

003.7  
602  
S  
TD-0006-04-02

TD-0006-04-02



Universidad de Ciencias Informáticas  
Departamento de Informatización

**Akademios**

**Sistema Automatizado para la Gestión Académica.**

**Subsistema de Registro Virtual.**

**Trabajo de Diploma para optar por el título de Ingeniero Informático.**

Autor: Daniel Leandro González Rojas

Tutor: Ing. Emil Lima Valdés

*Ciudad de La Habana, Junio 2004*

*"Año del 45 Aniversario del Triunfo de la Revolución"*

## **AGRADECIMIENTOS**

---

### **Agradecimientos**

A mi mamá, por darme la vida y por estar siempre a mi lado.

A mi papá, por exigirme ser mejor siempre y por enseñarme el camino a seguir.

A mi hermano, por estar conmigo en las buenas y malas.

A mi abuela, por haberme dado su amor incondicionalmente y por lo orgullosa que se sentiría de verme graduado; es para ella con todo mi corazón.

A mi abuelo,

A Emil, por su apoyo.

A Tania, por ayudarme en los momentos más difíciles.

A mis compañeros,

Y a todos los que de una forma u otra contribuyeron con la realización de este trabajo.

**MUCHAS GRACIAS**

### Resumen

El proyecto que aquí se presenta consiste en el desarrollo de un Sistema de Registro Académico Virtual en la Universidad de Ciencias Informáticas de Cuba (UCI).

La necesidad de desarrollarlo surge con la idea de crear una ciudad digital donde la mayoría de los procesos se encuentren automatizados. Para lograr este objetivo la institución cuenta con una infraestructura que soporta el desarrollo de sistemas informáticos capaces de gestionar gran cantidad de información con un costo mínimo de tiempo de respuesta y con la posibilidad de integración entre estos.

Con la implantación de este se da solución a un problema de primer orden del centro, pues dentro de las tareas fundamentales del instituto se encuentra la docencia y dentro de ella la gestión de la información referente al desarrollo docente de los estudiantes.

Esta solución permite a los involucrados en el proceso docente el procesamiento automatizado de esta información. Este es capaz de sustituir de forma íntegra a su similar de papel y a su vez incluir funcionalidades que su antecesor no presentaba, además de una fuerte integración en cuanto a intercambio de información con los demás sistemas de Gestión Docente del centro y con otros sistemas que necesiten u ofrezcan información al mismo.

Con una interfaz fácil y amigable, una ayuda que permite al usuario conocer las soluciones a los problemas mas frecuentes que puedan presentar y con la posibilidad de agregarle otros módulos para darle solución a las necesidades crecientes de los usuarios, este sistema representa una herramienta a considerar por parte de las instituciones que desarrollen una actividad similar a la del centro para la cual fue desarrollada.

<b>Introducción</b> .....	1
<b>Capítulo #1 Fundamentación teórica</b> .....	6
1.1 Estado del Arte .....	7
1.2.1 La tecnología .NET .....	12
1.2.1.1 Arquitectura Framework.NET .....	12
1.2.2 Lenguajes de programación .....	13
1.2.3 ASP.NET .....	14
1.2.4 Modelación de aplicaciones Web sobre la plataforma ASP.NET .....	18
1.2.5 Programación Multicapas .....	19
1.2.6 Servicios Web (Web Services) .....	21
1.2.6.1 Protocolos que usan los Servicios Web .....	22
1.3 Sistemas Gestores de Base de Datos (SGDB).....	23
1.3.1 Catálogos de sistema de bases de datos .....	23
1.3.2 Estructuras de almacenamiento físico y lógico.....	25
1.3.3 Datos de bandas .....	26
1.3.4 Registros de transacciones y recuperación automática .....	27
1.3.5 Redes .....	29
1.4 RUP .....	31
1.4.1 Características del Proceso Unificado.....	32
1.4.2 Rational Rose.....	33
<b>Capítulo #2 Características del sistema</b> .....	35
2.1 Situación problemática.....	36
2.1.1 Objetivos estratégicos de la organización y procesos de negocio que los soportan.....	36
2.2 Objeto de automatización .....	37
2.2.1 Registro Académico .....	37
2.3 Información que se maneja.....	39
2.4 Propuesta de sistema .....	39
2.4.1 Descripción general de la propuesta de sistema.....	40
2.4.1.1 Asistencia .....	40
2.4.1.2 Notas en las evaluaciones de una asignatura .....	40
2.4.1.3 Estado de los estudiantes .....	40
2.4.2 Otras soluciones existentes en el centro .....	41
2.5 Especificación de los requisitos de software.....	42
2.5.1 Requisitos funcionales .....	42
2.5.2 Requisitos no funcionales.....	45
2.5.2.2 Requisitos de usabilidad .....	45
2.5.2.3 Requisitos de Rendimiento .....	45
2.5.2.4 Requisitos de Soporte.....	45
2.5.2.5 Requisitos de Seguridad.....	46
2.6 Definición de los casos de uso.....	46
<b>Capítulo #3 Análisis y diseño del sistema</b> .....	48
3.1 Diagrama de clases del análisis.....	49
3.2 Diagramas de interacción del sistema .....	50
3.2.1 Diagrama de secuencia del caso de uso “Registrar asistencia”, sección “Registrar asistencia” .....	50

## ÍNDICE

---

3.2.2 Diagrama de secuencia del caso de uso “Registrar asistencia”, sección “Modificar Asistencia” .....	52
3.2.3 Diagrama de secuencias del caso de uso “Registrar notas”, sección “Registrar notas” .....	54
3.2.4 Diagrama de secuencia del caso de uso “Registrar notas”, sección “Modificar notas” .....	56
3.2.5 Diagrama de secuencia del caso de uso “Mostrar estado”, sección “Mostrar Estado” .....	58
3.3 Diagramas de clases de diseño .....	60
3.3.1 Diagrama de clases de diseño del caso de uso “Registrar asistencia” ..	60
3.3.3 Diagrama de clases de diseño del caso de uso “Mostrar estado” .....	64
3.4 Descripción de las clases .....	66
3.5 Diseño de la base de datos .....	76
3.5.1 Diagrama de entidad relación .....	76
<b>Conclusiones</b> .....	83
<b>Recomendaciones</b> .....	84
<b>Bibliografía</b> .....	86
<b>Anexos</b> .....	88
Anexo 1 .....	89
1.1 Modelo del dominio .....	89
1.1.1 Modelo del dominio del problema .....	89
1.2 Definición de los casos de uso del sistema .....	90
1.2.1 Definición de los actores .....	90
1.2.3 Diagrama de casos de uso del sistema .....	101
1.2.4 Casos de uso expandidos .....	102
<b>Glosario de términos</b> .....	111

## Introducción

La sociedad ha experimentado un desarrollo tecnológico acelerado en las últimas décadas. Con el surgimiento de las microcomputadoras es posible el almacenamiento de gran cantidad de información sin la utilización de papel y con poco trabajo manual humano; el desafío que plantea el momento actual es transformar la universidad en un centro especializado en la generación de conocimientos y para esto las funciones principales de las instituciones que se dediquen a esta tarea deben presentar un alto grado de informatización.

En la Universidad de Ciencias Informáticas (UCI), se ha interiorizado esto y se está trabajando en función de tener la mayor cantidad de información posible en soporte magnético y distribuida en su red de computadoras. Esto ha influido grandemente en el desarrollo de sistemas informáticos que sean capaces de sustituir los procesos que hasta el momento se desarrollan de forma manual o semi automatizada.

## Problema

¿Cómo lograr que los involucrados (directivos, secretarias, profesores y estudiantes) tengan un papel activo en el proceso de control del rendimiento académico de los estudiantes en un centro de estudios superiores?

## Actualidad y necesidad del trabajo

En la actualidad en el centro se están utilizando los sistemas de asistencia y de evaluaciones de los estudiantes los cuales trabajan aislados entre si y entre los demás sistemas del centro, además no presentan todas las potencialidades de un registro académico automatizado que además de sus funciones básicas, debe incluir otras que le proporcionan una mayor capacidad en la prestación de

## **INTRODUCCIÓN**

---

servicios a los usuarios. Por esta razón surge en el centro la necesidad de implementar un Sistema de Registro Académico Virtual con el cual los involucrados en los procesos de gestión de asistencia y de evaluaciones de los estudiantes contarán con una herramienta poderosa en este sentido y además integrada al sistema de gestión docente y a otros sistemas del centro.

### **Antecedentes**

Para el desarrollo de este trabajo surgió la necesidad de investigar como trabajan los Sistemas de Gestión Docente y dentro de ellos específicamente los que de una forma u otra se relacionaban con el Registro Académico. Para esto se visitaron varios sitios en internet que tratan este tema, dentro de los sitios que se visitaron se encuentran, el sitio Web de la Universidad Politécnica de Nicaragua [1], el sitio de la Universidad Politécnica de Madrid [2], entre otros, de los cuales se analizaron sus propuestas y se escogieron los aspectos positivos. Se analizaron los documentos de los sistemas de asistencia y evaluaciones que están actualmente en explotación en el centro y otros sistemas cubanos que podían ser de utilidad en este trabajo, todo esto mostró hacia donde se dirige esta tarea en el mundo y en Cuba y las características que debía presentar un Sistema de Registro Académico automatizado para satisfacer las necesidades de sus usuarios finales.

## **INTRODUCCIÓN**

---

### **Aportes prácticos esperados**

Con la implantación de este sistema se espera que los involucrados en este proceso cuenten con una poderosa herramienta de trabajo que sea capaz de gestionar de una forma rápida y eficiente la información que se maneja en un Registro Académico tradicional, e incluya otras funcionalidades que le brinden mayor riqueza a esta labor, al mismo tiempo que presente una fuerte integración con los demás sistemas del centro y que le proporcione un uso racional a los recursos informáticos de los cuales dependerá.

### **Objeto de estudio**

- Aspectos que debe presentar un sistema de registro académico automatizado.
- Tecnologías Web para el desarrollo de aplicaciones cliente/servidor.
- Integración de los servicios del sistema de Registro Académico Automatizado.
- Análisis del Sistema de Bases de Datos a utilizar.

### **Campo de acción**

- La gestión docente en una universidad, específicamente la gestión de Registros Académicos.
- Sistemas integrados a la gestión de Registros Académicos.
- Las tendencias actuales para el desarrollo de aplicaciones Web.
- Sistemas de gestión de bases de datos.
- Las tecnologías Visual Studio .Net y Microsoft SQL Server.



## **INTRODUCCIÓN**

---

### **Objetivos del trabajo**

#### **Objetivos generales**

Desarrollar un sistema automatizado que permita a todo los involucrados en el ciclo académico llevar eficientemente el control directo del rendimiento de los estudiantes de forma tal que sirva de apoyo a la toma de decisiones de los directivos académicos.

#### **Objetivos específicos**

- Analizar aspectos teóricos, conceptuales de los registros académicos.
- Estudiar acerca del los Registros Académicos que se encuentran en las universidades de Cuba, Latinoamérica y el mundo.
- Implementar un sistema de Registro Académico Automatizado para la Universidad de Ciencias Informáticas de Cuba que se ajuste a sus necesidades como institución, e integrarlo con los otros sistemas de gestión académicas del centro.

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

Para cumplir los objetivos se desarrollaron varias tareas que sirvieron de ubicación en el dominio del problema, cuáles podían ser las posibles soluciones del mismo y para tener un mejor conocimiento de las herramientas informáticas que se debían utilizar para dar solución al problema planteado, para lograr todo esto, se realizó una entrevista con los involucrados en el negocio, los cuales brindaron gran cantidad de información referente al trabajo con el Registro Académico en el centro; resultó extremadamente importante analizar el trabajo de los profesores en la institución pues estos son los mayores implicados en este proceso, por lo que se realizó una entrevista haciendo énfasis en los profesores los cuales podían aportar ideas interesantes para la realización de

## INTRODUCCIÓN

---

este trabajo, toda esta información se proceso y sirvió como punto de partida para la modelación del sistema. Se realizo un estudio de las herramientas informáticas que mas se estaban utilizando en el mundo y en Cuba para implementar la solución del problema, teniendo en cuenta que se trataba de una aplicación Web con las características que esto trae consigo, se analizaron las ventajas y desventajas de los sistemas gestores de bases de datos y cual era el mas óptimo para utilizar en este sistema. Se buscó información referente a las cuestiones legales de los registros académicos en los centros universitarios de Cuba y en específico en la Universidad de Ciencias Informáticas.

### **Estructuración del contenido con una breve explicación de sus partes**

El trabajo se dividió en tres capítulos que contienen la información referente a la investigación realizada, así como la parte de análisis y diseño del sistema a desarrollar la cual está organizada de la siguiente forma:

**Capítulo 1:** Fundamentación Teórica.

**Capítulo 2:** Características del sistema:

En este capítulo se realiza un estudio preliminar del sistema y el proceso de negocio en el cual se especifican los objetivos y los requerimientos esperados, así como las tecnologías elegidas para el fin.

**Capítulo 3:** Análisis y diseño del sistema:

Se definen las expansiones de los casos de usos y el modelo del dominio tratado, se desarrollan las secuencias del proceso y el diseño de la arquitectura del sistema.

### Capítulo #1 Fundamentación teórica

#### Introducción

En este capítulo se aborda el estado actual de desarrollo de las aplicaciones de gestión académica en el ámbito mundial, nacional y en la Universidad, además de un estudio de las tecnologías en las que se apoya el desarrollo del sistema en función de un análisis de las tendencias actuales.

Han sido implementados muchos sistemas de gestión académica, que de una forma u otra responden a las exigencias, cada día más crecientes, de responsabilidad, calidad y rapidez en todo proceso docente; de estructuras muy variadas donde cada uno le da el enfoque de la actividad que necesita informatizar, se han aplicado fundamentalmente las tecnologías Web.

### 1.1 Estado del Arte

La importancia que han adquirido en la sociedad los procesos docentes educativos en los centros universitarios, ha provocado que en el mundo hayan surgido un sin número de sistemas automatizados dirigidos al control eficiente y veloz de dichos procesos.

A continuación se hace un estudio crítico sobre los principales sistemas existentes en el ámbito nacional e internacional para la gestión académica.

#### **Ámbito Internacional**

AGORA: **A**plicación de **G**estión de **O**rganización **A**cadémica.

Abarca todas aquellas actividades que están ligadas con el medio académico, y se extienden desde las pruebas de accesos y preinscripción de los alumnos a la gestión de los títulos pasando por la matrícula, gestión de actas, expedientes, estadísticas y gestión de becas, incluyendo la gestión económica de la Universidad relacionada con la actividad académica.

Permite la estructura de cualquier plan de estudios dando cabida a las diversas variedades y peculiaridades de las Universidades, y a partir del mismo, debido a su definición flexible (mediante la estructuración de un grafo), se garantiza el seguimiento de cada uno de los alumnos y su trayectoria académica. A través del expediente admite consultar el historial académico del estudiante en un determinado plan de estudio, modificar o completar sus datos personales y da acceso a cualquier información relacionada con su presencia en la universidad. Incluye la solicitud de convalidaciones, adaptaciones o reconocimiento de créditos, el registro de traslado de expediente, la emisión de todo tipo de certificado y listado relacionado. La matrícula puede realizarse por la secretaría o por el sistema a través de la auto matrícula, realizándose la misma a un conjunto de asignaturas y asociándose a una gestión económica. Incluye además la gestión de grupos de asignaturas. Posibilita la elaboración de

informes propios incluyendo o restringiendo la información que desea relacionar.

### **GAUSS:** Gestión Académica Universitaria sobre Sistemas Informáticos

El sistema informático para la gestión académica, GAUSS, es una herramienta desarrollada por la Universidad de Cantabria en conjunto con la empresa Semicrol, gestiona todos los procesos de preinscripción, matrícula, pruebas de acceso y expedientes.

Este sistema ha permitido integrar totalmente los módulos para la ordenación académica y la gestión de la carga docente con los de recursos humanos, investigación, asuntos económicos, etc. Facilita la descentralización de los procesos a los distintos departamentos, la obtención de un expediente académico único y la presentación automatizada de la oferta formativa en la red.

Una de las aplicaciones de este sistema es el proceso de autoservicio para alumnos, que funciona por medio de terminales instalados en el campus y a través de Internet. Estas aplicaciones, permiten definir todos los ciclos y planes de estudio, personalizar estos planes para cada curso académico y sistematizar el tratamiento de los expedientes.

Ofrece información general sobre los planes de estudio, incluye aplicaciones para los alumnos (consulta de calificaciones, situación del expediente, peticiones de certificados, etc.), los profesores (tratamiento de las calificaciones, información sobre las asignaturas, horarios, contacto con los alumnos a través del correo electrónico, etc.) y el personal de administración y servicios.

### **GAT Universidades**

Es un sistema integrado de información para la gestión académica orientado a las universidades presenciales y no presenciales. Permite la gestión de todo el proceso administrativo y de comunicación entre la universidad y los alumnos, desde el ingreso hasta la finalización de los estudios. Automatiza los elementos claves del entorno de docencia, como son materiales didácticos, herramientas de comunicación, etc.

Todo el proceso de interacción con el estudiante se realiza, desde la primera solicitud de información hasta la entrega final del título, mediante una estructura de módulos creados a partir de la secuencia lógica del proceso: acceso, matrícula, exámenes, evaluación de estudios previos, gestiones varias y títulos.

Dentro de las principales herramientas que permiten el buen funcionamiento del sistema se encuentra el "expediente académico" (con una función principalmente informativa) y la "secretaría" (herramienta de gestión al servicio del administrador del sistema y del estudiante). El control académico del expediente se realiza de forma automática.

**SIGA:** Sistema de Información de Gestión Académica.

Gestión completa informatizada de un conservatorio, academia, colegio, instituto o universidad.

El usuario organizará sus estudios en varios planes de estudios, cada plan en varios cursos y cada curso con varias asignaturas. Ajuste horario semanal y total, presupuesto y precio de las asignaturas/cursos. Los alumnos se matriculan en estudios organizados en grupos y turnos (sesión).

SIGA dedica un módulo a la gestión de mensajería, donde controla el envío a través de empresas de transporte o mensajerías mediante soporte informático (disquete, etc.) Creación automática de un disquete con todos los datos de los

alumnos a los que se les quiere realizar un envío (documentación de su curso, exámenes, etc.).

Permite controlar de forma automática la asistencia del estudiante, estadísticas y porcentaje. Genera listados para los expedientes académicos, calificaciones de un alumno/curso, observaciones, diplomas, etc. Alta, baja, consulta y modificación de profesores. Da la posibilidad de generar cualquier listado o informe que se desee. Brinda la opción de generar horarios semanales/ mensuales/ anuales.

### **Estudios en el país**

#### **GESTACAD: Sistema de Gestión Académica**

Este sistema surgió con la idea de desarrollar un software que permitiera automatizar la gestión académica de las Universidad de Matanzas, el mismo gestiona parte la información académica de los estudiantes universitarios y la información de los profesores que forman parte del proceso docente educativo. Este sistema está en uso en la UCI.

El sistema está concebido por módulos, entre los que se diferencian, los módulos de actualización de datos y el sitio Web, a través del cual se muestran las diversas salidas de la aplicación.

El módulo de actualización permite realizar la matrícula de los nuevos ingresos, dar baja, rematricular un estudiante que ha sido baja, sin permitir ratificar la matrícula, realizar traslados, ni registrar datos necesarios para los graduados.

El módulo de información permite buscar un estudiante, mostrar una estadística general de cuantos hay por criterios (facultad, centro de procedencia, vía de ingreso, provincia, etc.). Al interactuar con este sistema se puede notar que existen problemas con la navegabilidad, el ambiente de trabajo resulta en

ocasiones restringido e inflexible; aspectos que recrudescen la búsqueda de información, la interfaz requiere ser más agradable, interesante y atractiva a la vista del usuario.

### **UCIMAT**

Este sistema fue implementado en el curso 2002/2003 con vistas a agilizar el censo de estudiantes. Permite búsquedas de estudiantes por determinados criterios, incluye la matrícula y modificación de los datos de los estudiantes, así como reportes generales de los datos que están matriculados en el sistema, impidiendo que se puedan realizar otras operaciones importantes entre los que se puede destacar el reingreso, registro de traslado u otros datos requeridos.

Este sistema desarrollado por profesores del centro, actualmente es utilizado y permite automatizar parte de la gestión académica de una universidad Cubana de forma general, gestiona la información académica de los estudiantes universitarios y la información de los profesores que forman parte del proceso docente educativo.

El sistema está concebido por módulos, entre los que se diferencian, los de actualización de datos y el sitio Web, a través del cual se muestran las diversas salidas de la aplicación.

### **1.2 Herramientas utilizadas para el desarrollo del sistema**

Entre las herramientas que actualmente se utilizan en el mundo por los desarrolladores, se encuentran fundamentalmente:

- SQL Server, MySQL y Oracle como gestores de bases de datos.
- Apache e IIS (Internet Information Service) como servidores Web.



Y se ha incorporado desde hace muy poco tiempo una muy potente herramienta: Visual Studio .NET que compite con el J2EE (Java Enterprise Edition).

Para el desarrollo del sistema se realizó un estudio sobre las posibles herramientas a utilizar en su construcción. Teniendo en cuenta las tendencias actuales y las novedades en este campo.

### **1.2.1 La tecnología .NET**

Microsoft.NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años, con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios, de forma tal que puedan ser suministrados remotamente, comunicándose y combinándose unos con otros totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados.

.NET ofrece un entorno de desarrollo de aplicaciones llamado *Visual Studio .NET* que consta de varios lenguajes de programación como Visual Basic .NET, Visual C#, Visual FoxPro y Visual C++ .NET. Estos lenguajes combinan las características de los lenguajes existentes con nuevas posibilidades para proporcionar un potente sistema de desarrollo. A continuación, se detallan algunas de las características de la Arquitectura .NET.

#### **1.2.1.1 Arquitectura Framework.NET**

La arquitectura .NET (.NET Framework) es el modelo de programación de la plataforma .NET para construir y ejecutar los servicios .NET. El objetivo de esta arquitectura es la de reducir la complejidad en el desarrollo de este tipo de

aplicaciones, permitiendo a los desarrolladores centrarse en escribir la lógica específica del servicio a desarrollar.

### 1.2.2 Lenguajes de programación

#### Lenguaje C#

Algunos de los lenguajes más utilizados en el campo de la ingeniería y la programación de sistemas son C y C++, pues proporcionan el nivel de abstracción preciso para construir una aplicación compleja, y ofrecen mecanismos de bajo nivel que permiten a los programadores hacer uso de las características más avanzadas de las plataformas sobre las que se ejecutan sus programas. Por el contrario, Java está lastrado por su compromiso con la portabilidad y es inherentemente ineficiente. Microsoft ha creado C# que combina algunas de las características más avanzadas de Java con algunas de las más potentes de C y C++, construido especialmente para adaptarse de manera natural al *framework*.

Las principales características que identifican al lenguaje C# son:

**Sencillez:** Elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET.

**Modernidad:** Incorpora en el propio lenguaje elementos que son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes hay que simular.

**Orientación a objetos:** Es más puro pues no admiten funciones ni variables globales, todo el código y datos han de especificarse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.

**Orientación a componentes:** Su sintaxis incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas.

**Seguridad de tipos:** Incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente.

**Eficiencia:** El código incluye numerosas restricciones para alcanzar su seguridad y no permite el uso de punteros. A diferencia de Java, en el mismo es posible saltarse dichas restricciones manipulando objetos a través de punteros.

### 1.2.3 ASP.NET

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

- **Mejor rendimiento:** Puede aprovechar las ventajas del enlace anticipado, la compilación just-in-time, la optimización nativa y los servicios de caché desde el primer momento. Esto supone un incremento espectacular del rendimiento antes de siquiera escribir una línea de código.
- **Compatibilidad con herramientas de primer nivel:** El marco de trabajo de ASP.NET se complementa con un diseñador y una caja de herramientas muy completos en el entorno integrado de programación (Integrated Development Environment, IDE) de Visual Studio.
- **Eficacia y flexibilidad:** 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 que mejor se adapte a la aplicación o dividirla en varios lenguajes.

## CAPÍTULO 1 “FUNDAMENTACIÓN TEÓRICA”

---

- **Simplicidad:** 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. No se requiere el reinicio del servidor, ni siquiera para implementar o reemplazar el código compilado en ejecución.
- **Escalabilidad y disponibilidad:** 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:** Permite insertar código en el nivel adecuado, siendo 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 completa seguridad de que las aplicaciones están a salvo.

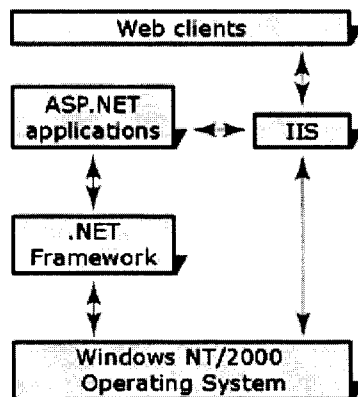


Figura 1 Arquitectura ASP.NET

Con la aparición del comercio electrónico se ha producido un incremento de la complejidad del desarrollo de este tipo de sistemas, lo cual supone un conjunto de desafíos para los desarrolladores, de los cuales pudiera citarse [3]:

- Implementación de interfaces Web enriquecidas.
- Separación del cliente y el servidor.
- Ejecución sin control de estado.
- Capacidades del cliente desconocidas.
- Complicaciones en el acceso a datos.
- Complicaciones con la escalabilidad.

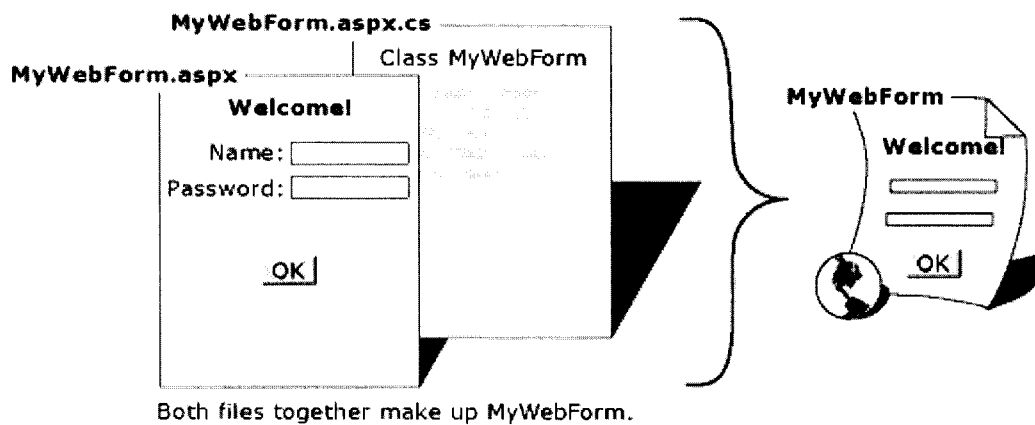


Figura 2 WebForms

La plataforma ASP.NET asume estos retos proporcionando a los desarrolladores las siguientes características:

- **Modelo de objetos intuitivo y consistente:** El marco de trabajo de las páginas ASP.NET presentan un modelo de objetos que permite concebir a los formularios como unidades, no como piezas separadas en el cliente

y en el servidor. Con este modelo, se programan las páginas en una forma más intuitiva que en las aplicaciones Web tradicionales, incluyendo la capacidad de establecer propiedades para los elementos del formulario y responder a eventos. Por otro lado, los controles del servidor de ASP.NET son una abstracción del contenido físico de una página HTML y de la interacción directa entre el navegador y el servidor. En sentido general, se pueden utilizar los controles del servidor de la misma forma en que pudiera trabajarse con los controles en aplicaciones clientes sin tener que pensar en como crear el HTML para presentar y procesar los controles y su contenido.

- **Modelo de programación dirigido por eventos:** Las páginas WebForms traen a las aplicaciones Web el familiar modelo de escribir manipuladores para eventos que ocurran tanto en el cliente como en el servidor. El marco de trabajo de ASP.NET abstrae este modelo de tal forma que el mecanismo subyacente de captura del evento en el cliente, su transmisión al servidor y la llamada a método apropiado es automática y transparente para el programador. El resultado es una clara y fácil estructura de código que soporta desarrollo dirigido por eventos.
- **Administración de estado intuitiva:** El marco de trabajo de ASP.NET automáticamente manipula la tarea de mantener el estado de la información específica de la aplicación. Esta es llevada a cabo sin un uso intensivo de los recursos del servidor y puede ser implementada con o sin el envío de cookies al navegador.
- **Aplicaciones independientes del navegador:** ASP.NET permite la creación de toda la lógica de la aplicación en el servidor, eliminado la necesidad de producir código para diferentes navegadores. Sin embargo, este aún permite que automáticamente se tome ventaja de características específicas de los diferentes navegadores mediante la escritura de código del lado del cliente para mejorar el rendimiento.

### 1.2.4 Modelación de aplicaciones Web sobre la plataforma ASP.NET

Para dar respuesta al problema de la modelación de aplicaciones Web aparece las extensiones de UML para Web propuestas por Jim Conallen en 1999 [2]. Esta iniciativa se basa en las características de UML de permitir extensiones del lenguaje mediante la utilización de valores etiquetados, estereotipos y restricciones para dotar a los diagramas de una nueva semántica propia del problema que se está modelando.

Estas exenciones parten de la separación entre componentes en el cliente y componentes en el servidor. De ahí la aparición de sus dos conceptos más importantes, la página cliente y la página servidora. A partir de esta idea aparecen los demás elementos de la extensión: Formularios, Frameset, Target, Script, XML, entre otros.

Como se vio en la sección anterior una de las ideas más innovadoras de al plataforma ASP.NET es la unificación en una sola unidad conceptual, el WebForm, de la lógica que corre tanto en el cliente como en el servidor, haciendo transparente al desarrollador la gestión de los eventos y del estado. Esto hace posible que las aplicaciones Web puedan modelarse tal y como se modelan las aplicaciones tradicionales basadas en ventanas, sin tener que utilizar las extensiones propuestas por Conallen que describen a las páginas cliente y servidora.

Basándose en lo expresado anteriormente, en el presente trabajo se ha optado por modelar las páginas como clases de interfaz tradicionales, sin hacer la distinción entre páginas clientes y páginas servidoras. Con esto se simplifica el diseño modelándose los elementos tal como serán utilizados en la etapa de implementación. Además se ha adoptado como política del proyecto global un conjunto de medidas para aprovechar al máximo las abstracciones de la plataforma ASP.NET, las cuales son:

- Eliminar la programación en el cliente. Aunque esto supone un deterioro del desempeño de la aplicación, elimina la necesidad de escribir código específico para cada uno de los navegadores más utilizados.
- Utilización intensiva de controles en el servidor. Esta medida apoya la anterior y permite dejarle a estos la generación del código específico para cada navegador
- No utilización de elementos ajenos a la plataforma como applets java y componentes ActiveX en el cliente.
- No utilización de marcos.

### 1.2.5 Programación Multicapas

Siguiendo la filosofía del modelo actual de desarrollo del software, para la realización del sistema se propone organizar los elementos de la aplicación en componentes independientes buscando alcanzar una mayor efectividad a la hora de administrarlos.

Para ello se seguirá los fundamentos de la programación en múltiples capas ya que esta además de facilitar una administración eficiente de los componentes que la integran, proporciona rapidez a todas las funcionalidades Cliente-Servidor y la magnitud de la aplicación lo exige. Tal y como plantea la arquitectura de esta tecnología, los componentes primarios de la aplicación serán divididos y programados por separados y en tiempo de ejecución serán unidos. De forma tal que si una de las capas definidas sufre cambios, no se vean afectados el resto de las capas ni el resultado final del producto.



## CAPÍTULO 1 “FUNDAMENTACIÓN TEÓRICA”

---

Se definen tres capas, tal es el caso de: Presentación, Reglas del Negocio y Acceso a Datos.

### **Presentación**

En esta capa se diseña todo lo que constituye la interfaz gráfica y la interacción del usuario con el sistema.

### **Reglas del Negocio**

Contiene todas las subrutinas creadas con el propósito de regular alguna acción del usuario.

### **Acceso a Datos**

En esta capa se programa todo lo que tiene que ver con el acceso a la base de datos. Esta capa queda encargada de tomar la información de la base de datos dada una petición de la capa de Reglas del Negocio, que a su vez es generada por la capa de presentación.



Figura 3 Aplicaciones de tres capas

### **Ventajas del modelo**

- Desarrollos paralelos (en cada capa)
- Aplicaciones más robustas debido al encapsulamiento
- Mantenimiento y soporte más sencillo (es más sencillo cambiar un componente que modificar una aplicación monolítica)
- Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad)
- Alta escalabilidad. La principal ventaja de una aplicación distribuida bien diseñada es su buen escalado, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware. El crecimiento es casi lineal y no es necesario añadir más código para conseguir esta escalabilidad.

### **1.2.6 Servicios Web (Web Services)**

Los Web Services son componentes software que permiten a los usuarios usar aplicaciones de negocio que comparten datos con otros programas modulares, vía Internet. Son aplicaciones independientes de la plataforma que pueden ser fácilmente publicadas, localizadas e invocadas mediante protocolos Web estándar, como XML, SOAP, UDDI o WSDL. El objetivo final es la creación de un directorio de online de Web Services, que pueda ser localizado de un modo sencillo y que tenga una alta fiabilidad.

El principal objetivo que se logra, es la interoperabilidad y la integración.

### 1.2.6.1 Protocolos que usan los Servicios Web

XML (Extensible Markup Language): Es un metalenguaje de marcas que permite definir cómo es la información que se transmite. Esto permite una comunicación de datos entre distintos sistemas. Es la base de los Servicios Web, y a pesar de su sencillez aparente, está transformando completamente la creación y el uso de software. Es la solución a un problema de comunicación entre programas de ordenador, pues la información generalmente queda fuertemente ligada al programa con el cual fue creada, y es así como se pierde mucho tiempo en pasar de un formato de definición a otro. El contenido almacenado en un documento XML se puede transferir fácilmente a través de la red.

SOAP (Single Object Access Protocol): Es un protocolo de mensaje liviano basado en XML, usado para codificar los mensajes de Web Services antes de enviarlos por la red. Los mensajes SOAP son independientes de cualquier sistema operativo y protocolo, y pueden ser transportados usando una variedad de protocolos de internet, incluyendo HTTP, SMTP y MIME. Permite que programas que corren en diferentes sistemas operativos se comuniquen.

WSDL (Web Service Description Language): Es un lenguaje en formato XML que define las operaciones que proporciona un servicio, desarrollado conjuntamente por Microsoft e IBM.

UDDI (Universal Description Discovery and Integration): Es un directorio universal de Servicios Web basado en XML, que permite publicar, localizar y utilizar servicios Web.

### **1.3 Sistemas Gestores de Base de Datos (SGDB)**

Un Sistema de Gestión de Bases de Datos (SGBD) Consiste en un conjunto de programas, procedimientos y lenguajes que proporcionan las herramientas necesarias para trabajar con una base de datos. Incorporar una serie de funciones que permitan definir los registros, sus campos, sus relaciones, insertar, suprimir, modificar y consultar los datos [4].

. En la actualidad existen numerosos sistemas gestores de bases de datos, entre ellos el Microsoft Access, Oracle, MySQL, Visual Fox Pro y SQL Server.

SQL Server posee un conjunto de utilidades y ventajas que lo hacen ser el más utilizado por los desarrolladores. De la misma forma se puede hacer referencia al oracle como otro de los sistemas gestores de bases de datos que más votos ha obtenido por parte de los desarrolladores.

Antes de comenzar a exponer algunas características de ambos gestores a modo de comparación, se puede decir que el Oracle es un manejador de base de datos relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información. Es el conjunto de datos que proporciona la capacidad de almacenar y acude a estos de forma consecuente con un modelo definido como relacional. Además es una suite de productos que ofrece una gran variedad de herramientas.[5].

#### **1.3.1 Catálogos de sistema de bases de datos**

Cada base de datos de Oracle ejecuta un catálogo de sistema centralizado, o diccionario de datos, que reside en la tabla SYSTEM. En Microsoft SQL Server 2000, las bases de datos conservan su propio catálogo de sistema, que contiene información acerca de:

## CAPÍTULO 1 “FUNDAMENTACIÓN TEÓRICA”

---

- Objetos de la base de datos (tablas, índices, procedimientos almacenados, vistas, desencadenadores, etc.).
- Restricciones.
- Usuarios y permisos.
- Tipos de datos definidos por el usuario.
- Definiciones de réplicas.
- Archivos utilizados por la base de datos.

SQL Server también guarda un catálogo de sistema centralizado en la base de datos **master**, que, además de los catálogos de sistema, almacena parte de la información de las bases de datos individuales, como:

- Nombres de bases de datos y ubicación del archivo principal de cada base de datos.
- Cuentas de inicio de sesión de SQL Server.
- Mensajes del sistema.
- Valores de configuración de la base de datos.
- Servidores remotos o vinculados.
- Información de la actividad actual.
- Procedimientos almacenados de sistema.

Al igual que con la tabla SYSTEM de Oracle, para tener acceso a cualquier otra base de datos, deberá estar disponible la base de datos **master** de SQL Server. Por lo tanto, se recomienda realizar una copia de seguridad de la base de datos **master** después de realizar un número de cambios significativo como medida de seguridad. Los administradores de la base de datos también pueden reflejar los archivos que componen la base de datos **master**.

Para obtener una lista detallada de las tablas del sistema incluidas en la base de datos **master** y en el resto de las bases de datos, consulte "Tablas de sistema" en los Libros en pantalla de SQL Server.

### 1.3.2 Estructuras de almacenamiento físico y lógico

El RDBMS de Oracle está compuesto por tablas que, a su vez, se componen de archivos de datos. Los archivos de datos de la tabla reciben formato como unidades internas denominadas *bloques*. El administrador de bases de datos (DBA) establece el tamaño del bloque cuando se crea por primera vez la base de datos de Oracle. Cuando se crea un objeto en una tabla de Oracle, el usuario puede indicar su espacio en unidades denominadas *extensiones* (inicial, siguiente, mínima y máxima). Si no se define explícitamente un tamaño de extensión, se crea una extensión predeterminada. El tamaño de una extensión de Oracle puede variar y debe contener una cadena de cinco bloques contiguos, como mínimo.

Microsoft SQL Server 2000 utiliza grupos de archivos en las bases de datos para controlar la ubicación física de las tablas e índices. Los grupos de archivos son contenedores lógicos de uno o varios archivos cuyos datos se completan proporcionalmente entre todos los archivos que componen el grupo.

Si no se han definido grupos de archivos, los objetos de la base de datos se guardan en el grupo de archivos predeterminado, creado implícitamente al crear la base de datos. Los grupos de archivos permiten:

- Distribuir grandes tablas entre varios archivos para aumentar el rendimiento de E/S.
- Almacenar índices en archivos independientes de sus tablas, para aumentar el rendimiento de E/S y la utilización simultánea del disco.
- Guardar columnas de tipo **text**, **ntext** e **image** (objetos grandes) en archivos independientes de la tabla.
- Almacenar objetos de base de datos en secciones específicas del disco.
- Realizar copias de seguridad y restaurar tablas individuales o grupos de tablas de un grupo de archivos.

SQL Server da formato a los archivos en unidades internas denominadas *páginas*. El tamaño de cada página es fijo: 8.192 bytes (8 KB). Las páginas están organizadas en extensiones fijas de ocho páginas contiguas. Al crear una tabla o un índice en una base de datos de SQL Server, se asigna automáticamente a una página de una extensión. A medida que se amplía una tabla o un índice, SQL Server le asigna automáticamente su propia extensión. De este modo, el almacenamiento de tablas pequeñas e índices es más efectivo que si se asignara la extensión completa, como sucede en Oracle.

Para obtener más información, consulte "Arquitectura de bases de datos físicas" en los Libros en pantalla de SQL Server.

### **1.3.3 Datos de bandas**

Los segmentos del tipo de Oracle no son necesarios en la mayoría de instalaciones de Microsoft SQL Server. En cambio, SQL Server puede distribuir o dividir en bandas más eficazmente los datos con un RAID basado en hardware o un RAID basado en software, disponibles mediante la utilidad Administrador de discos de Windows NT, o bien a través de otro fabricante. Con RAID puede configurar grupos de bandas, que se componen de varias unidades de disco mostradas como una unidad lógica. Si se crean archivos de base de datos en un conjunto de bandas, el subsistema de disco asume la responsabilidad de la distribución de la carga de E/S entre los diferentes discos. Se recomienda a los administradores que distribuyan los datos en varios discos físicos mediante RAID.

La configuración de una matriz RAID recomendada para SQL Server es RAID 1 (espejo) o RAID 5 (conjuntos de bandas con unidad de paridad adicional, para redundancia). Se recomienda también la opción RAID 10 (reflejo de conjuntos de bandas con paridad), aunque resulta mucho más cara que las dos anteriores.

Los conjuntos de bandas son apropiados para distribuir la E/S aleatoria realizada normalmente en los archivos de base de datos.

Los archivos de registro de transacciones deben optimizarse para la entrada y salida secuencial, y asegurarse sobre un punto de error único. Por lo tanto, se recomienda la opción RAID 1 (reflejo) para los registros de transacciones. Al realizar la migración, el tamaño de esta unidad debe ser, al menos, igual a la suma del tamaño de los registros de recuperación en conexión de Oracle más las tablas de segmentos de inversión de Oracle. Cree uno o varios archivos de registro que utilicen todo el espacio reservado para la unidad lógica. A diferencia de los datos almacenados en grupos de archivos, las entradas de los registros de transacción se guardan siempre en forma secuencial y no se rellenan proporcionalmente.

### 1.3.4 Registros de transacciones y recuperación automática

El RDBMS de Oracle realiza una recuperación automática cada vez que se inicia. Comprueba que el contenido de los archivos de la tabla esté coordinado con el contenido de los archivos de registro de recuperación en conexión. Si no lo está, Oracle aplica el contenido de los archivos de registro de recuperación en conexión a los archivos de la tabla (hacia adelante) y, a continuación, elimina las posibles transacciones no comprometidas que encuentre en los segmentos de inversión (hacia atrás). Si Oracle no puede obtener la información que precisa de los archivos de registro de recuperación en conexión, consulta los archivos de registro de recuperación archivados.

Microsoft SQL Server 2000 también efectúa una recuperación de datos automática mediante la comprobación de todas las bases de datos del sistema siempre que se inicia. En primer lugar, comprueba la base de datos **master** y, a continuación, lanza los subprocesos para recuperar el resto de las bases de datos del sistema. El mecanismo de recuperación automática comprueba el



archivo de transacciones de cada base de datos de SQL Server. Si este registro contiene transacciones sin confirmar, éstas se deshacen. El mecanismo de recuperación comprueba seguidamente en el registro de transacción las transacciones sin confirmar que aún no se hayan escrito en la base de datos. Si encuentra alguna, ejecuta de nuevo las transacciones y continúa.

Cada registro de transacciones de SQL Server tiene la funcionalidad combinada de un segmento de inversión de Oracle y uno de recuperación en conexión. Cada base de datos cuenta con un registro de transacciones propio que registra todos los cambios realizados en ella y es compartido por todos sus usuarios. Cuando se inicia una transacción que produce una modificación de los datos, se guarda en el registro un suceso BEGIN TRANSACTION, además del suceso de modificación. Este suceso se utilizará durante la recuperación automática para determinar el punto de inicio de la transacción. Según se reciben las instrucciones de modificación de los datos, se van escribiendo los cambios en el registro de transacciones, antes de hacerlo en la propia base de datos. SQL Server dispone de un mecanismo de control automático que garantiza que las transacciones completas se escriban regularmente de la caché de disco del servidor SQL Server al archivo de registro de transacciones. Un punto de control guarda en la base de datos cada página almacenada en la caché que ha sido modificada desde el último punto de control. Al establecer puntos de control en estas páginas mantenidas en la caché, también conocidas como páginas desfasadas, se asegura que todas las transacciones completas queden grabadas en el disco. Este proceso reduce el tiempo de recuperación necesario si se produjera un error en el sistema, como un corte del suministro eléctrico. Para cambiar este valor, modifique la configuración del intervalo de recuperación mediante el Administrador corporativo de SQL Server o el procedimiento almacenado del sistema **sp\_configure** de Transact-SQL.

### 1.3.5 Redes

Oracle SQL\*Net admite conexiones de red entre servidores de bases de datos de Oracle y sus clientes. Se comunica con el protocolo de flujo de datos de sustrato de red transparente (TNS, *Transparent Network Substrate*) y permite a los usuarios ejecutar diferentes protocolos de red sin necesidad de escribir código especializado.

Con Microsoft SQL Server, las Net-Libraries (librerías de red) admiten conexiones de red entre los clientes y el servidor mediante el uso del protocolo Secuencia de datos tabulares (TDS). Éstas permiten conexiones simultáneas entre clientes que utilizan Canalizaciones con nombre, sockets TCP/IP u otros mecanismos de comunicación entre procesos (IPC).

Específicamente del SQL Server se puede decir que es la oferta completa de base de datos y análisis. Tanto por la capacidad para consultar la base de datos mediante un explorador como por la compatibilidad con el Lenguaje de marcado extensible (XML, *Extensible Markup Language*), SQL Server 2000 es la base de datos totalmente habilitada para Web. Además, SQL Server 2000 ostenta marcas de referencia en cuanto a escalabilidad y confiabilidad, que son críticas para el éxito de una base de datos empresarial.

Tanto si lo que se mide es la velocidad en el desarrollo de aplicaciones como la velocidad del procesamiento de transacciones, SQL Server 2000 es la base de datos más rápida, lo que la convierte en la opción principal para la empresa que busca agilidad en sus operaciones. Una de las grandes ventajas de este producto, es que permite exportar o importar bases de datos completas, por lo que no es necesaria la creación de una nueva base de datos si se desea cambiar de gestor además proporciona las características necesarias para poner en marcha cualquier aplicación en el menor tiempo posible, destacándose en sus sencillas tareas de administración y en su capacidad de analizar la información. Las funciones definidas por el usuario, la integridad referencial en

casca y el depurador integrado de Transact SQL le permiten al programador la reutilización del código para simplificar el proceso de desarrollo.

### Utilidades

- **Almacén de datos:** Recopila bases de datos procedentes de diversas fuentes: Internet, Intranet.
- **Herramienta de Análisis:** A través del análisis de datos las empresas pueden adoptar posturas flexibles para responder a los cambios del sector.
- **Soporte para Internet:** Permite integrar cualquier aplicación informática para Internet, como las aplicaciones de Comercio Electrónico.

### Ventajas

- **Escalabilidad:** La flexibilidad de SQL Server hace posible responder a las nuevas necesidades de Internet que surjan en el futuro. Se adapta a las necesidades de la empresa, soportando desde unos pocos usuarios a varios miles. Empresas centralizadas u oficinas distribuidas, replicando cientos de sitios.
- **Eficiencia:** Los nuevos menús de tareas de Microsoft Windows XP facilitan la utilización de su sistema y sus programas, facilitando la realización de su trabajo diario.
- **Velocidad:** SQL Server está preparado para gestionar un tráfico de Comercio Electrónico muy elevado.
- **Seguridad:** SQL Server ofrece la mayor fiabilidad en la gestión de datos, Un único ID de login tanto para red como para la DB para mejorar la seguridad y facilitar la administración. Encriptación de procedimientos almacenados para la integridad y seguridad de código de aplicación.

- **Disponibilidad:** SQL Server aumenta la disponibilidad de las aplicaciones empresariales con el trasvase de registros, las copias de seguridad en línea y los clústeres de conmutación por error.
- **Potencia:** SQL Server posee los mejores registros en estadísticas tanto en transacciones totales como en coste por transacción.

### 1.4 RUP

Se hizo uso de las herramientas de la metodología RUP (*Rational Unified Process*) para facilitar el desarrollo del sistema.

El Proceso Unificado es un proceso de desarrollo de software que contiene un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software (Figura1). Más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, tipos de organizaciones, niveles de actitud y tamaños de proyecto. Está basado en componentes, lo cuál quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. Utiliza el *Lenguaje Unificado de Modelado* (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema software. Garantiza la elaboración de todas las fases de un producto de software orientado a objetos.

UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

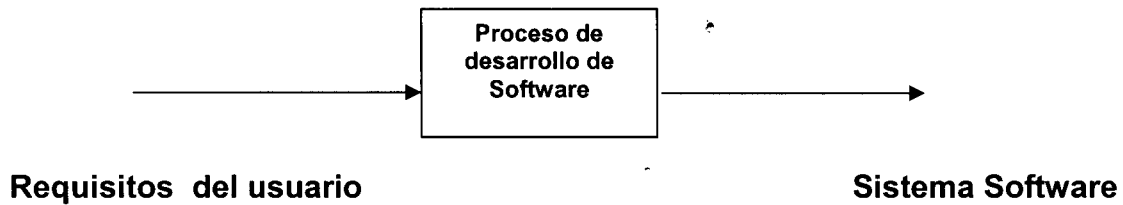


Figura 4. Un proceso de desarrollo de software.

### 1.4.1 Características del Proceso Unificado

Los verdaderos aspectos definitorios del Proceso Unificado, y que lo convierten en único, se resumen en tres frases clave - dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental.

- **Dirigido por los casos de uso:**

Teniendo en cuenta que la razón de ser de un sistema es brindar servicios a los usuarios, RUP define caso de uso como el conjunto de acciones que debe realizar un sistema para dar un resultado de valor a un determinado usuario y los utiliza tanto para especificar los requisitos funcionales del sistema, como para guiar todos los demás pasos de su desarrollo, dígame diseño, implementación y prueba.

- **Estar centrado en la arquitectura:**

La arquitectura es una vista del diseño completo con las características más importantes, dejando a un lado los detalles. Esta no solo incluye las necesidades de los usuarios e inversores, sino también otros aspectos técnicos como el hardware, sistema operativo, sistema de gestión de base de datos, protocolos de

red; con los que debe coexistir el sistema. En otras palabras, la arquitectura representa la forma del sistema, la cual va madurando en su interacción con los casos de uso hasta llegar a un equilibrio entre funcionalidad y características técnicas.

- **Ser iterativo e incremental:**

La alta complejidad de los sistemas actuales hace que sea factible dividir el proceso de desarrollo en varios mini-proyectos. Cada uno de estos mini-proyectos se les denomina iteración y pueden o no representar un incremento en el grado de terminación del producto completo. En cada iteración los desarrolladores seleccionan un grupo de casos de uso, los cuales se diseñan, implementan y prueban. La planificación de iteraciones hace que se reduzcan los riesgos de los costes de un solo incremento, no sacar al mercado un producto en el tiempo previsto, mantener la motivación del equipo pues puede ver avances claros a corto plazo y que el desarrollo pueda adaptarse a los cambios en los requisitos.

### **1.4.2 Rational Rose**

Es una herramienta para “modelado visual”, que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida del desarrollo de software. Permite completar una gran parte de las disciplinas (flujos fundamentales) del proceso unificado de Rational (RUP) e incluye un conjunto de herramientas de ingeniería inversa y generación de código que allanan el camino hasta el producto final.

### Conclusiones

Después de haber realizado un análisis de la información recopilada sobre los diferentes sistemas de gestión académica, se puede concluir que ningún sistema de los estudiados se adecua a las características requeridas por los objetivos del sistema propuesto. De la comparación entre los sistemas gestores de bases de datos SQL Server 2000 y Oracle se puede decir que para el sistema el SQL Server 2000 es el que más posibilidades brinda por sus características propias que lo hacen ideal para sistemas como el que se propone desarrollar.

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. Lo cual será utilizado para garantizar la integración del conjunto de aplicaciones que forman el sistema de Gestión Académica.

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 [7], de ahí que se utilicen estas características para el desarrollo de la aplicación.

Finalmente para el desarrollo del sistema se utilizarán las siguientes tecnologías por las características fundamentadas con anterioridad: la plataforma .NET de Microsoft (C# y ASP .NET), como Gestor de Base de Datos SQL Server, como metodología y aplicación para el diseño RUP y Racional Rose.

### **Capítulo #2 Características del sistema**

#### **Introducción**

En el presente capítulo se explica como es el uso de los registros académicos en el centro, se identifican las necesidades de los usuarios y se describe el objeto a automatizar, también se realiza un estudio de los sistemas docentes que tienen el mismo objetivo del que se propone.

Se presenta además la propuesta del sistema y se especifican los requerimientos funcionales y no funcionales. Se realiza la definición de los casos de uso, de los actores que intervienen en ellos y se determinan los casos de uso del primer ciclo de iteración.



### **2.1 Situación problemática**

En los centros de enseñanza del país, a no ser en algunos puntuales, la gestión docente se realiza de forma manual, aunque existen muchas actividades que se desarrollan de forma semi automatizada, como es el caso de la impresión de documentos finales de las notas de estudiantes, las planillas de solicitud de matrícula, las notas finales de los estudiantes en las asignaturas, etc.

La Universidad de Ciencias Informáticas de Cuba (UCI), es una institución de nueva creación, a la cual se le ha asignado una serie de recursos informáticos los cuales deben de reportar un alto nivel de automatización de los procesos de negocios de la misma. Actualmente se está trabajando en este sentido. El caso particular del registro académico se desarrolla en la actualidad totalmente de forma manual, aunque hay algunos reportes que se realizan con ayuda de herramientas informáticas, como es el caso del reporte de las notas de los estudiantes, el listado de estudiantes por grupos, etc.

Se desea automatizar las actividades que conciernen al registro de profesores en la UCI, el cual poseerá funcionalidades que faciliten la realización de estas operaciones y además contará con una interfaz amigable para los usuarios finales del sistema.

#### **2.1.1 Objetivos estratégicos de la organización y procesos de negocio que los soportan**

Dentro de los objetivos estratégicos de la organización se encuentra la formación docente de los estudiantes, la cual es su principal función, para lograr que todos los procesos relacionados con esta tarea se cumplan la institución a trazado una serie de estrategias que viabilizan el trabajo en esta rama y que permiten a los involucrados un uso eficiente y consiente de los recursos puestos a su disposición, la tendencia de estos procesos en este centro es hacia la

automatización, pues se ha trabajado bastante en la infraestructura informática, con la cual los usuarios cuentan con herramientas capaces de sustituir la labor manual por la automatizada, en la actualidad los procesos docentes se desarrollan en su mayoría de forma manual o semi automatizada, pues las herramientas existentes no son capaces de satisfacer todas las necesidades de los clientes para los que fueron creadas, esto trae consigo que no se le este dando un uso racional a los recursos con los cuales se cuenta y que se pierda la potencialidad que brinda toda esta infraestructura.

### **2.2 Objeto de automatización**

#### **2.2.1 Registro Académico**

Al profesor al inicio de un año lectivo se le entrega un registro académico en copia dura, en el cual se incluye el nombre, segundo nombre, primer apellido y segundo apellido de los estudiantes que conforman el grupo al que le va a impartir clases de una asignatura, el mismo cuenta además con casillas para registrar la inasistencia de los estudiantes y de otras en las cuales el profesor define el tipo de evaluación que realizó y las notas de estos estudiantes en la asignatura; si esta tiene examen final, con la información de la asistencia el profesor tiene argumentos para decidir si el estudiante tiene derecho a presentarse al examen final o no; si la asignatura no tiene examen final entonces el profesor puede decidir si el estudiante se presenta a examen mundial directo o no.

Con la información de las notas de las evaluaciones realizadas en la asignatura, el profesor puede conocer los estudiantes que presentan problemas docentes en su asignatura, además que tiene almacenados los resultados de sus estudiantes en las diferentes evaluaciones que haya realizado en su asignatura y puede obtener criterios para otorgar las notas finales de los estudiantes.

## **CAPÍTULO 2 “CARACTERÍSTICAS DEL SISTEMA”**

---

Con la información de asistencia y notas de los estudiantes en una asignatura, el profesor entrega a los directivos del centro un resumen de las notas de los estudiantes en las evaluaciones de su asignatura, así como los estudiantes que presentan problemas de asistencia.

Con esta misma información el profesor genera un resumen y se la entrega a los estudiantes, para que conozcan su situación docente, así como la cantidad de ausencias que presentan en las mismas y si estas son justificadas o no.

Todas estas actividades se desarrollan en la Universidad de Ciencias Informáticas de Cuba de forma semi automatizada, o sea, todo lo relacionado con la entrada al registro de las ausencias y notas de los estudiantes se desarrolla de forma manual, el profesor es el que escribe estos datos directamente en el registro; en el caso de los resúmenes de asistencia y notas, algunos generan documentos de Microsoft Word o de Microsoft Excel en los cuales introducen estos resultados y los entregan o los envían por correo electrónico a los interesados.

Esto es un breve acercamiento que permite tener una visión de cómo se desarrolla el trabajo con el Registro Académico en la Universidad de Ciencias Informáticas de Cuba.

Actualmente se cuenta con un sistema de gestión académica el cual incluye un módulo de asistencia y de evaluaciones, estos solo tiene función informativa, pues como se señalaba anteriormente este proceso en casi toda su magnitud se está desarrollando de forma manual; esto trae consigo el uso de grandes cantidades de papel para la creación de las copias de los registros que se le entregan a los profesores. A cada profesor se le entregan tantos registros como grupos de clase tenga, esto representa para el país un gran desperdicio de papel. El profesor para saber el por ciento de inasistencia de los estudiantes y el promedio que tienen en su asignatura tiene que calcularlo, pues lo que tienen en el registro son los datos brutos de estas dos cosas.

Por todo lo antes expuesto se decidió la implementación en la Universidad de Ciencias Informáticas, un Sistema de Registro Académico, el cual sea capaz de sustituir todas las tareas que se desarrollan de forma manual y además incluya otras funcionalidades optimizan los datos que se almacenan en él, con una interfaz fácil y amigable para los usuarios finales del sistema.

Para desarrollar todo lo que se propone en el sistema se cuenta con las herramientas necesarias, así como el apoyo de la institución y el soporte técnico necesario.

### **2.3 Información que se maneja**

El registro de profesores es un documento que cuenta con una parte donde se escriben los nombres completos de los estudiantes, pertenecientes a un grupo que recibirán docencia de una asignatura y con casillas en blanco para que el profesor registre las ausencias de los estudiantes a su turno de clases, así como otras casillas que están destinadas para registrar las notas que los estudiantes alcancen en las evaluaciones que realiza el profesor.

### **2.4 Propuesta de sistema**

El sistema que se propone presenta funcionalidades que ayudarán al trabajo de los involucrados en esta labor de la institución, pues el mismo mostrará una interfaz fácil y amigable donde estos tendrán las herramientas de trabajo necesarias para la gestión de las inasistencias y las notas de los estudiantes. Brindará información importante para los directivos del centro y para los estudiantes, pues estos podrán conocer en todo momento del curso el estado de asistencia y de notas de los estudiantes pertenecientes a un grupo académico en una asignatura.

### **2.4.1 Descripción general de la propuesta de sistema**

#### **2.4.1.1 Asistencia**

Para tomar la asistencia el profesor selecciona en la página principal del sistema la opción de registrar asistencia, este le muestra los grupos de clases a los que el profesor le imparte docencia para que seleccione el grupo y le muestra el listado de los estudiantes que pertenecen a dicho grupo, el profesor registra la asistencia para cada uno de los estudiantes y selecciona el tipo de inasistencia en caso de que la hubiera.

#### **2.4.1.2 Notas en las evaluaciones de una asignatura**

Para introducir las notas de los estudiantes en una asignatura, el profesor selecciona en la página principal del sistema la opción de registrar notas, este le muestra los grupos de clases a los que el profesor le imparte docencia para que el profesor seleccione el grupo y le muestra el listado de estudiantes pertenecientes a dicho grupo, además le da la opción al profesor de seleccionar el tipo de evaluación que va a realizar, para cada evaluación en una asignatura existe una forma de calificación y esta forma de calificación tiene valores de calificación que son al final los resultados que alcanza el estudiante en la evaluación, el profesor registra las notas para cada estudiante.

#### **2.4.1.3 Estado de los estudiantes**

Los estados que se pueden observar de los estudiantes son, de asistencia y académico; el interesado selecciona el grupo y la asignatura en los que desea observar el estado de los estudiantes y se le muestran los estudiantes pertenecientes al grupo de clases seleccionado con el estado de asistencia o docente en esa asignatura.

### **2.4.2 Otras soluciones existentes en el centro**

En el centro existe un sistema de gestión académica, el cual fue desarrollado por la universidad de Las Villas, el mismo no está desarrollado para un centro con las características de la UCI, pues hay varios conceptos que se manejan en el mismo que son totalmente distintos para las dos universidades, como es el caso de la matrícula de los estudiantes por años, las asignaturas fijas para un año de clases, las formas de evaluación y de calificación fijas y además no presenta ningún tipo de íter conectividad con los demás sistemas del centro, por esto este sistema cumple, hasta el momento una función informativa, pues solo es utilizado por los docentes para la publicación la asistencia y las notas de los estudiantes a sus turnos de clases.

Se están explotando además los sistemas de Asistencia y de evaluaciones, en los cuales el profesor introduce la información referente a estos temas para los estudiantes de los grupos a los que le imparte docencia, estos sistemas también tienen función informativa, pues los interesados acceden a ellos y observan el estado de los estudiantes en estos aspectos, estos tampoco presentan íter conectividad con los demás sistemas del centro.

### 2.5 Especificación de los requisitos de software

#### 2.5.1 Requisitos funcionales

1. Registrar asistencia de estudiantes.
  - 1.1 El profesor registra la inasistencia de los estudiantes a su turno de clases.
  - 1.2 El tipo de inasistencia puede ser:
    - 1.2.1 Justificada.
    - 1.2.2 Injustificada.
2. Registrar notas de estudiantes.
  - 2.1 El profesor especifica el tipo de evaluación a realizar
    - 2.1.1 Las evaluaciones pueden ser:
      - 2.1.1.1 Evaluaciones parciales
        - 2.1.1.1.1 El profesor registra la nota de los estudiantes
      - 2.1.1.2 Evaluaciones Frecuentes.
        - 2.1.1.2.1 El profesor registra las notas de los estudiantes.
3. Mostrar estado del estudiante.
  - 3.1 El usuario solicita el estado del estudiante en una asignatura.
  - 3.2. Los estados pueden ser:
    - 3.2.1 Estado de asistencia de los estudiantes.
    - 3.2.2 Estado académico de los estudiantes.

3.3 Se muestra el estado de los estudiantes.

4. Mostrar estudiantes con problemas

4.1 Se solicitan los datos de los estudiantes con problemas.

4.3 Los problemas pueden ser:

4.3.1 Problemas de asistencia.

4.3.2 Problemas docentes.

4.4 Se muestran los estudiantes con problemas.

5. Enviar alertas

5.1 Las alertas se pueden enviar a profesores y estudiantes por:

5.1.1 Problemas de asistencia.

5.1.1.1 El estudiante recibe una alerta donde se le informa de problemas de asistencia en una asignatura.

5.1.2 Problemas docentes.

5.1.2.1 El estudiante recibe una alerta donde se le informa de problemas docentes en una asignatura.

5.1.3 Necesidad de realizar evaluación.

5.1.3.1 El profesor recibe una alerta donde se le informa de la necesidad de realizar una evaluación en un grupo académico.

6. Configurar Registro

6.1 El usuario especifica los parámetros a configurar



6.1.1 Los parámetros pueden ser:

6.1.1.1 configurar alertas.

6.1.1.1.1 El usuario especifica:

6.1.1.1.1.1 Contenido de la alerta.

6.1.1.2 configurar % de asistencias en una asignatura.

6.1.1.2.1 El usuario especifica el nuevo % de asistencias para una asignatura.

6.1.1.3 configurar nota mínima para tener aprobada una asignatura.

6.1.1.3.1 el usuario especifica la nueva nota mínima para el aprobado en una asignatura.

6.1.1.4 configurar los colores con los que se van a colorear los estudiantes con problemas de asistencia y docentes.

6.1.1.4.1 El usuario especifica los nuevos colores.

7. Generar reporte final de notas

7.1 La secretaria especifica:

7.1 formato del documento final de notas.

7.2 la secretaria genera el reporte final de notas

7.2.1 el reporte final de notas se envía al subsistema de expediente académico.

### **2.5.2 Requisitos no funcionales**

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Para definirlos se tienen en cuenta propiedades de factibilidad, usabilidad, confiabilidad.

#### **2.5.2.1 Requisitos de Interfaz o apariencia**

- Interfaz con un diseño sencillo que contenga pocos gráficos, con vista a acelerar la velocidad de respuesta hacia el usuario debido a la complejidad de los procesos llevados a cabo en Akademos.
- La interfaz debe limitarse a presentar las funcionalidades netamente de Akademos logrando la concentración del usuario en las tareas que esté realizando.

#### **2.5.2.2 Requisitos de usabilidad**

- Documentar bien la aplicación y proporcionar materiales de ayuda para hacer mejor uso de todos los servicios que este ofrece.

#### **2.5.2.3 Requisitos de Rendimiento**

- El tiempo de respuesta de cada uno de las páginas debe ser menor que un minuto, excepto en aquellas que por las actividades que realizan, requieran más tiempo, tal es el caso de la generación de Horarios y la Formación de grupos administrativos.
- Estará implementado sobre una tecnología Web, facilitando su uso a través de la red.

#### **2.5.2.4 Requisitos de Soporte**

- El sistema debe ser integrado, de manera que se mantenga la colaboración entre todos los subsistemas que lo componen.

## **CAPÍTULO 2 “CARACTERÍSTICAS DEL SISTEMA”**

---

- La base de datos que utilizará el sistema como medio de almacenamiento de la información estará soportada sobre un gestor de bases de datos SQLServer, permitiéndole interactuar con otros sistemas estableciendo vías de compatibilidad.
- La capa que contiene las Reglas del Negocio debe estar sobre IIS en Windows.
- Lograr la solidez de los datos realizando mantenimientos automatizados en la base de datos, orientados a la actualización y corrección de la información, a horas del día donde haya la menor cantidad de usuarios conectados.

### **2.5.2.5 Requisitos de Seguridad**

- Debe contar con varios niveles de acceso para permitir el trabajo organizado con el sistema.
- El acceso se realizará por el Subsistema de Seguridad, integrado a los todos los subsistemas de Gestión Académica. El mismo permitirá la gestión de usuarios y roles así como el acceso de estos mediante el directorio activo en la UCI.
- Toda la información guardada de los estudiantes será confidencial, por lo que solo podrá ser consultada por personal autorizado y estará prohibida su divulgación.

### **2.6 Definición de los casos de uso (ver anexo 1)**

### **Conclusiones**

En este capítulo se trató el tema del problema, la situación problémica que da origen al desarrollo de la solución, el objeto de automatización, una breve descripción de la solución del problema, los requerimientos funcionales y no funcionales que se deben cumplir para el desarrollo de la aplicación y el diagrama de los casos de uso del sistema.

### Capítulo #3 Análisis y diseño del sistema

#### Introducción

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.

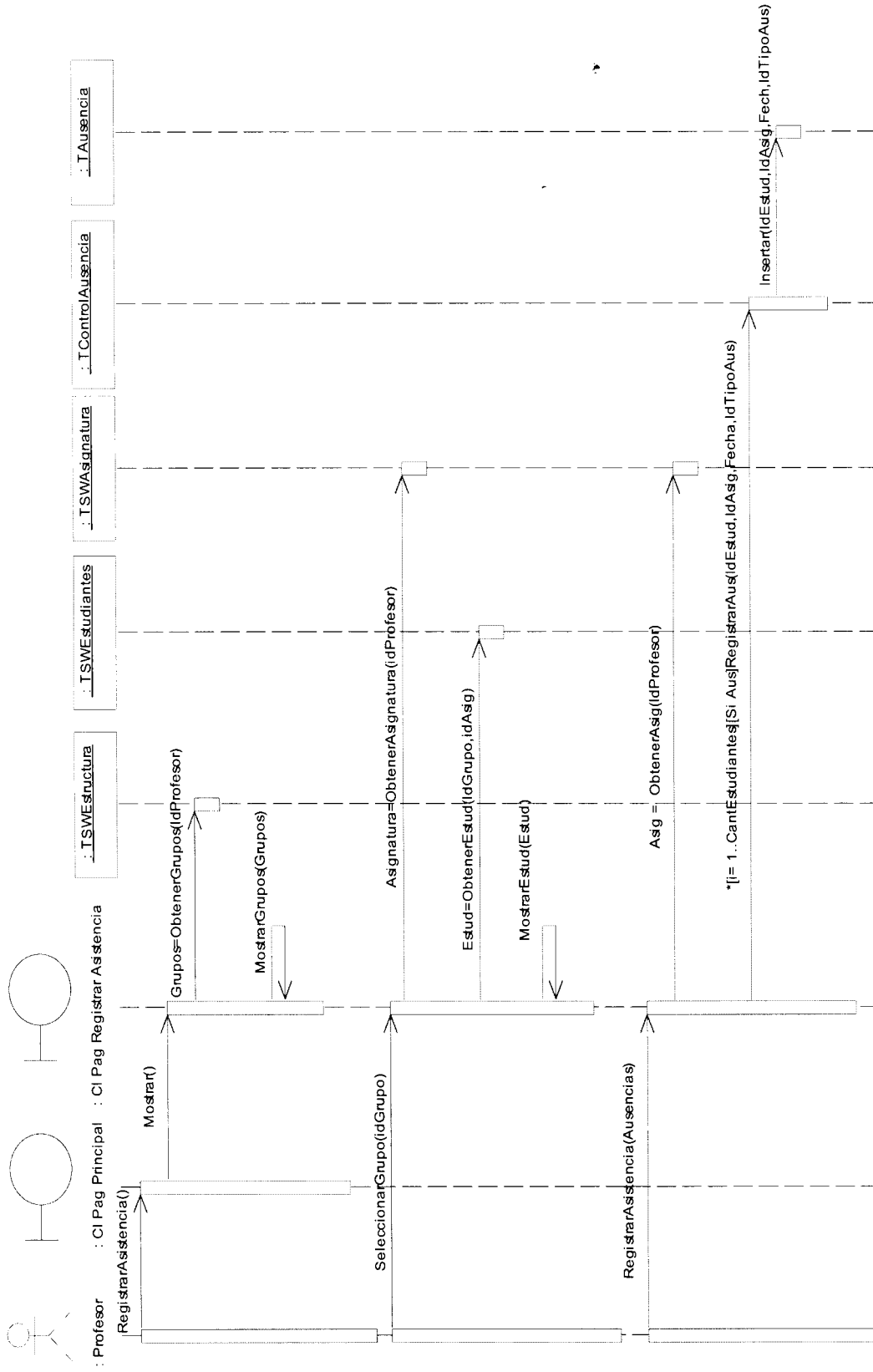
En la presente documentación no se han utilizado las extensiones de UML para Web pues se ha aprovechado la abstracción que supone la utilización de la plataforma ASP.NET mediante el uso de los WebForms, A diferencia de las extensiones Web los vínculos entre las páginas se han modelado como asociaciones entre clases de interfaz. Tampoco se han representado los formularios en las páginas ni sus respectivos elementos asumiendo que la página completa es un formulario.



**3.2 Diagramas de interacción del sistema.**

**3.2.1 Diagrama de secuencia del caso de uso “Registrar asistencia”,  
sección “Registrar asistencia”**

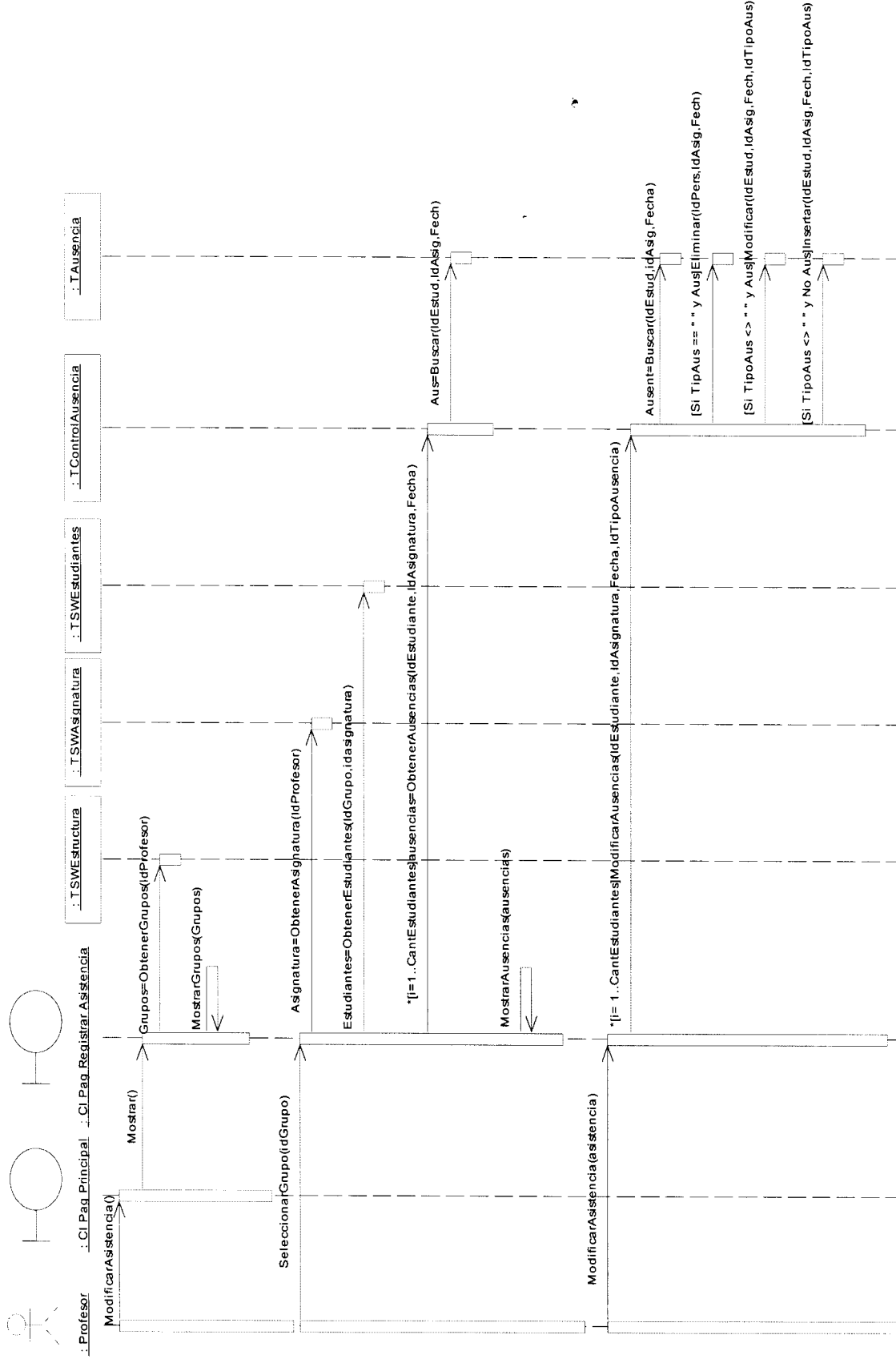
# CAPÍTULO 3 “ANÁLISIS Y DISEÑO DEL SISTEMA”





**3.2.2 Diagrama de secuencia del caso de uso “Registrar asistencia”,  
sección “Modificar Asistencia”**

# CAPÍTULO 3 “ANÁLISIS Y DISEÑO DEL SISTEMA”

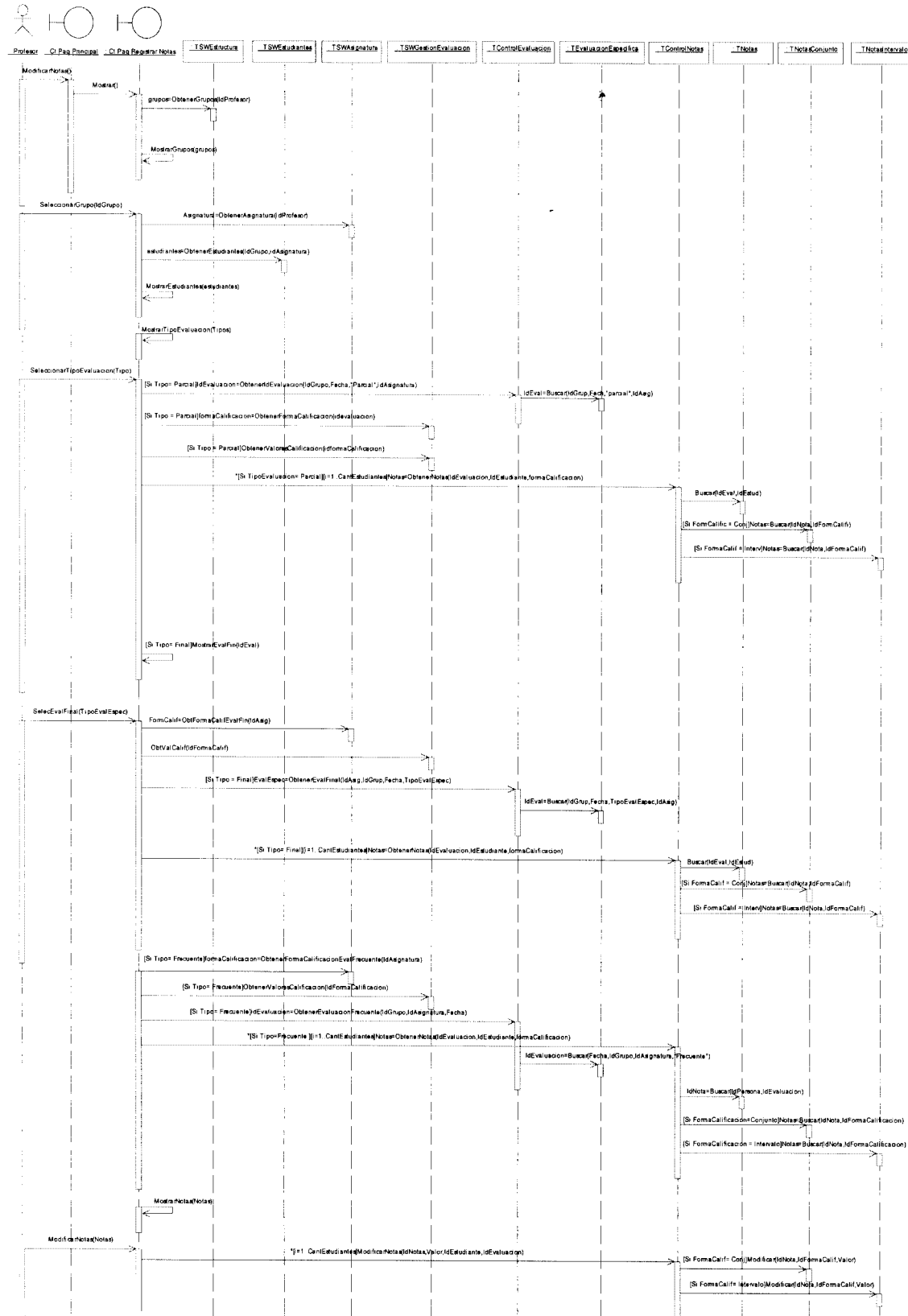


**3.2.3 Diagrama de secuencias del caso de uso “Registrar notas”, sección “Registrar notas”**



**3.2.4 Diagrama de secuencia del caso de uso “Registrar notas”, sección “Modificar notas”**

# CAPÍTULO 3 “ANÁLISIS Y DISEÑO DEL SISTEMA”



**3.2.5 Diagrama de secuencia del caso de uso “Mostrar estado”, sección “Mostrar Estado”**

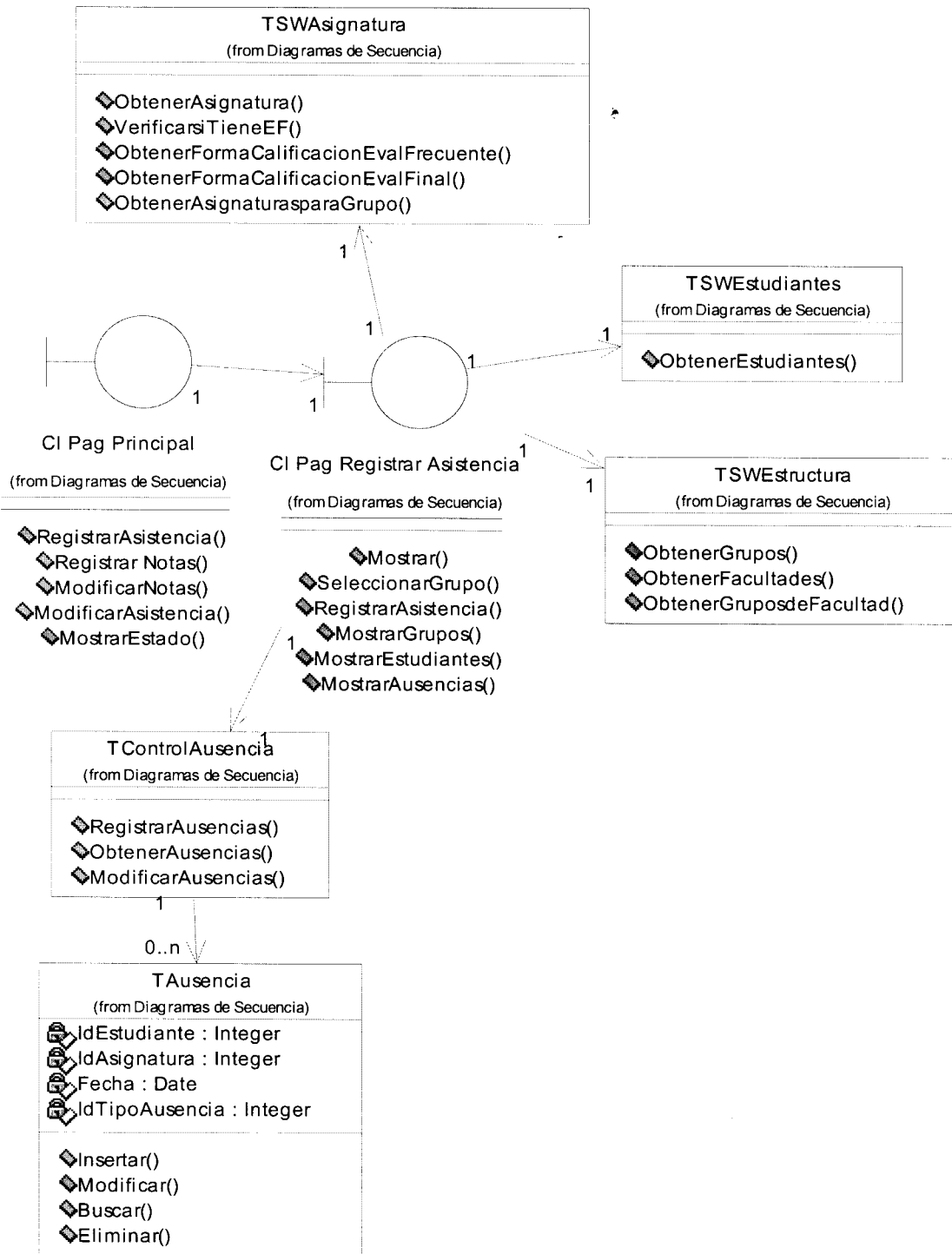




**3.3 Diagramas de clases de diseño**

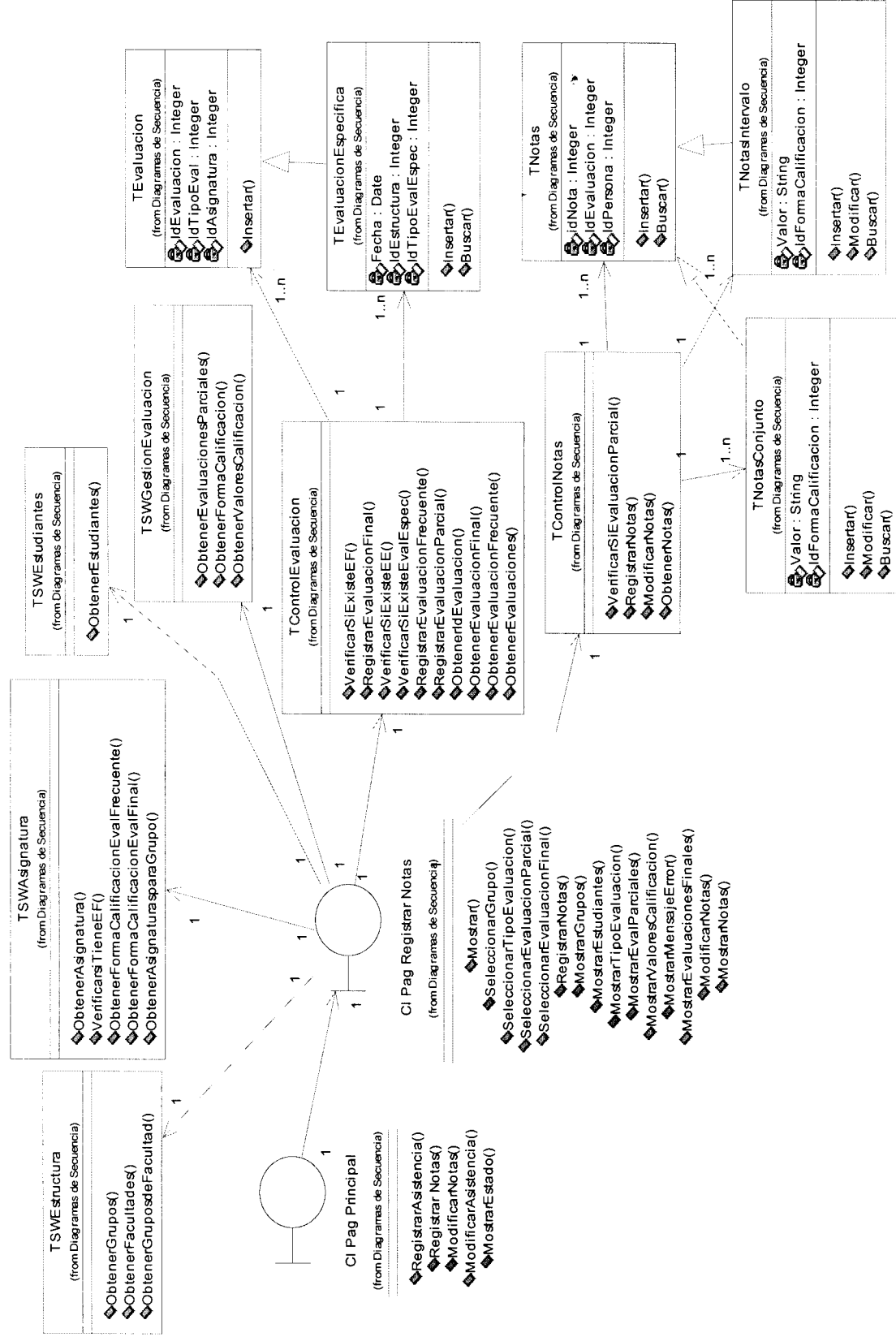
**3.3.1 Diagrama de clases de diseño del caso de uso “Registrar asistencia”**

# CAPÍTULO 3 “ANÁLISIS Y DISEÑO DEL SISTEMA”



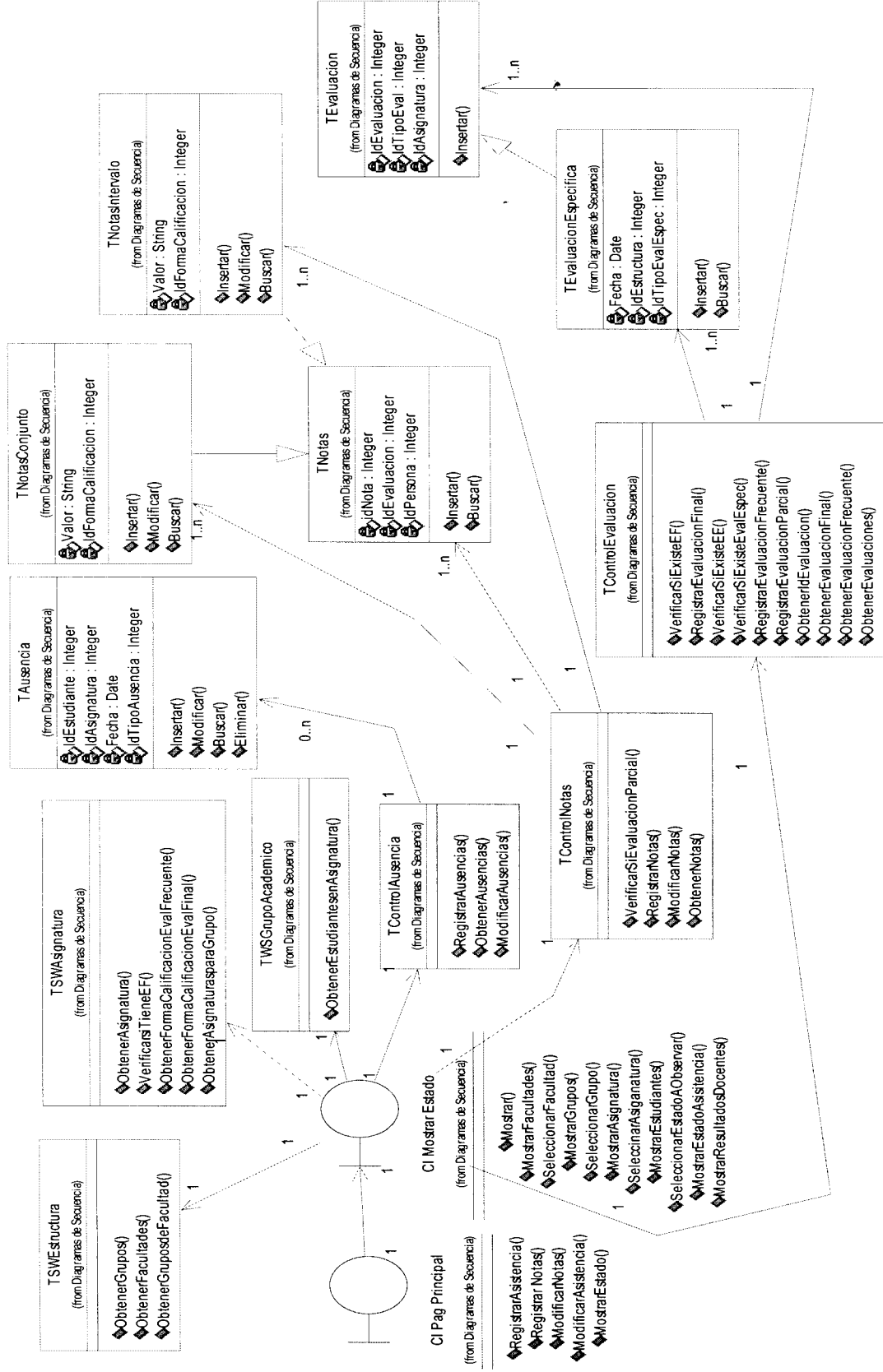
**3.3.2 Diagrama de clases de diseño del caso de uso “Registrar notas”**

# CAPÍTULO 3 “ANÁLISIS Y DISEÑO DEL SISTEMA”



**3.3.3 Diagrama de clases de diseño del caso de uso “Mostrar estado”.**

# CAPÍTULO 3 “ANÁLISIS Y DISEÑO DEL SISTEMA”



**3.4 Descripción de las clases**

<b>Nombre:</b> TControlAusencia	
<b>Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad</b>	
Nombre:	TControlAusencia()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	RegistrarAusencias(int IdEstudiante,int IdAsignatura, int IdTipoAusencia)
Descripción:	Método que permite registrar la asistencia de los estudiantes a un turno de clases.
Nombre:	ObtenerAusencias(int IdEstudiante,int IdAsignatura, string Fecha)
Descripción:	Método que permite obtener las ausencias del los estudiantes de un grupo en una fecha.
Nombre:	ModificarAusencias(int IdEstudiante,int IdAsignatura,string Fecha,int IdTipoAusencia)
Descripción:	Método que permite modificar las ausencias de los estudiantes de un grupo, en una fecha.

Tabla 1 Descripción de la clase TControlAusencia.

<b>Nombre:</b> TAusencia	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Fecha	string
IdAsignatura	int
IdEstudiante	int
IdTipoAusencia	int
<b>Para cada responsabilidad</b>	
Nombre:	TAusencia()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	Insertar(int IdEstudiante, int IdAsignatura,int IdTipoAusencia,string Fecha)
Descripción:	Método que permite insertar la inasistencia los estudiantes a un turno de clases de una asignatura.
Nombre:	Buscar(int IdEstudiante, int IdAsignatura, string Fecha)
Descripción:	Método que permite buscar las ausencias de los estudiantes a los turnos de clases de una asignatura.
Nombre:	Eliminar(int IdEstudiante,int IdAsignatura, string Fecha)
Descripción:	Método que permite eliminar la ausencia de los estudiantes a un turno de clases de una asignatura.
Nombre:	Modificar(int IdEstudiante,int Asignatura, int Fecha)
Descripción:	Método que permite modificar las ausencias de los estudiantes a un turno de clases de una asignatura.

Tabla 2 Descripción de la clase TAusencia.



### CAPÍTULO 3 “ANÁLISIS Y DISEÑO DEL SISTEMA”

<b>Nombre:</b> TControlEvaluacion	
<b>Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad</b>	
Nombre:	TControlEvaluacion()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	RegistrarEvaluacionParcial(int IdEvaluacion,int IdGrupo, int Fecha)
Descripción:	Método que permite registrar una evaluación parcial.
Nombre:	VerificarSiExisteEF(int IdAsignatura,int IdTipoevaluacionEspecificas, int IdGrupo)
Descripción:	Método que permite verificar si hay un examen final registrado para esa asignatura en ese grupo, devuelve un bool.
Nombre:	RegistrarEvaluacionFinal(int IdAsignatura,int IdTipoEvaluacionEspecificas,string Fecha, int IdGrupo)
Descripción:	Método que permite registrar una evaluación final de una asignatura realizada en una fecha en un grupo.
Nombre:	VerificarSiExisteEE(int IdAsignatura,int TipoEvaluacionEspecificas,int IdGrupo)
Descripción:	Método permite verificar si se ha realizado un examen extraordinario de esa asignatura en ese grupo, devuelve un bool.
Nombre:	RegistrarEvaluacionFrecuente(int IdEstudiante,int IdAsignatura, int TipoEvaluacionEspecificas,string Fecha )
Descripción:	Método que permite registrar una evaluación frecuente de

### CAPÍTULO 3 “ANÁLISIS Y DISEÑO DEL SISTEMA”

	una asignatura, de una estudiante perteneciente a un grupo en una fecha.
--	--

Tabla 3 Descripción de la clase TControlEvaluacion.

<b>Nombre:</b> TEvaluacion	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Id	Int
IdTipoevaluacion	Int
IdAsignatura	Int
<b>Para cada responsabilidad</b>	
Nombre:	TEvaluacion()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	Insertar(int IdAsignatura,int IdTipoEvaluacion)
Descripción:	Método que permite registrar una evaluación en una asignatura.

Tabla 4 Descripción de la clase TEvaluacion

<b>Nombre:</b> TEvaluacionEspecificas(hereda de TEvaluacion)	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Fecha	string
IdEstructura	Int
IdTipoEvaluacionEspecificas	Int
<b>Para cada responsabilidad</b>	

### CAPÍTULO 3 “ANÁLISIS Y DISEÑO DEL SISTEMA”

Nombre:	TEvaluacionEspecifica()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	Insertar(int Id,int IdEstructura, int IdTipoEvaluacionEspecifica)
Descripción:	Método que permite registrar una evaluación específica realizada en un grupo.
Nombre:	Buscar(string Fecha, int IdEstructura, int IdAsignatura, int TipoEvaluacionEspecifica)
Descripción:	Método que permite buscar una evaluación específica que haya sido registrada.

Tabla 5 Descripción de la clase TEvaluacionEspecifica

<b>Nombre:</b> TControlNotas	
<b>Clase controladora</b>	
<b>Para cada responsabilidad</b>	
Nombre:	TControlNotas()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	VerificarSiEvaluacionParcial(IdEvaluacion,IdGrupo)
Descripción:	Método que permite saber si la evaluación parcial tiene notas en ese grupo.
Nombre:	RegistrarNotas(int IdEstudiante,int IdEvaluacion, int valor, int FormaCalificacion)
Descripción:	Método que permite Registrar las notas de los estudiantes en una evaluación de una asignatura.

### CAPÍTULO 3 “ANÁLISIS Y DISEÑO DEL SISTEMA”

---

Tabla 6 Descripción de la clase TControlNotas

<b>Nombre:</b> TNotas	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Id	Int
IdEvaluacion	Int
IdEstudiante	Int
<b>Para cada responsabilidad</b>	
Nombre:	TNotas()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	Insertar(int IdEvaluacion, int IdEstudiante)
Descripción:	Método que permite registrar una nota de una persona en una evaluación.

Tabla 7 Descripción de la clase TNotas.

<b>Nombre:</b> TNotasConjunto(hereda de TNotas)	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Valor	Int
IdFormaCalificacion	Int
<b>Para cada responsabilidad:</b>	
Nombre:	TNotasConjunto()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	Insertar(int Id, int Valor, int IdFormaCalificacion)

### CAPÍTULO 3 “ANÁLISIS Y DISEÑO DEL SISTEMA”

Descripción:	Permite registrar una nota de conjunto de una estudiante en una evaluación.
Nombre:	Modificar(int IdNota,int IdFormaCalificacion,int Valor)
Descripción:	Método que permite modificar las notas de conjunto de los estudiantes en una evaluación.
Nombre:	Buscar(int IdNota, int IdFormaCalificacion)
Descripción:	Método que permite buscar las notas de conjunto de los estudiantes.

Tabla 8 Descripción de la clase TNotasConjunto.

<b>Nombre:</b> TNotasIntervalo(hereda de TNotas)	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Valor	Int
IdFormaCalificacion	Int
<b>Para cada responsabilidad</b>	
Nombre:	TNotasIntervalo()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	Insertar(int Id, int Valor, int IdFormaCalificacion)
Descripción:	Permite registrar una nota de intervalo de una estudiante en una evaluación.
Nombre:	Modificar(int IdNota,int IdFormaCalificacion,int Valor)
Descripción:	Método que permite modificar las notas de intervalo de los estudiantes en una evaluación.

### CAPÍTULO 3 “ANÁLISIS Y DISEÑO DEL SISTEMA”

---

Nombre:	Buscar(int IdNota, int IdFormaCalificacion)
Descripción:	Método que permite buscar las notas de intervalo de los estudiantes.

Tabla 9 Descripción de la clase TNotasIntervalo.

<b>Nombre: CIPagRegistrarAsistencia</b>	
<b>Clase Interfaz</b>	
<b>Para cada responsabilidad</b>	
Nombre:	Mostrar()
Descripción:	Muestra la página registrar asistencia.
Nombre:	SeleccionarGrupo(int IdGrupo)
Descripción:	Método que permite al profesor seleccionar el grupo de clases.
Nombre:	RegistrarAsistencia(Object Asistencia)
Descripción:	Método que permite al profesor registrar la asistencia
Nombre:	
Descripción:	

Tabla 10 Descripción de la clase CIPagRegistrarAsistencia.

<b>Nombre: CIPagRegistrarNotas</b>	
<b>Clase Interfaz</b>	
<b>Para cada responsabilidad</b>	
Nombre:	Mostrar()
Descripción:	Muestra la página registrar notas.
Nombre:	SeleccionarGrupo(int IdGrupo)
Descripción:	Método que permite al profesor seleccionar el grupo de clases.
Nombre:	SeleccionarTipoEvaluacion(Object Tipo)
Descripción:	Permite al profesor seleccionar el tipo de evaluación que va a realizar.
Nombre:	SeleccionarEvaluacionParcial(int IdEvaluacion)
Descripción:	Permite al profesor seleccionar la evaluación parcial a realizar.
Nombre:	SeleccionarEvaluacionFinal(Object TipoEvaluacionEspecifica)
Descripción:	Permite al profesor seleccionar el tipo de evaluación específica a realizar.
Nombre:	RegistrarNotas(int Valor)
Descripción:	Permite al profesor registrar notas.

Tabla 11 Descripción de la clase CIPRegistrarNotas.

<b>Nombre: CIPagMostrarEstado</b>	
<b>Clase Interfaz</b>	
<b>Para cada responsabilidad</b>	
Nombre:	Mostrar()
Descripción:	Muestra la página mostrar estado.
Nombre:	SeleccionarFacultad(int IdFacultad)
Descripción:	Método que permite al usuario seleccionar la facultad.
Nombre:	SeleccionarGrupo(int IdGrupo)
Descripción:	Permite al usuario seleccionar el grupo.
Nombre:	SeleccionarAsignatura(int IdAsignatura)
Descripción:	Permite al usuario seleccionar asignatura.
Nombre:	SeleccionarEstadoAObservar(Object Estado)
Descripción:	Permite al usuario seleccionar estado del estudiante, que desea observar.

Tabla 12 Descripción de la clase CIPMostrarEstado.



3.5 Diseño de la base de datos

3.5.1 Diagrama de entidad relación

Este diagrama de entidad relación, conforma solo la parte del subsistema Registro Virtual, como se puede observar en las tablas hay llaves foráneas que pertenecen a relaciones con otras tablas del sistema Akademos, las cuales pertenecen a módulos del mismo, del cual el sistema obtiene y ofrece información.

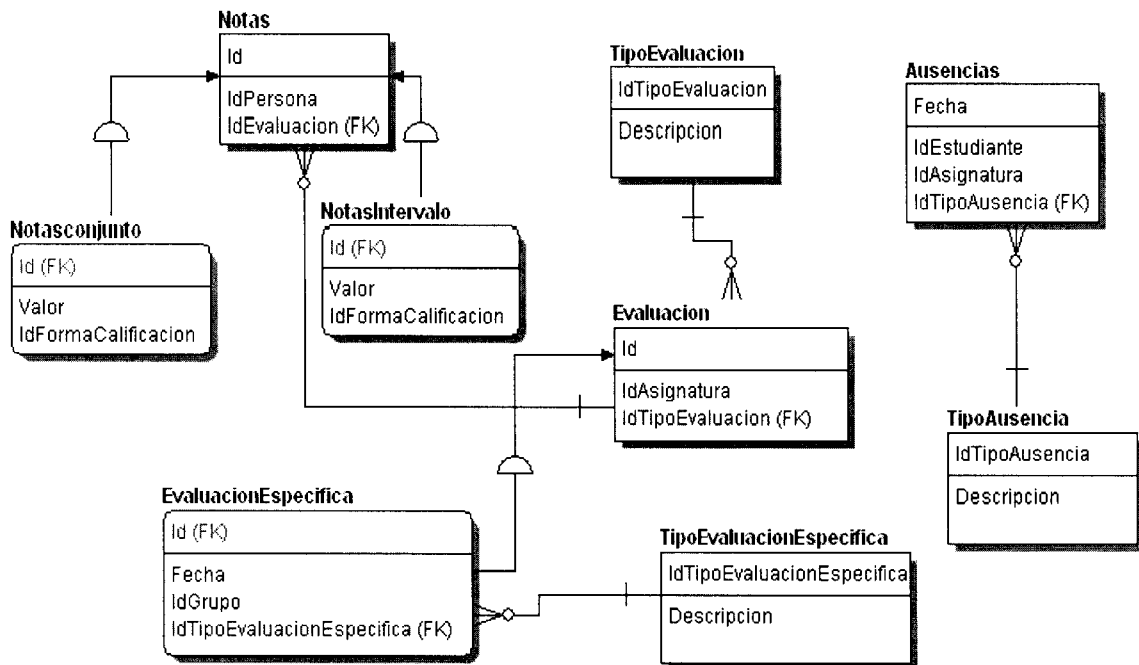


Figura 1 Modelo lógico de datos de la base de datos.

**3.5.2 Descripción de las tablas de la base de datos**

<b>Ausencia</b>		
En esta tabla se almacenan las ausencias de los estudiantes a los turnos de clases		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Fecha	Date	En este campo se almacena la fecha de la ausencia.
IdAsignatura	Integer	Este campo es una llave extranjera y almacena el identificador de la asignatura a la que estuvo ausente el estudiante.
IdEstudiante	Integer	Este campo es una llave extranjera y almacena el identificador del estudiante que estuvo ausente a un turno de clases de una asignatura.
IdTipoAusencia	Integer	Este campo es una llave extranjera y almacena el identificador del tipo de ausencia que presento el estudiante.

Tabla 13 Descripción de la tabla Ausencia.

<b>TipoAusencia</b>		
En esta tabla se almacenan los tipos de ausencias que puede presentar un estudiante.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
IdTipoAusencia	Counter	Es un auto numérico, para identificar el tipo de ausencia.
Tipo	VarChar	Es el tipo de ausencia que puede presentar una estudiante

Tabla 14 Descripción de la tabla TipoAusencia.

<b>Evaluacion</b>		
En esta tabla se almacenan las evaluaciones que se le realizan a los estudiantes		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id	Counter	Es un auto numérico, que identifica a las evaluaciones que se registran.
IdTipoEvaluacion	Integer	Este campo es una llave extranjera y almacena el identificador del tipo de evaluación que se registró.
IdAsignatura	Integer	Este campo es una llave extranjera y almacena el identificador de la asignatura a la que estuvo ausente el estudiante.

Tabla 15 Descripción de la tabla Evaluacion

<b>EvaluacionEspecificas</b>		
En esta tabla se almacenan las evaluaciones específicas que se le realizan a los estudiantes		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id	Integer	Este campo es una llave extranjera y almacena el identificador de las evaluaciones que se han registrado.
Fecha	Date	Este campo almacena la fecha donde se registró la evaluación.
IdEstructura	Integer	Este campo es una llave extranjera y almacena el identificador del grupo al que se le registró la evaluación.
IdTipoEvalEspec	Integer	Este campo es una llave extranjera y almacena el identificador de la evaluación específica que se registró.

Tabla 16 Descripción de la tabla EvaluacionEspecificas.

### CAPÍTULO 3 “ANÁLISIS Y DISEÑO DEL SISTEMA”

---

<b>TipoEvalEspecifica</b>		
En esta tabla se almacenan los tipos de evaluaciones específicas		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
IdTipoEvalEspec	Counter	Este campo es un auto numérico que identifica los tipos de evaluación específica.
Descripción	VarChar	En este campo se almacenan los tipos de evaluaciones específicas.

Tabla 17 Descripción de la tabla TipoEvalEspecifica.

<b>TipoEvaluacion</b>		
En esta tabla se almacenan los tipos de evaluaciones específicas		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
IdTipoEval	Counter	Este campo es un auto numérico que identifica los tipos de evaluaciones.
Descripción	VarChar	En este campo se almacenan los tipos de evaluaciones.

Tabla 18 Descripción de la tabla TipoEvaluacion.

<b>Notas</b>		
En esta tabla se almacenan los tipos de evaluaciones específicas		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id	Counter	Este campo es un auto numérico que identifica las notas.
IdPersona	Integer	Este campo es una llave extranjera y almacena el identificador de los estudiantes que tienen notas.
IdEvaluacion	Integer	Este campo es una llave extranjera y almacena el identificador de las evaluaciones que tienen notas.

Tabla 19 Descripción de la tabla Notas.

<b>NotasIntervalo</b>		
En esta tabla se almacenan los tipos de evaluaciones específicas		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id	Integer	Este campo es una llave extranjera y almacena el identificador de las notas.
Valor	Integer	En este campo se almacenan los valores de las notas de Intervalo.
IdFormaCalificacion	Integer	Este campo es una llave extranjera y almacena el identificador de la forma de calificación.

Tabla 20 Descripción de la tabla NotasIntervalo.

<b>NotasConjunto</b>		
En esta tabla se almacenan los tipos de evaluaciones específicas		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id	Integer	Este campo es una llave extranjera y almacena el identificador de las notas.
Valor	Integer	En este campo se almacenan los valores de las notas de conjunto.
IdFormaCalificacion	Integer	Este campo es una llave extranjera y almacena el identificador de la forma de calificación.

Tabla 21 Descripción de la tabla NotasConjunto.

### **Conclusiones**

En el capítulo anterior se presentó el análisis y diseño del sistema, los diagramas que fueron necesarios realizar con la ayuda del Rational Rose para obtener una mejor visión de la solución del problema presentado; además se presentó el diseño de la base de datos en la cual se van a almacenar los datos persistentes resultados de la gestión de la información del sistema de Registro Virtual.

### **Conclusiones**

El desarrollo del sistema de Registro Virtual se logró a partir de la información que fue suministrada por los involucrados en esta tarea, además de una buena selección de las herramientas informáticas a utilizar y del estudio que realizado por parte del equipo de trabajo sobre las necesidades reales de los clientes; contribuyó también la búsqueda de las soluciones existentes en el mundo y en Cuba; esta información fue de gran utilidad pues ilustra cómo se había tratado este tema en el mundo y en Cuba, así se pudo dar solución a este problema de primer orden de la institución y se abrió el camino para que otros investigadores y desarrolladores continúen esta labor, que no acaba aquí, esto es solo el comienzo de una tarea que llevará tiempo desarrollar en su totalidad pero que al final va a reportar resultados satisfactorios para el personal involucrado en esta labor.



### Recomendaciones

- Realizar el análisis y diseño del sistema para los casos de uso que no fueron expandidos para enriquecer aun más esta herramienta de trabajo.
- Para una próxima iteración considerar los cambios que puedan surgir en el negocio y que influyan en el comportamiento del sistema.
- 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 a través de él.
- Elaborar el manual de usuarios del sistema.
- Extender su uso a toda la universidad para lograr un desarrollo homogéneo de todo el proceso académico.

### Referencias bibliográficas

- [1] “Universidad Politécnica de Nicaragua”, UPOLI, en <http://www.upoli.nc>
- [2] “Universidad Politécnica de Madrid”, en <http://www.upm.es>
- [1] “Introduction to Web Forms Pages”, MSDN, Abril, 2003.
- [2] CONALLEN, Jim. “Modeling Web Applications with UML”, en <http://www.conallen.org/whitepapers/webapps/ModelingWebApplications.htm>, marzo, 1999.
- [3] “Bases de Datos” en <http://www.abcdatos.com/tutoriales/tutorial/l6455.html>
- [4] “ABC de Oracle” en <http://www.programatium.com/oracle.htm>
- [5] Quintana Rodríguez, Julio T. “Sistemas distribuidos con componentes”, en <http://www.uv.mx/iiesca/revista4/distribuidos.htm>

### Bibliografía

1. Fowler, Martín. "UML Gota a Gota". Primera Edición. Addison Wesley Longman. 1999.
2. Larman, Craig. *UML y Patrones, Introducción al análisis y diseño orientado a objetos*. Prentice-Hall, 2002.
3. Ceria, Santiago. *Ingeniería de Software I. Casos de Uso. Un Método Práctico para Explorar Requerimientos*.
4. Desarrollo basado en RUP bajo la herramienta Rational Rose  
<http://lml.ls.fi.upm.es/mdp/si/> (4/4/2004)
5. Universidad <http://www.microsoft.com/spanish/msdn/comunidad/uni.net/> .NET  
<http://www.microsoft.com/spanish/msdn/comunidad/uni.net/> (4/4/2004)
6. Curso práctico de desarrollo de aplicaciones con Visual Studio .NET  
<http://www.microsoft.com/spanish/msdn/comunidad/uni.net/> (4/4/2004)
7. Villariño, Luis. *UML para Web*. 2002.
8. Booch, G., Rumbaugh, J., Jacobson, I. *El Lenguaje Unificado de Modelado*. Addison-Wesley. 1999.
9. Booch, G., Rumbaugh, J., Jacobson, I. *The Unified Software Development Process*. Addison-Wesley. 1999.
10. Cockburn, A. *Using Goal-Based Use Cases*. JOOP, 1997
11. García Molina, J. *Towards Use Case and Conceptual Models through Business Modeling*. Conference on Conceptual Modelling. 2000
12. Larman, C. *Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design*. Prentice-Hall, 1998.
13. Gamma, E. *Design Patterns, Elements of Reusable Object-Oriented Software*. Addison-Wesley. 1995.
14. Meyer, B. *Construcción de software orientado a objetos*. Prentice-Hall. 1998.
15. Riel, A. *Object-Oriented Design Heuristics*. Addison-Wesley. 1996.
16. Jim Conallen. *Building Web Applications with UML*. Addison-Wesley. 1999.
17. "Introduction to Web Forms Pages", MSDN, Abril, 2003.

## **BIBLIOGRAFÍA**

---

18. CONALLEN, Jim. "Modeling Web Applications with UML", en <http://www.conallen.org/whitepapers/webapps/ModelingWebApplications.htm>, marzo, 1999.
19. "Bases de Datos" en <http://www.abcdatos.com/tutoriales/tutorial/l6455.html>
20. "ABC de Oracle" en <http://www.programatium.com/oracle.htm> Quintana Rodríguez, Julio T. "Sistemas distribuidos con componentes", en <http://www.uv.mx/iiesca/revista4/distribuidos.htm>
21. "Universidad Politécnica de Nicaragua" en <http://www.upoli.nc>

**Anexos**

## Anexo 1

### 1.1 Modelo del dominio

#### 1.1.1 Modelo del dominio del problema

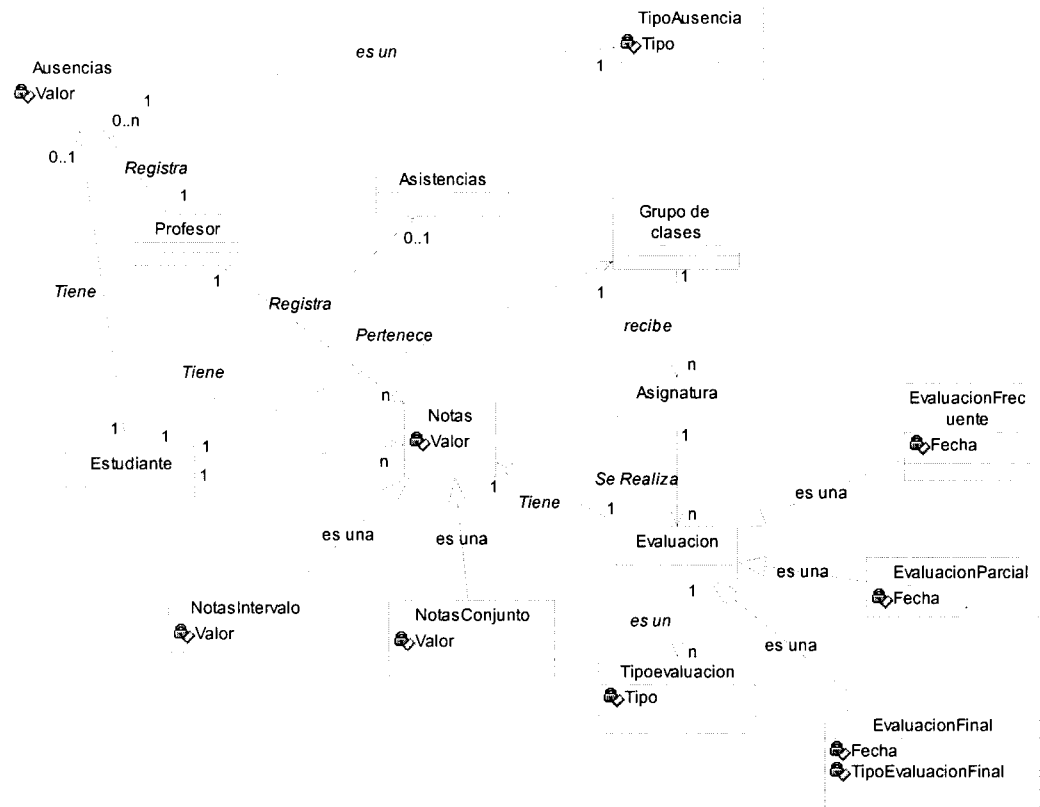


Figura 1 Modelo del dominio del problema.

**1.2 Definición de los casos de uso del sistema****1.2.1 Definición de los actores**

<b>ACTOR</b>	<b>DESCRIPCIÓN</b>
Profesor	Representa a un usuario que pertenezca al rol profesor. Este tipo de usuarios son los únicos usuarios que pueden gestionar la asistencia y las notas de los estudiantes.
Usuario	Representa a cualquier usuario humano del sistema
Estudiantes	Representa a un usuario que pertenece al rol Estudiantes. Este tipo de usuario puede ver el listado de asistencia y notas de los estudiantes en las diferentes asignaturas.
Directivos	Representa a cualquier directivo del centro. Este tipo de usuario puede ver el listado de notas de los estudiantes, además puede configurar el registro del profesor.
Servidor de correo	Este usuario representa al servidor de correo del centro. Es el encargado de entregar las alertas que genera el sistema de registro.
Sistema de expediente académico	Este usuario representa al subsistema de expediente académico. Recibe el resumen final de notas de los estudiantes.
Agente de alertas	Este usuario representa al agente de

	<p>alertas del sistema. Es el encargado de enviar alertas a los profesores y estudiantes.</p>
--	---

Tabla 1 Definición de los actores del sistema.

### 1.2.2 Descripción de de los casos de uso

Nombre del caso de uso	Registrar asistencia
Descripción del caso de uso	Permite registrar la asistencia de los estudiantes pertenecientes a un grupo académico a un turno de clases de una asignatura.
Actores	Profesor
Precondiciones	<p>El actor debe haberse logueado en el sistema</p> <p>El sistema debe mostrar el listado de estudiantes pertenecientes al grupo académico de clases que le corresponde docencia en ese momento.</p>
Flujo normal del caso de uso	<ol style="list-style-type: none"> <li>1. El sistema muestra el listado de estudiantes que le corresponde docencia en ese momento.</li> <li>2. El profesor registra los estudiantes que están ausentes a su turno de clases.</li> <li>3. El profesor argumenta sobre la ausencia de estudiantes a clases.</li> </ol>



## ANEXOS

	Los estudiantes con problemas de asistencia se sobresaltan con otro color en el listado del grupo.
Flujo alternativo del caso de uso	1. El sistema comprueba la validez de los datos, si los datos no son válidos se le avisa al actor permitiendo que los corrija.
Poscondiciones	Los datos de la asistencia de estudiantes se almacenan en el sistema.

Tabla 2 Descripción del caso de uso Registrar Asistencia.

<b>Nombre del caso de uso</b>	<b>Registrar notas</b>
Descripción del caso de uso	Permite registrar las notas de los estudiantes en las evaluaciones de una asignatura.
Actores	Profesor
Precondiciones	<p>El profesor debe haberse logueado en el sistema</p> <p>El profesor debe haber registrado la asistencia de los estudiantes a su turno de clases.</p> <p>El sistema debe mostrar la lista de estudiantes pertenecientes al grupo académico de clases que le corresponde docencia en ese momento y que no estén ausentes a ese turno de clases.</p>
Flujo normal del caso de uso	1. El sistema muestra el listado de

	<p>estudiantes que le corresponde docencia en ese momento y que no están ausentes a ese turno de clases.</p> <p>El sistema muestra un campo de texto para introducir las notas de los estudiantes.</p> <p>Los estudiantes con problemas académicos se sobresaltan con otro color en el listado del grupo.</p>
Flujo alternativo del caso de uso	El sistema comprueba la valides de los datos, si los datos no son validos se le avisa al actor permitiendo que los corrija.
Poscondiciones	Los datos de las notas de estudiantes se almacenan en el sistema y almacena el nombre del profesor y la dirección IP de la maquina de donde tomo la asistencia.

Tabla 2 Descripción del caso de uso registrar notas.

<b>Nombre del caso de uso</b>	<b>Mostrar estado de estudiantes</b>
Descripción del caso de uso	Permite a los actores solicitar los datos de asistencia y académicos de los estudiantes en evaluaciones de una asignatura de un grupo académico.
Actores	Profesor, Alumno, Directivos
Precondiciones	<p>Los actores deben haberse logueado en el sistema.</p> <p>La información sobre la asistencia y las</p>

	<p>evaluaciones de los estudiantes debe encontrarse en el sistema.</p> <p>El actor debe solicitar al sistema que le muestre un listado con los datos de asistencia y de los resultados de los estudiantes de un grupo académico en una asignatura.</p>
Flujo normal del caso de uso	<p>1. En caso que el actor seleccione mostrar listado de asistencia de estudiantes, el sistema muestra un listado de estudiantes con el estado de la asistencia a un turno de clases de una asignatura.</p> <p>En caso que el actor seleccione resultados de los estudiantes, el sistema muestra un listado con el estado de los resultados académicos de los estudiantes en una asignatura.</p>
Flujo alternativo del caso de uso	Si la información del estado de estudiantes no se encuentra en el sistema se devuelve un mensaje de error.
Poscondiciones	El actor tiene acceso a la información de los datos académicos y de asistencia de los estudiantes en una asignatura.

Tabla 4 Descripción del caso de uso mostrar estado de estudiantes.

<b>Nombre del caso de uso</b>	<b>Mostrar estudiantes con problemas</b>
Descripción del caso de uso	Permite solicitar los datos académicos y los de asistencia de los estudiantes de un grupo académico que presentan problemas de asistencia y docentes.
Actores	Profesor, Alumno, Directivos
Precondiciones	<p>Los actores deben haberse logueado en el sistema.</p> <p>La información sobre las notas en las evaluaciones de la asignatura de los estudiantes así como la información de la asistencia a clases de los estudiantes en esa asignatura deben haber sido entradas al sistema.</p> <p>El actor debe solicitar al sistema que le muestre un listado con los datos de los estudiantes con problemas de asistencia y docentes en una asignatura.</p>
Flujo normal del caso de uso	<ol style="list-style-type: none"><li data-bbox="748 1422 1360 1794">1. En caso que el actor seleccione mostrar los estudiantes que presentan problemas de asistencia en una asignatura, el sistema muestra el listado de estudiantes que presentan problemas de asistencia en una asignatura.</li><li data-bbox="748 1806 1360 1857">2. En caso que el actor seleccione</li></ol>

	mostrar listado de estudiantes con problemas docentes en una asignatura, el sistema muestra un listado de los estudiantes que presentan problemas docentes en una asignatura.
Flujo alternativo del caso de uso	
Poscondiciones	El actor tiene acceso a la información sobre los estudiantes que presentan problemas de asistencia y docentes en una asignatura.

Tabla 5 Descripción del caso de uso mostrar estudiantes con problemas.

<b>Nombre del caso de uso</b>	<b>Enviar alertas</b>
Descripción del caso de uso	<p>Permite al sistema enviar alertas a estudiantes y profesores por:</p> <ul style="list-style-type: none"> <li>- Necesidad de realizar evaluación(profesor)</li> <li>- Proximidad de evaluación (estudiante)</li> <li>- Problemas de asistencia (estudiante)</li> <li>- Problemas docentes (Estudiantes).</li> </ul>

## ANEXOS

---

Actores	Alumno, Servidor de correo, Profesor
Precondiciones	<p>Los actores deben haberse logueado en el sistema.</p> <p>La información sobre las notas y las evaluaciones de la asignatura de los estudiantes así como la información de la asistencia a clases de los estudiantes en esa asignatura deben haber sido entradas al sistema.</p> <p>Se debe contar con los datos de los estudiantes que presentan problemas de asistencia a clases así como los que presentan problemas docentes, en una asignatura.</p> <p>Se debe contar con los datos de la cantidad de evaluaciones que debe realizar un profesor en un periodo de tiempo determinado.</p>
Flujo normal del caso de uso	<ol style="list-style-type: none"><li>1. Los mensajes por problemas de asistencia se generan automáticamente a partir del listado de estudiantes que presentan problemas de asistencia en una asignatura.</li><li>2. Los mensajes por problemas docentes de los estudiantes se</li></ol>

	<p>generan automáticamente a partir del reporte de estudiantes con problemas docentes, pertenecientes a un grupo académico.</p> <p>3. Los mensajes por proximidad de evaluaciones se generan a partir de la información que tiene el sistema cantidad de evaluaciones que debe realizar un profesor a un grupo académico en una asignatura.</p>
Flujo alternativo del caso de uso	En caso de no encontrarse la información de los estudiantes con problemas de asistencia y académicos, se enviara un mensaje de error.
Poscondiciones	El actor recibe un mensaje en cual se le informa de problemas docentes, de asistencia y de proximidad de evaluación en una asignatura.

Tabla 6 Descripción del caso de uso Enviar alertas.

Nombre del caso de uso	Configurar Registro
Descripción del caso de uso	<p>Permite configurar lo siguiente en el registro:</p> <ul style="list-style-type: none"> <li>- Alertas que se le van a enviar al los profesores y estudiantes.</li> <li>- % de asistencias para tener derecho a la prueba final de una asignatura.</li> </ul>

**ANEXOS**

---

	Nota mínima para el aprobado en una asignatura.
Actores	Profesor, servidor de correo
Precondiciones	Los usuarios deben haberse logueado en el sistema.
Flujo normal del caso de uso	<ol style="list-style-type: none"><li>1. El sistema muestra un conjunto de propiedades que pueden ser configuradas por el usuario.</li><li>2. El usuario escoge la propiedad o las propiedades configurar.</li><li>3. El sistema guarda la nueva configuración del registro</li></ol>
Flujo alternativo del caso de uso	Las propiedades a configurar tienen que estar definidas en el sistema
Poscondiciones	El registro académico se reconfigura con la nueva configuración.

Tabla 5 Descripción del caso de uso configurar registro.



Nombre del caso de uso	Generar Reporte Final de Notas
Descripción del caso de uso	Permite generar el resumen final de notas de los estudiantes.
Actores	Secretaria, Sistema Expediente Académico
Precondiciones	<p>Le Secretaria debe haberse logueado en el sistema.</p> <p>La información sobre las notas finales de los estudiantes debe encontrarse en el sistema.</p> <p>La secretaria debe solicitar generar resumen final de notas.</p>
Flujo normal del caso de uso	<ol style="list-style-type: none"> <li>1. El sistema muestra las notas de los estudiantes en las asignaturas.</li> <li>2. La secretaria genera el resumen final de notas de los estudiantes que contiene la nota de los estudiantes en las asignaturas.</li> </ol>
Flujo alternativo del caso de uso	Si los datos de las notas no se encuentran en el sistema, se envía un mensaje de error.
Poscondiciones	El subsistema de expediente académico recibe un documento con las notas de los estudiantes en las asignaturas.

Tabla 7 Descripción del caso de uso generar reporte final de notas.

### 1.2.3 Diagrama de casos de uso del sistema

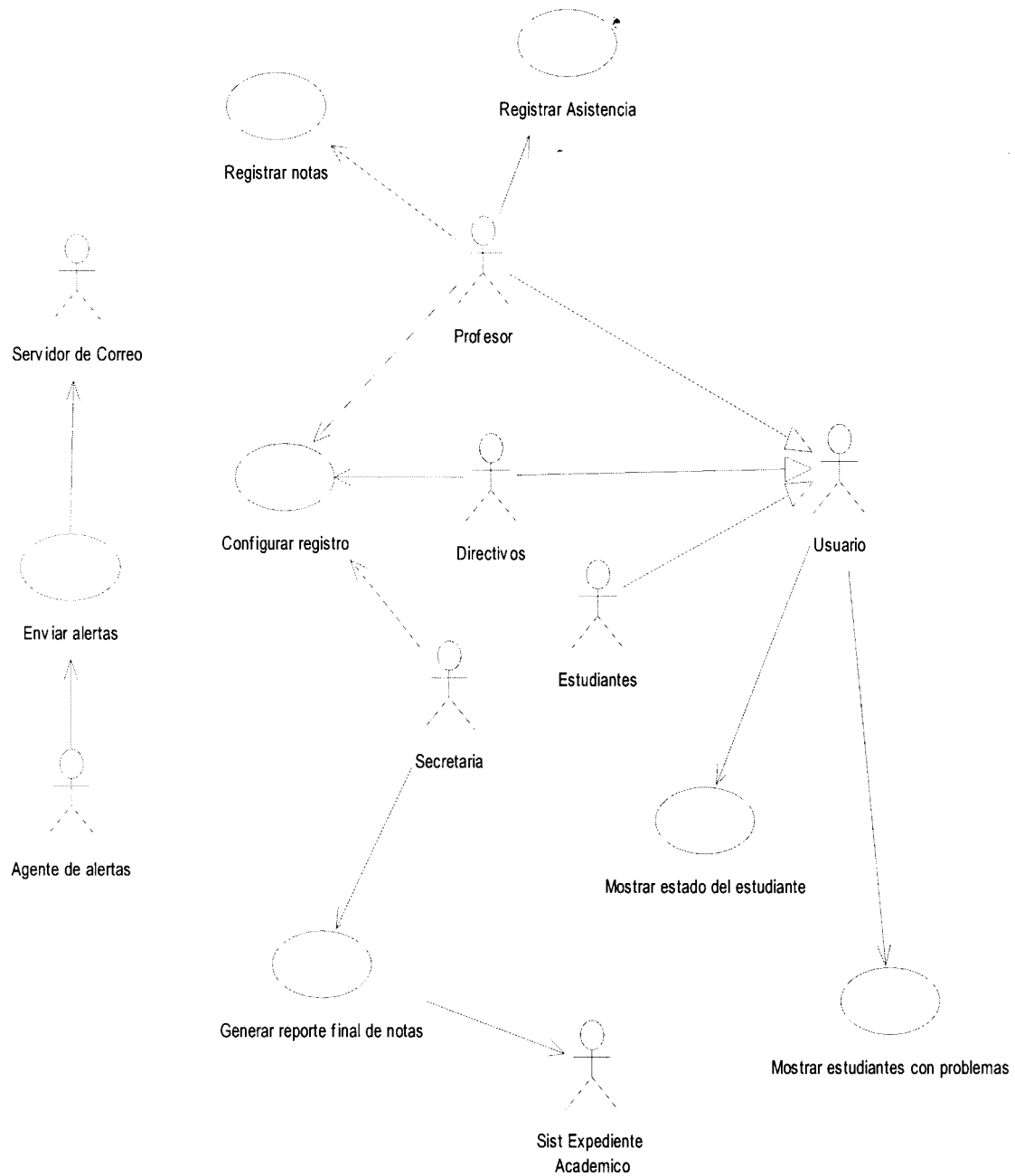


Figura 2 Diagrama de casos de uso del sistema.

## 1.2.4 Casos de uso expandidos

<b>Caso de Uso (1)</b>	<b>Registrar asistencia</b>	
<b>Actores</b>	Profesor	
<b>Propósito</b>	Registrar la asistencia de los estudiantes a una asignatura en un turno de clases.	
<b>Resumen</b>		
El caso de uso inicia una vez que el profesor se ha autenticado en el sistema y solicita registrar la asistencia de los estudiantes a su turno de clases.		
<b>Referencias</b>	R1	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
Página "Registrar Asistencia"		
1. El caso de uso comienza cuando el profesor pide registrar la asistencia de los estudiantes a su turno de clases.	2. El sistema muestra la página "Registrar asistencia", con un listado de los grupos a los que el profesor les imparte clases.	
3. El profesor selecciona el grupo en el que desea tomar la asistencia.	4. El sistema muestra un listado de los estudiantes pertenecientes al grupo de clases que el profesor seleccionó, con las opciones de:  Marcar inasistencia, seleccionar tipo de inasistencia y  Aceptar(A), Cancelar(B)	
5. El profesor toma la asistencia, selecciona el tipo de inasistencia y escoge una de las opciones disponibles	6. Si el profesor selecciona.  A) El sistema registra los datos de inasistencia de estudiantes y del tipo de	

	<p>inasistencia.</p> <p>B) El sistema muestra la página "Registrar Asistencia", con los controles limpios.</p>
Sección "Modificar datos de asistencia"	
Página "Registrar asistencia"	
7. El caso de uso comienza cuando el profesor solicita modificar la asistencia de los estudiantes.	8. El sistema muestra la página "Registrar asistencia" que contiene el listado de los grupos a los que el profesor les imparte clases y la fecha donde desea modificar la asistencia.
9. El profesor selecciona la fecha y el grupo al que le desea modificar la asistencia.	<p>10. El sistema muestra el listado de estudiantes de ese grupo académico con los valores de inasistencia y en caso de tener inasistencia, el tipo.</p> <p>Con las opciones de:</p> <p>(A)Modificar, (B)Cancelar</p>
10. El profesor modifica los valores de inasistencia de los estudiantes en ese grupo académico.	<p>11. Si el profesor escoge:</p> <p>(A) El sistema modifica los valores de inasistencia de los estudiantes en ese grupo académico.</p> <p>(B) El sistema muestra la página "Modificar Asistencia"</p>

<p><b>Precondición</b></p> <p>El Profesor se ha autenticado en el sistema.</p>
<p><b>Poscondición</b></p> <p>Se ha actualizado los datos de asistencia de los estudiantes.</p>
<p><b>Caminos Alternativos</b></p>

Tabla 8 Expansión del caso de uso registrar asistencia.

<b>Caso de Uso (2)</b>	<b>Registrar Notas</b>	
<b>Actores</b>	Profesor	
<b>Propósito</b>	Registrar las notas de los estudiantes pertenecientes a un grupo académico en una asignatura.	
<b>Resumen</b>		
El caso de uso inicia una vez que el profesor se ha autenticado en el sistema y solicita registrar las notas en una evaluación de los estudiantes pertenecientes a un grupo en una asignatura.		
<b>Referencias</b>	R2	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
Página "Registrar Notas"		
1. El caso de uso comienza cuando el profesor pide registrar las notas de una evaluación.	2. El sistema muestra la página "Registrar Notas" con un listado de los grupos a los que le imparte clases el profesor.	
3. El profesor selecciona el grupo al que desea registrar las notas.	4. El sistema muestra el listado de estudiantes de ese grupo y la opción de seleccionar tipo de evaluación a	

	<p>realizar.</p> <p>Los tipos de evaluaciones son: Evaluaciones Frecuentes(A), Evaluaciones Parciales(B) y Evaluaciones Finales(C)</p> <p>y las opciones de: Aceptar(D), Cancelar(E)</p>
<p>5. El profesor selecciona tipo de evaluación a realizar.</p>	<p>6. Si selecciona:</p> <p>A) El sistema crea una evaluación frecuente para ese grupo y muestra para cada uno de los estudiantes los valores posibles que puede tomar una evaluación frecuente para esta asignatura.</p> <p>B) El sistema muestra la opción de seleccionar la evaluación parcial a realizar.</p> <p>C) El sistema muestra la opción de seleccionar el tipo de evaluación final a realizar.</p>
<p>7. Si (B) El profesor selecciona la evaluación parcial a realizar.</p> <p>Si (C) El profesor selecciona la evaluación final a realizar.</p>	<p>8. Si (B)</p> <p>El sistema verifica que no haya notas registradas para esta evaluación en dicho grupo.</p> <p>Si no existen notas el sistema crea</p>

	<p>esta evaluación parcial para el grupo y muestra para cada uno de los estudiantes los posibles valores que puede tomar esta evaluación parcial.</p> <p>Si (C)</p> <p>Si la evaluación final es “Examen Final” o “Examen Extraordinario” el sistema verifica que la asignatura tenga examen final.</p> <p>El sistema verifica que no haya notas registradas para la evaluación final seleccionada.</p> <p>Si no existen notas el sistema crea esta evaluación final para el grupo y muestra para cada uno de los estudiantes los posibles valores que puede tomar una evaluación final en dicha asignatura.</p>
<p>9. El profesor selecciona las notas de los estudiantes y selecciona una de las opciones disponibles.</p>	<p>10. Si el profesor selecciona:</p> <p>D) El sistema guarda las notas de los estudiantes que pertenecen a ese grupo en esa evaluación.</p> <p>E) El sistema muestra la página “Registrar Notas” con los controles en blanco.</p>
<p>Sección “Modificar notas”</p>	

<p>Página "Modificar Notas"</p>	
<p>11. El caso de uso comienza cuando el profesor solicita Modificar notas de estudiantes en una evaluación de su asignatura.</p>	<p>12. El sistema muestra la página "Modificar Notas" que contiene:  El listado de los grupos a los que el profesor imparte clases.</p>
<p>13. El profesor escoge el grupo al que le desea modificar las notas y la fecha en que se realizó la evaluación.</p>	<p>14. El sistema muestra el listado de estudiantes pertenecientes al grupo de clases que el profesor seleccionó y la opción de seleccionar tipo de evaluación a realizar:  (A) Modificar, (B) Cancelar</p>
<p>15. El profesor selecciona tipo de evaluación.</p>	<p>16. Si selecciona:  A) El sistema muestra para cada uno de los estudiantes del grupo la nota de la evaluación frecuente que se realizó para ese grupo en esa fecha para esta asignatura.  B) El sistema muestra la opción de seleccionar la evaluación parcial a realizar.  C) El sistema muestra la opción de seleccionar la evaluación final a realizar.</p>



<p>17. Si (B) El profesor selecciona la evaluación parcial a realizar.</p> <p>Si (C) El profesor selecciona la evaluación final a realizar.</p>	<p>18. El sistema muestra para cada uno de los estudiantes del grupo la nota de la evaluación seleccionada que se realizó para ese grupo en esa fecha para esta asignatura.</p>
<p>19. El profesor modifica los resultados de los estudiantes y selecciona una de las opciones disponibles.</p>	<p>20. Si el profesor selecciona:</p> <p>A) El sistema actualiza los datos de las evaluaciones de los estudiantes.</p> <p>B) El sistema muestra la página “Modificar Notas” con los resultados anteriores de los estudiantes en esa evaluación.</p>
<p><b>Precondición</b></p> <p>El Profesor se ha autenticado en el sistema</p>	
<p><b>Poscondición</b></p> <p>Se ha actualizado las evaluaciones realizadas a los estudiantes y las notas obtenidas en las mismas.</p>	
<p><b>Caminos Alternativos</b></p> <ol style="list-style-type: none"> <li>1. Si en el inciso B) del paso 8 existen notas registradas para este grupo de la evaluación parcial seleccionada se muestra un mensaje de error.</li> <li>2. Si en el inciso C) del paso 8 existen notas registradas para este grupo de la evaluación final seleccionada se muestra un mensaje de error.</li> <li>3. Si en el inciso C) del paso 8 en caso de haberse escogido “Examen Final”</li> </ol>	

<p>o “Examen Extraordinario” si la asignatura no tiene examen final se muestra un mensaje de error.</p> <p>4. Si en el inciso A) del paso 16 no existe ninguna evaluación frecuente realizada para ese grupo en esa fecha para esta asignatura se muestra un mensaje de error.</p> <p>5. Si en el paso 18 la evaluación seleccionada no existe para ese grupo en esa fecha para esta asignatura se muestra un mensaje de error.</p>
---

Tabla 9 Expansión del caso de uso registrar notas.

<b>Caso de Uso (3)</b>	<b>Mostrar estado de los estudiantes.</b>	
<b>Actores</b>	Profesor, estudiantes, directivos	
<b>Propósito</b>	Mostrar el estado de los estudiantes pertenecientes a un grupo en una asignatura, en un momento del curso escolar.	
<b>Resumen</b>		
El caso de uso inicia una vez que el autor se ha logueado en el sistema y solicita ver el estado de los estudiantes pertenecientes a un grupo de clases en una asignatura.		
<b>Referencias</b>	R3	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
<b>Página “Mostrar estado estudiantes”</b>		
1. El caso de uso comienza cuando el actor solicita ver el estado de los estudiantes pertenecientes a un grupo de clases.	2. El sistema muestra la opción de seleccionar la facultad, el grupo de clases y la asignatura y si desea ver el estado de asistencia (A) o docente (B)	

	de los estudiantes.
3. El actor selecciona la facultad.	4. El sistema muestra todos los grupos de la facultad seleccionada.
4. El actor selecciona el grupo.	5. El sistema muestra todas las asignaturas en las que están matriculados los estudiantes de ese grupo.
6. El actor selecciona la asignatura	7. El sistema muestra los estudiantes de ese grupo que están matriculados para esa asignatura.
8. El actor selecciona las opciones disponibles.	9. Si el actor selecciona:  A) El sistema muestra el estado de asistencia de cada uno de los estudiantes para esa asignatura.  B) El sistema muestra el estado académico de cada uno de los estudiantes para esa asignatura.
<b>Precondición</b>	
El Actor se ha autenticado en el sistema	
<b>Poscondición</b>	
Se muestra el estado de asistencia o docente de los estudiantes de un grupo en una asignatura.	
<b>Caminos Alternativos</b>	
1. Si en el paso 9, no hay resultados de asistencia o docentes, se le muestra un aviso al actor.	

Tabla 10 Expansión del caso de uso mostrar estado del estudiante.

### Glosario de términos

**Akademios:** Sistema Automatizado para la Gestión Académica.

**APIs:** Applications Programming Interfaces. Conjunto de subrutinas que aportan los sistemas operativos a los programas de aplicación para el acceso a los recursos y servicios prestados por el ordenador.

**COM:** Modelo de componentes de Microsoft. Permite interconectar componentes software y aplicaciones de una manera "simple", segura y homogénea.

**COM+:** Nueva versión de COM que Microsoft incluye en .NET y que pretende solucionar alguno de los problemas que presenta el antiguo COM.

**Compilador JIT (Just-In-Time):** Herramienta que genera el código de máquina real que se ejecuta en la plataforma que tenga la computadora.

**Db:** Data Base o Base de Datos.

**Gestión académica:** La gestión académica involucra el planeamiento y la programación curricular, la producción de materiales didácticos, la ejecución, la evaluación, articulación de los aprendizajes y la enseñanza y aprendizaje enmarcados en un concepto de exigencia, excelencia académica y mejora continua.

**Comprende los siguientes procesos:**

- a) Investigaciones y evaluaciones que orienten la oferta académica y su viabilidad.
- b) Programación curricular.
- c) Producción de los materiales didácticos
- d) Ejecución del proceso educativo.
- e) Evaluación de programas, carreras, cursos, materiales didácticos y la gestión académica.

## **GLOSARIO DE TÉRMINOS**

---

**J2EE:** Java 2 Enterprise Edition. Estándar para el desarrollo de aplicaciones empresariales multicapas. Simplifica las aplicaciones empresariales basándolas en componentes modulares y estandarizados, proveyendo un completo conjunto de servicios a estos componentes, y manejando muchos de las funciones de la aplicación de forma automática, sin necesidad de una programación compleja”.

**Registro Académico:** Documento oficial que se le entrega al personal docente al inicio de un curso lectivo y que permite el almacenamiento de los datos de asistencia a clases y de los resultados en las diferentes evaluaciones que realizan los estudiantes.

**SOAP:** Simple Access Object Protocol.

**SQL:** Structured Query Language. Lenguaje estándar de comunicación con bases de datos.

**UNIX:** Sistema operativo portable, flexible, potente, con entorno programable, multiusuario y multitarea, muy difundido.

**XML:** Extensible Markup Language (Lenguaje extensible de etiquetas) Es un meta-lenguaje que permite definir lenguajes de marcado adecuado a usos determinados. Se propone como lenguaje de bajo nivel (a nivel de aplicación, no de programación) para intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo, y casi cualquier cosa que se pueda pensar.