMA--/SIG=12tq08r2g/EXP=1215195081/\\*\\*http%3A/atenea.udistrital.edu.co/profesores/jdimate/basedatos1/tema1\\_4.htm](http://es.wrs.yahoo.com/_ylt=A0geunBJFm1IXoYBIDGT.Qt.;_ylu=X3oDMTE0bnFvbHYxBHNIYwNzcgRwb3MDMQRib2xvA2FjMgR2dGIkA0VTQzAxMF8xMA--/SIG=12tq08r2g/EXP=1215195081/**http%3A/atenea.udistrital.edu.co/profesores/jdimate/basedatos1/tema1_4.htm).

Pereira, J.E. *Aplicaciones Web y ASPs*. . [Recuperado el 2008 enero 6]; Disponible

en: [http://www.mercadeo.com/57\\_ASPs.htm](http://www.mercadeo.com/57_ASPs.htm).

*¿Qué es un servidor? - Definición de servidor*. 2008 [Recuperado 2008 enero 10];

Disponible en: <http://www.masadelante.com/faq-servidor.htm>.

*¿Qué es un Sistema Gestor de Bases de Datos o SGBD ?* [Recuperado 2008 mayo

11]; Disponible en: <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.

*recursos5*. [Recuperado 2008 marzo 11]; Disponible en:

<http://www.gestiopolis.com/recursos5/docs/eco/apligesti.htm>

Rizzi, I.F. [Recuperado 2008 junio 5]; Disponible en:

<http://www.itba.edu.ar/capis/rtis/articulosdeloscuadernosetaaprevia/RIZZI-COMPLEJIDAD.pdf>

*RUP*. [Recuperado 2008 mayo 12]; Disponible en: : <http://es.wikipedia.org/wiki/RUP>.

Santos., A.C., *Tecnología de Gestión de Recursos Humanos*. . 1999: La Habana: Academia ed.

Seco, J.A.G. *El lenguaje de programación C#*. 2001 [Recuperado el 2008 abril 24]; Disponible en: <http://www.programacion.com/tutorial/csharp/3/>.

UML. . [Recuperado 2008 mayo 6]; Disponible en: <http://es.wikipedia.org/wiki/UML>.

VALLE, J.G. *Definición arquitectura cliente servidor*. 2005 [Recuperado 2008 mayo 19]; Disponible en: <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml#qcliente>.

Vargas, G.d.J.S. [Recuperado 2008 junio 21]; Disponible en: <http://www.monografias.com/trabajos30/base-datos/base-datos.shtml>

*Ventajas del Modelo C/S a las Organizaciones*. [Recuperado 2008 marzo 24]; Disponible en: <http://www.inei.gob.pe/biblioineipub/bancopub/inf/Lib5038/ven2.htm>.

Vilas, A. [Recuperado 2008 mayo 17]; Disponible en: <http://www-gris.det.uvigo.es/~avilas/UML/node49.html>.

## Diagrama de colaboración Gestionar Profesor

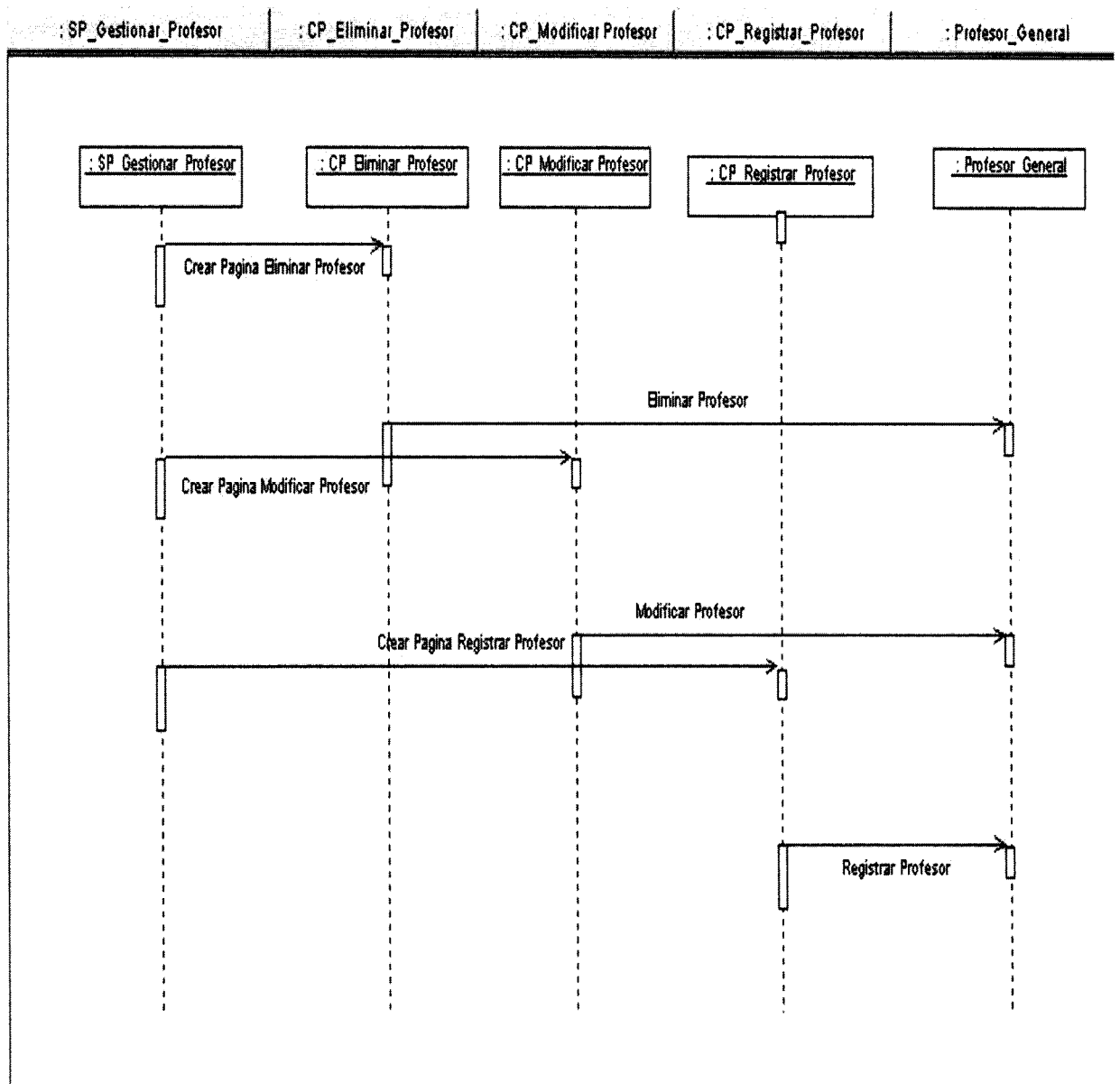
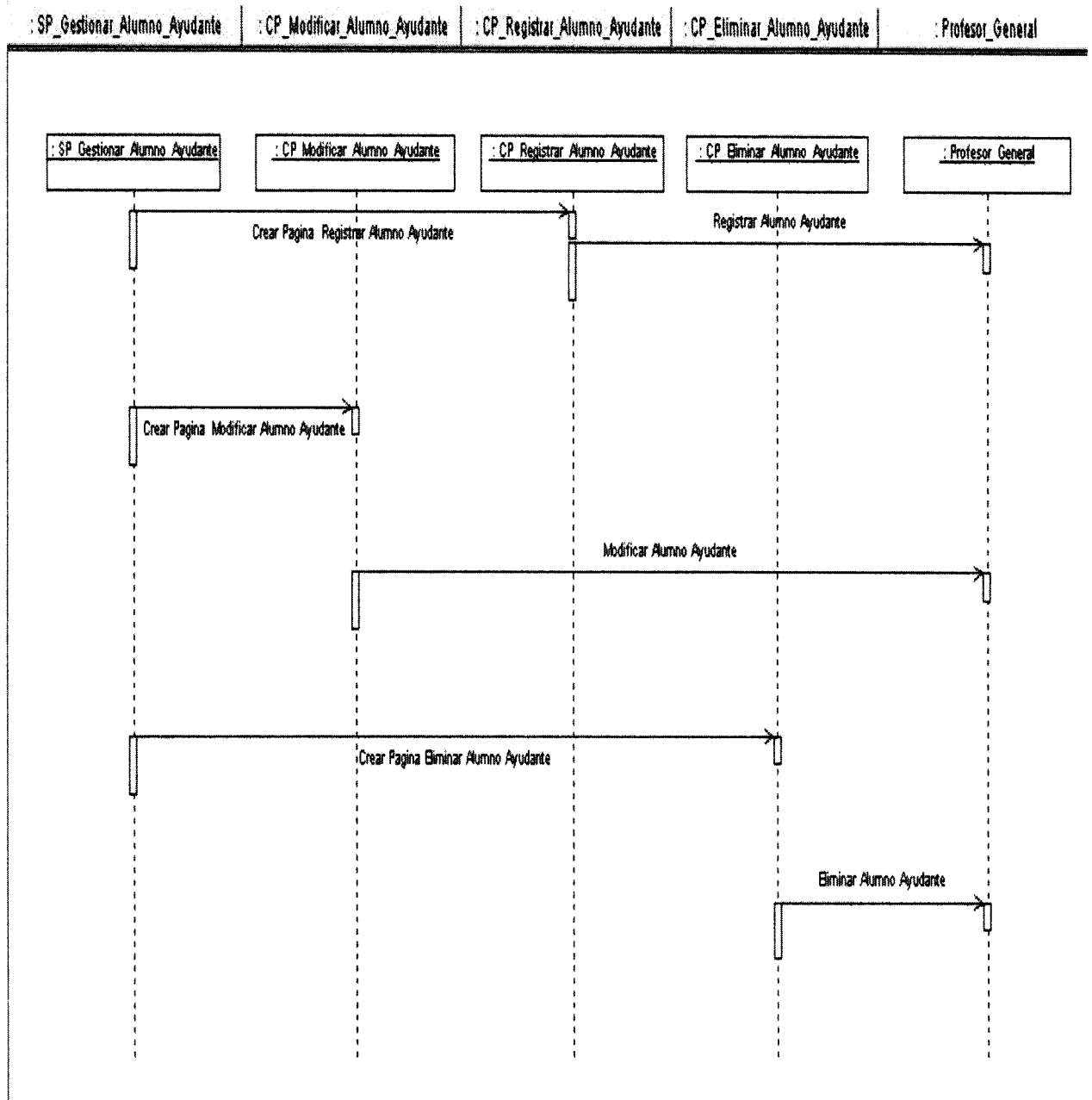


Figura 21 Diagrama de colaboración Gestionar Profesor

## Diagrama de colaboración Gestionar Alumno Ayudante.



**Figura 22** Diagrama de colaboración Gestionar Alumno Ayudante

# Diagrama de colaboración Buscar

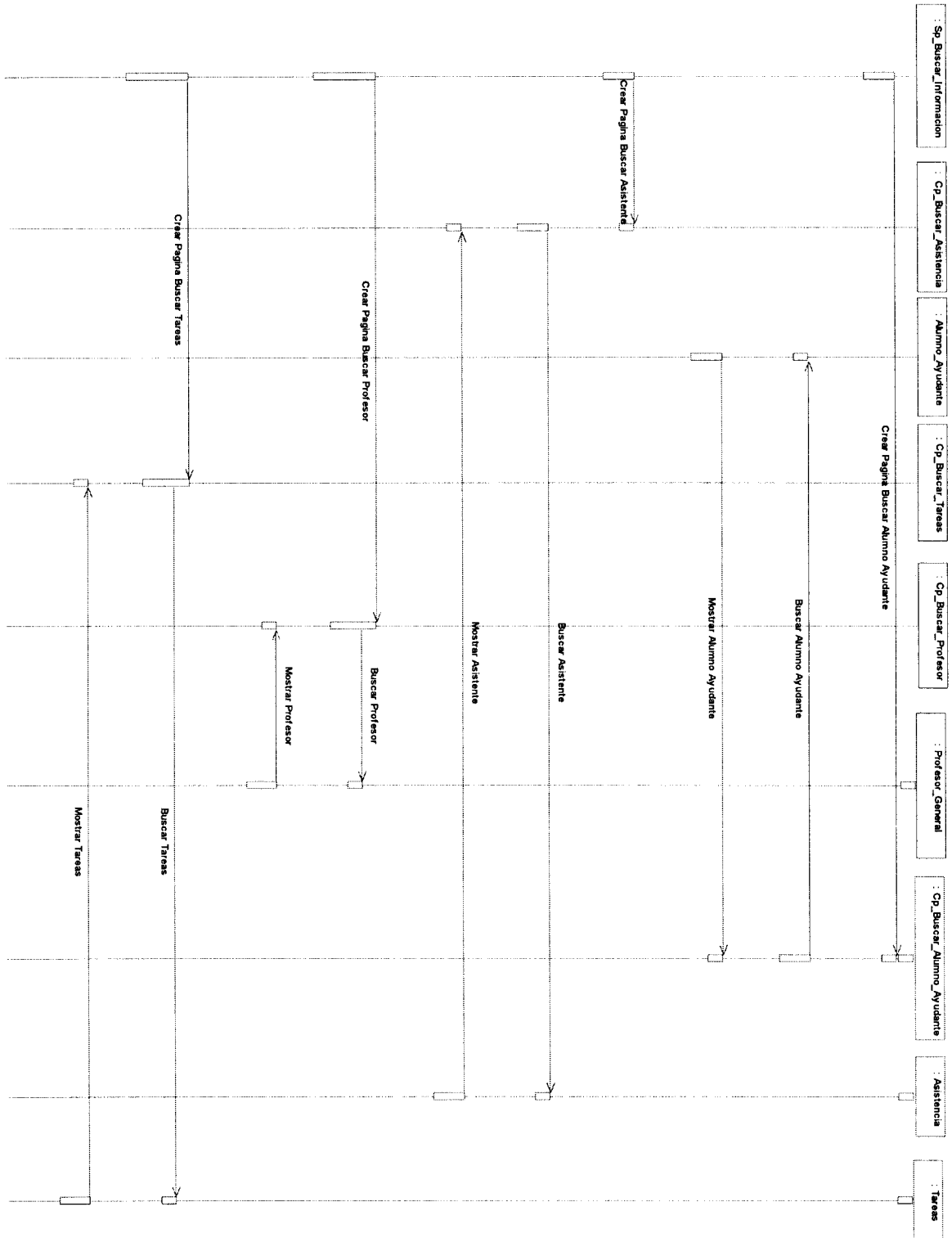


Figura 23 Diagrama de colaboración Buscar



## GLOSARIO DE TÉRMINOS

- **ASP** Active Server Page. Páginas web que son ejecutadas en el servidor, en tiempo de corrida al cliente se le envía el resultado de este proceso.
- **HTTP** (HyperText Transfer Protocol; Protocolo de transferencia de hipertexto): Es un protocolo del nivel de aplicación usado para la transferencia de información entre sistemas, de forma clara y rápida. Ha sido usado por el World-Wide Web desde 1990.
- **JAVA:** Lenguaje de programación para la elaboración de pequeñas aplicaciones exportables a la red (applets) y capaces de operar sobre cualquier plataforma a través, normalmente, de navegadores WWW. Permite dar dinamismo a las páginas web.
- **SQL** Server Query Language. Lenguaje de consulta que permite la recuperación de información desde un servidor de bases de datos.
- **SGBD** Colección de datos interrelacionados y un conjunto de programas para acceder dichos datos. Son productos de software fabricados por diferentes proveedores y con diferentes características que los diferencian unos de otros: Dbase, Fox, Access, Informix, Unify, Oracle, Interbase.
- **Código abierto** (en inglés *open source*) es el término con el que se conoce al software distribuido y desarrollado libremente. Fue utilizado por primera vez en 1998 por algunos usuarios de la comunidad del software libre, tratando de usarlo como reemplazo al ambiguo nombre original en inglés del software libre (free software).
- **Memoria de acceso aleatorio** (en inglés *Random Access Memory*, frecuentemente abreviado para **RAM**) es un tipo de memoria que permite la lectura y la escritura, utilizada como memoria primaria en sistemas electrónicos digitales.
- **Foreach:** sentencia de lenguaje de programación *c#*.
- **Polimorfismo:** capacidad que tienen los objetos de una clase de responder a un mismo mensaje o evento en función de los parámetros utilizados durante su invocación. Un objeto polimórfico es una entidad que puede contener valores de diferentes tipos durante la ejecución del programa.

- **Switch** :(en castellano "conmutador") es un dispositivo electrónico de interconexión de redes de ordenadores de enlace de datos.
- **Goto** :(*ir a* en ingles) es una sentencia o instrucción muy común en los lenguajes de con el objetivo de controlar el flujo del programa.
- **P/Invoke**: (en inglés **Platform Invocation Services**), es una herramienta que permite llamar a los códigos nativos de las dlls.
- **Mainframe**: es una computadora grande, potente y costosa usada principalmente por una gran compañía para el procesamiento de una gran cantidad de datos; por ejemplo, para el procesamiento de transacciones bancarias.
- **XPath**: (XML Path Language) es un lenguaje que permite construir expresiones que recorren y procesan un documento XML.
- **XML**: sigla en inglés *Extensible Markup Language* (lenguajes de marcas extensibles), es un metalenguaje extensible de etiquetas, permite definir la gramática de lenguajes específicos.
- **HTTP**: define la sintaxis y la semántica que utilizan los elementos software de la arquitectura web (clientes, servidores, proxies) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.
- **OLAP**: es en ingles de procesamiento analítico en línea (On- Line Analytical Processing). Es una solución utilizada en el campo de la llamada inteligencia empresarial (o Business Intelligence) cuyo objetivo es agilizar la consulta de grandes cantidades de datos. Para ello utiliza estructuras multidimensionales que contienen datos resumidos de grandes bases de datos o Sistemas Transaccionales.
- **CASE**:(Computer Aided Software Engineering,Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de coste y dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software.
- **LAN**: es una red de área local, o red local, es la interconexión de varios ordenadores y periféricos. (LAN es la abreviatura inglesa de Local Area Network, red de área local). Su extensión está limitada físicamente a un edificio o a un entorno de hasta 100 metros.

- **WAN:** es una Red de Área Amplia (Wide Area Network en inglés), es un tipo de red de computadoras capaz de cubrir distancias desde unos 100 hasta unos 1000 km, dando el servicio a un país o un continente.