

003.7
Cuz
S
TD 0022-04-01

TD-0022-04-01

Universidad de Oriente



Trabajo de Diploma

Título

*Sistema automatizado para la Gestión Académica.
Subsistema "Expediente virtual"*



Autor

Virgen Cuza Noa

Tutor

Ing. Emil Lima Valdés

Curso 2003-2004

Resumen

Para la Universidad de las Ciencias Informáticas es de vital importancia contar con una herramienta que permita la gestión del proceso docente. Hasta el momento se han utilizado algunos sistemas aislados, pero estos no cumplen con las necesidades de la institución, por lo que se ha hecho imposible la total automatización de los procesos docentes. En este sentido se desarrolla el sistema de gestión académica AKADEMOS, un sistema informático que permite la automatización de los procesos de gestión académica requeridos por la universidad, el mismo está dividido en módulos, el presente trabajo permite describir el desarrollo del módulo Expediente académico.

Mediante este se podrá mantener una versión digital de los expedientes de los estudiantes de un centro de estudios superiores, y garantizará la captación de la información asociada a estos directamente desde donde se produce.

Para materializar este propósito central, se siguieron los pasos que proponen el Proceso Unificado. Está implementado sobre la plataforma Microsoft .NET, específicamente en el lenguaje C#, como gestor de base de datos se utilizó SQL Server. El presente documento contiene los resultados obtenidos de los flujos de trabajo de Requisitos, Análisis y Diseño del módulo Expediente académico.

Índice

Introducción	1
Capítulo 1. Fundamentación Teórica.....	4
Introducción	4
Estado del Arte.....	4
Ámbito Internacional.	4
Estudios en el país.....	7
Herramientas utilizadas para el desarrollo del sistema.....	8
La tecnología .NET.....	9
Arquitectura Framework.NET	9
Lenguajes de programación.....	9
Modelación de aplicaciones Web sobre la plataforma ASP.NET	14
Programación Multicapas.....	15
Servicios Web (Web Services).....	17
Protocolos que usan los Servicios Web.....	17
Sistemas Gestores de Base de Datos (SGDB).....	18
RUP.....	25
Rational Rose.....	27
Conclusiones.....	28
Capítulo 2. Características del Sistema.....	29
Introducción.....	29
Objeto de estudio	29
Problema:	29
Objeto de automatización.....	29
Información que se maneja.....	30
Propuesta de sistema.....	31
Especificación de los requisitos del sistema.....	31
Dependencias y relaciones.....	31
Requerimientos Funcionales.....	32
Requerimientos no funcionales.....	34
Casos de uso del sistema.....	36
Definición de los casos de usos.....	37
Casos de uso por ciclo.....	39
Diagrama de casos de uso	40
Expansión de los casos de uso.....	42
Conclusiones.....	42
Capítulo 3. Análisis y diseño del Sistema.....	43
Introducción	43
Modelo de clases de análisis.....	44
Diagramas de secuencia del sistema.....	45
Diagramas de clases del diseño.....	45
Descripción de las clases.....	45
Diseño de la Base de Datos (BD).....	45
Descripción de las tablas.....	46
Tratamiento de errores	46
Seguridad	47
Conclusiones:.....	47
Conclusiones.....	48

Recomendaciones.....	49
Referencias Bibliográficas.....	50
Bibliografía.....	51
Glosario de Términos.....	53
Anexos.....	57

Introducción

La universidad es la generadora del potencial humano que se necesita para la transformación y desarrollo de la sociedad. En los momentos actuales se ha tomado conciencia acerca de la importancia que tiene este tipo de educación para el desarrollo de la humanidad en las diferentes esferas de la vida, por tanto, se ha desprendido una gran demanda de la matrícula en la educación superior.

Se espera que la Universidad de las Ciencias Informáticas (UCI) en pocos años sea el centro que cuente con la mayor cantidad de estudiantes con respecto a las demás universidades del país, esto da cabida a que en la misma aparezcan diariamente necesidades asociadas al proceso de gestión académica.

La gestión académica es una de las piedras angulares de la organización universitaria, tiene una importancia significativa en el entramado administrativo, que afecta no sólo a los alumnos que entendemos que son los clientes hacia quien se presta el servicio de la educación superior, sino también a los otros dos estamentos básicos de la universidad, como son los profesores, que han de prestar una enseñanza de calidad, y el personal de administración y servicios que debe administrarla.

La gestión académica está compuesta por varios procesos, entre los cuales encontramos la gestión del expediente académico, el mismo es el conjunto de documentos de carácter oficial y confidencial de un estudiante, estos documentos contienen una amplia gama de información, donde pueden estar incluidas:

- La fecha y el lugar de nacimiento, la dirección de los padres
- Las notas, calificaciones de exámenes, materias estudiadas, especialización y actividades académicas, cartas oficiales respecto a datos del estudiante en la escuela.
- Constancias de educación especial.
- Informes disciplinarios.

- Registros médicos o de salud, creados o recopilados y mantenidos por la escuela.
- Documentación de las escuelas a las cuales ha asistido el estudiante, premios conferidos y títulos recibidos.

Actualmente no se cuenta con una aplicación para el proceso de gestión de la información del expediente, de ahí que el problema que pretende resolver el presente trabajo sea:

¿Cómo lograr almacenar digitalmente toda la información asociada al paso del estudiante por la universidad?

Para dar solución a dicho problema se toma como objetivo general:

Desarrollar un sistema automatizado que permita mantener una versión digital de los expedientes de los estudiantes de un centro de estudios superiores y que garantice la captación de la información asociada a estos directamente desde donde se produce.

Como objetivos específicos se plantean los siguientes:

- Permitir al usuario el uso de plantillas para su posterior uso en los expedientes.
- Brindar facilidades para llenar los formularios que formarán parte del expediente, los cuales están determinados por las plantillas creadas.

Para cumplir con los objetivos propuestos se prevé que el sistema tenga las funcionalidades siguientes:

- Gestionar las plantillas por las que serán creados los documentos.
- Gestionar los documentos del expediente.
- Prestar servicios a los restantes subsistemas que de una forma u otra necesitan interactuar con el subsistema Expediente.
- Velar por la seguridad de los documentos y las plantillas, dándole al usuario sus respectivos permisos.

En el Capítulo 1 de este documento se hace una breve referencia a los sistemas existentes en el mundo y a nivel nacional del proceso de gestión académica. Además de realizarse un estudio de las tendencias y tecnologías actuales sobre las que se apoya la propuesta del sistema. El Capítulo 2 hace una descripción general del objeto de estudio, de los procesos que serán objeto de automatización, así como un análisis de otras soluciones existentes con la propuesta. Los requisitos funcionales y no funcionales y los casos de uso del sistema a desarrollar. En el Capítulo 3 se hace el análisis y el diseño del sistema donde se define el modelo de análisis, se realizan los diagramas de secuencia, se describen las clases y se hace el diseño de la base de datos.

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 y nacional, 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 varios 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.

Estado del Arte.

La importancia que han adquirido en la sociedad los procesos docente-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: Aplicación de Gestión de Organización Académica.

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 se permite consultar el historial académico del estudiante en un

determinado plan de estudio, modificar o completar sus datos personales y se 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 automatrícula, 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 matricula de los nuevos ingresos, dar baja, rematricular un estudiante que ha sido baja, sin permitir ratificar la matricula, 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 recrudecen 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 de nuestro 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.

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.

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.

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.

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.

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.
- **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.

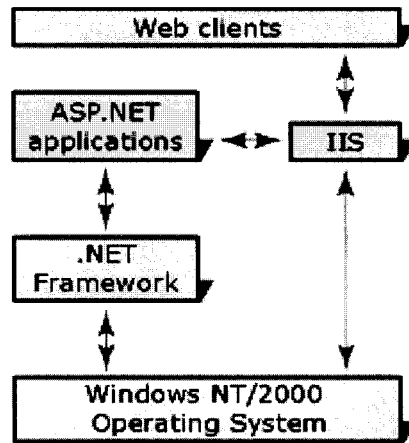


Figura 1 Arquitectura ASP.NET

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 [IWFP]:

- 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.

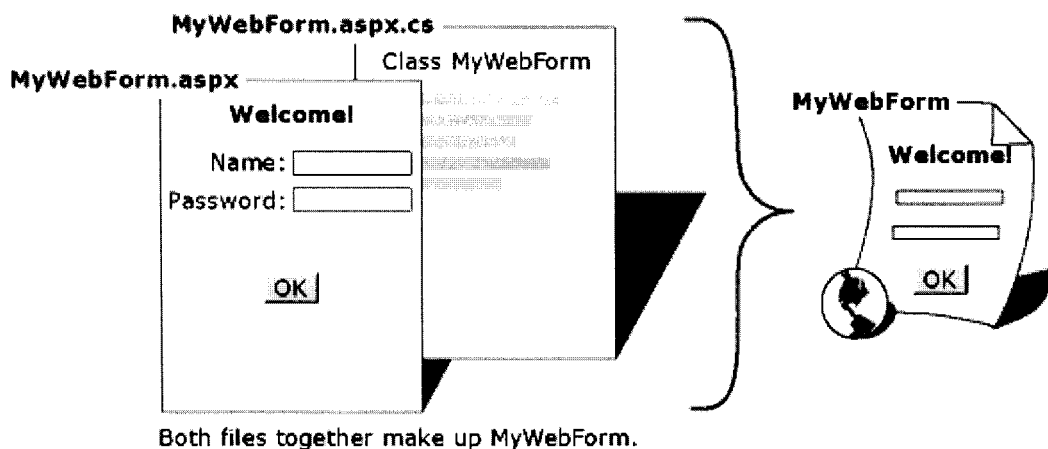


Figura 2 WebForms

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 paginas 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, eliminando 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.

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 [CON]. 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 extensiones 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 la 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.

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.

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.

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.

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.

Sistemas Gestores de Base de Datos (SGDB).

Un Sistema de Gestión de Bases de Datos (SGBD) no es más que 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 [BAS].

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 podemos 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, podemos 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 [ABC].

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:

- 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.

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.

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.

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.

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 podemos 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 cascada 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.

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.

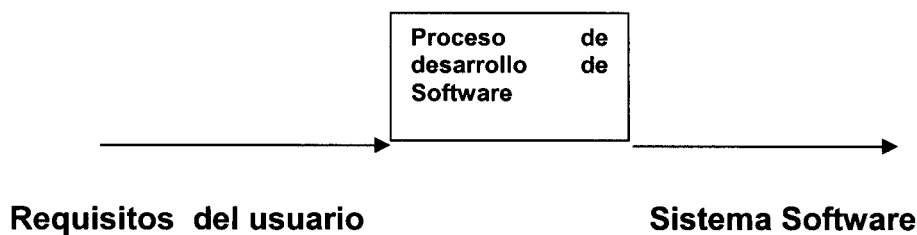


Figura 1. Un proceso de desarrollo de software.

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-proyecto 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.

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 en nuestro caso el SQL Server 2000 es el que más posibilidades nos brinda como desarrolladores por sus características propias que lo hacen ideal para sistemas como el que nos proponemos 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 [QRJ], de ahí que utilizemos estas características para el desarrollo de nuestra 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 siguiente capítulo se define el objeto de estudio, objeto de automatización, información que se maneja, sistema propuesto, especificación de los requerimientos del sistema y se obtienen los casos de uso.

Objeto de estudio

Problema:

En la UCI parte de los procesos de la gestión académica se realizan mediante una aplicación nombrada GestAcad, esta cuenta con la deficiencia que no satisface todas las necesidades de la universidad, debido a que los servicios que brinda son limitados, y en el caso específico que nos ocupa, o sea, el trabajo con los expedientes de forma digital, sus servicios son casi nulos, a lo sumo, da la posibilidad de imprimir algunos reportes para guardarlos posteriormente de esa forma en el expediente, lo cual trae consigo trabajo de oficina adicional y gastos en consumo de papel. Razones que condujeron a la dirección de la UCI a solicitar una subsistema encargado de digitalizar expedientes docentes y que permita a la vez lograr una mayor eficiencia en los trámites y procedimientos administrativos, minimizar la utilización de documentos basados en papel, logrando así una mayor velocidad y simplificación en la gestión administrativa, mayor control del flujo de información que se maneja, que redundará en el beneficio de la administración y de los administrados.

Objeto de automatización.

Teniendo presente las carencias con que cuenta la universidad para la gestión del expediente académico, y la necesidad de agilizar este proceso, deseamos dar solución a la misma, automatizando los diferentes procesos que darán cumplimiento al buen funcionamiento de la gestión del expediente.

Un expediente se crea cuando un estudiante es matriculado, el proceso de matriculación lo realiza el Subsistema de Matricula, al cual se le debe brindar un

servicio para registrar el expediente en estado de prematrícula y posteriormente completar la matrícula cuando lo desee.

Los expedientes deben ser actualizados mediante la incorporación, eliminación o modificación de documentos una vez que el usuario autorizado, el subsistema de registro o el subsistema de matrícula lo exija y haya completado el formulario correspondiente de la forma que le corresponda a cada uno de ellos.

En el expediente estarán documentos que puede ir ser llenados o no a través de una plantilla, cada documento llevará consigo además un historial que permitirá saber quien lo creó y quienes lo han modificado, de modo que en caso de errores o ilegalidades se podrá rastrear quién que lo cometió. El cierre de un expediente es decisión del subsistema de matrícula.

Podrán reproducirse sobre papel los expedientes electrónicos cuando sea necesario, ya sea dentro o fuera de la repartición administrativa de que se trate, o para proceder a su archivo sobre papel. El expediente electrónico tendrá la misma validez jurídica y probatoria que el expediente tradicional.

Información que se maneja.

El contenido del expediente puede ser muy amplio y diverso; sin embargo, es indispensable que contenga los siguientes datos:

Datos de identificación del educando: Nombre completo, fecha y lugar de nacimiento, domicilio, nombre de los padres.

Estos datos estarán contenidos en el documento hoja de matrícula, adicionalmente de otra información que pueda tener asociado el mismo.

El expediente maneja otros documentos como son:

Título correspondiente al nivel medio superior de la educación, certificación de estudios terminados en el nivel medio superior, en caso del expediente pertenecer a un varón debe llevar la certificación de licenciamiento del servicio militar, modelo de tramitación de baja, modelo de notificación de ratificación de la baja, resoluciones, etc.

La información que se maneja digitalmente comprende datos e imágenes relacionadas con los expedientes. Esta debe ser altamente confiable. El volumen de información es considerablemente grande, debido a que abarca numerosos datos de una cantidad indeterminada de personas.

Propuesta de sistema.

Se propone un sistema que realice todas las funcionalidades y logre todos los objetivos propuestos anteriormente en el objeto de automatización. Su funcionamiento está descrito de la siguiente forma:

El subsistema de matrícula decide hacer una prematrícula, este creará un expediente en estado de prematrícula para el estudiante. Dicho expediente puede ser posteriormente abierto dada una orden del mismo subsistema y de este modo el expediente pasara a estado abierto. Una vez que el expediente se encuentra en este estado queda presto a actualizaciones que solo serán realizadas por usuarios autorizados y de la forma antes explicada. El expediente puede ser cerrado solamente por el subsistema de matricula, luego que el expediente es cerrado no se destruye, pues es de interés mantener físicamente el expediente para futuras consultas.

A diferencia de los sistemas expuesto en el capítulo 1, nuestra propuesta le brinda la posibilidad al usuario de confeccionar sus propios formularios, brindando así una mayor flexibilidad al sistema para adecuarse a cualquier centro educativo. Aunque tenemos la desventaja de contar siempre con la herramienta de software InfoPath para lograr tal flexibilidad.

Especificación de los requisitos del sistema.

Dependencias y relaciones.

El Subsistema de Gestión de Expediente está relacionado con un conjunto de subsistemas que forman el Sistema de Gestión Académica AKADEMOS, entre los que encontramos el Subsistema de Matricula, éste interactúa con el subsistema para crear un expediente del estudiante luego de ser este matriculado, además de

actualizar algunos archivos pertenecientes a los documentos oficiales que respaldan al estudiante durante toda su vida estudiantil ; el Subsistema de Plan de Estudio facilita a la aplicación los planes de estudios, estos benefician al sistema para registrar las plantillas a un determinado plan de estudio.; el subsistema de Registro permite actualizar en el expediente del estudiante lo referente a las evaluaciones de las asignaturas cursadas por un estudiante en el transcurso que es matrícula del centro; el Subsistema de Reportes será el que proporcionará los reportes estadísticos que solicite el usuario; por último el Subsistema de Seguridad y Gestión de Usuarios, el cual es el encargado de toda la seguridad de los subsistemas integrados en el Sistema de Gestión Académica (AKADEMOS), este tiene una relación estrecha con el Directorio Activo.

Requerimientos Funcionales

Los requerimientos funcionales definen las responsabilidades del sistema, o sea, las funciones que éste será capaz de realizar.

Requisitos funcionales

1. Registrar plantilla.
 - 1.1. El usuario especifica:
 - 1.1.1. Nombre de la plantilla.
 - 1.1.2. Dirección donde se encuentra la plantilla.
 - 1.1.3. Plan de estudio al que pertenece.
2. Crear expediente.
 - 2.1. El expediente lo crea el Subsistema de Matrícula una vez que matriculó al estudiante. Para esto especifica:
 - 2.1.1. CI del estudiante
3. Actualizar expediente
 - 3.1. Agregar documentos al expediente de un estudiante.
 - 3.1.1. Para esto el usuario especifica el tipo del documento
 - 3.1.2. Un documento puede ser:
 - 3.1.2.1. Documento con formato

- 3.1.2.2. Documento sin formato
- 3.1.3. Para agregar un documento con formato el usuario especifica los valores de los campos editables del documento.
- 3.1.4. Para agregar un documento sin formato el usuario especifica:
 - 3.1.4.1. Nombre del documento.
 - 3.1.4.2. Camino de la imagen del documento.
- 3.2. Modificar documento del expediente.
- 3.3. Eliminar documento del expediente.
- 4. Modificar estado de un expediente.
 - 4.1. El usuario especifica el Id del expediente.
 - 4.2. Id del nuevo estado.
- 5. Gestionar permisos
 - 5.1. Registrar permisos.
 - 5.1.1. El Administrador especifica
 - 5.1.1.1. Nombre de red del usuario
 - 5.1.1.2. Rol usuario
 - 5.1.1.3. Tipo de permiso que tendrá el usuario.
 - 5.1.1.4. Plantillas o documentos sobre los que ejercerá los permisos.
 - 5.1.1.5. Alcance del permiso.

Requerimientos no funcionales

Los requerimientos no funcionales responden a cualidades que el producto debe tener y las características para que este sea atractivo, confiable, usable y seguro.

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.

Requisitos de usabilidad

- El sistema Akademos debe ser concebido para soportar al menos dos idiomas (inglés y español), dando la posibilidad de ser extensible a otros idiomas con vistas a un posible intercambio con otros países.
- Documentar bien la aplicación y proporcionar materiales de ayuda para hacer mejor uso de todos los servicios que este ofrece.

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.

Requisitos de Soporte

- El sistema debe ser integrado, de manera que se mantenga la colaboración entre todos los subsistemas que lo componen.
- 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.
- Los documentos deben estar confeccionados en InfoPath.

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.

Requisitos Legales

- Este producto no violará ninguna Ley o Licencia de terceros Productos que puedan ser utilizados con él.

Casos de uso del sistema.

Los casos de uso son documentos narrativos que describen la secuencia de los eventos de un actor (agente externo) que utiliza un sistema para completar un proceso [ALV00].

Para la definición de los casos de usos se necesita:

1. Identificar los actores.
2. Identificar los casos de usos.
3. Describir los casos de usos.
4. Identificación de los Actores.

Los actores son el conjunto de usuarios, personas, máquinas, software, hardware, etc. que interactúan con el sistema, y que “puede intercambiar información o puede ser un recipiente pasivo de información” [ALV 00].

Actores	Descripción.
Administrador	Representa a un usuario que pertenezca al rol Administrador. Estos tipos de usuarios son los únicos usuarios que pueden realizar la gestión de la información de los usuarios del sistema, creando nuevos permisos para los usuarios o modificando los permisos existentes.
Subsistema Matrícula	Este actor se encarga de crear los expedientes del estudiantes, modificar es estado del expediente y actualizar determinados documentos del mismo.
Secretarias de las facultades Secretaria General	Actualizan determinados documentos del expediente, pertenecen al rol Actualizador.

Subsistema de Registro,	Actualizan determinados documentos del expediente, pertenece al rol Subsistema Actualizador.
Subsistema Generador de Informe	Es el subsistema encargado de generar todos los reporte de los subsistema integrados en el sistema AKADEMOS.
Actualizador	Representa a cualquier usuario humano del sistema.
Subsistema Actualizador	Representa a las componentes de software que necesite hacer uso de los servicios del subsistema de expediente.

Definición de los casos de usos.

A través de las siguientes tablas se enuncian y describen los casos de uso del sistema y se referencia con los requerimientos funcionales planteados.

CU-1	Crear expediente.
Actores	Subsistema Matricula
Descripción: El subsistema Matrícula solicita crear un expediente al Subsistema Expediente para un estudiante, brindando Carné de identidad del estudiante.	
Referencias	2

CU-2	Actualizar expediente
------	-----------------------

Actores	Actualizador, Subsistema Actualizador
<p>Descripción:</p> <p>Este caso de uso permite insertar, modificar o eliminar un documento en el expediente de un determinado estudiante.</p>	
Referencias	3

CU-3	Modificar estado del expediente
Actores	Subsistema Matrícula
<p>Descripción:</p> <p>Modificar el estado del expediente de un determinado estudiante.</p>	
Referencias	4

CU-4	Gestión plantilla
Actores	Actualizador
<p>Descripción:</p> <p>Este caso de uso permite registrar, modificar o eliminar plantillas por la que serán asociados los documentos del expediente.</p>	
Referencias	4

CU-5	Gestionar permisos
------	--------------------

Actores	Administrador
El Administrador gestiona los permisos de los usuarios del sistema agregando nuevos permisos o modificando los permisos existentes.	
Referencias	5

Casos de uso por ciclo.

Los ciclos de desarrollo se organizan en torno a los requerimientos de los casos de uso. Se asigna un ciclo para desarrollar uno o más casos, en dependencia de la complejidad que estos posean.

A continuación los casos de usos del primer ciclo.

Cód	Nombre de caso de uso	Justificación de la selección.
CU-1	Crear expediente	Estos casos de uso son prioritarios para el desarrollo del sistema, presentan mayor grado de complejidad, desempeñan las principales funcionalidades, permiten todo lo relacionado con el almacenamiento de la información e influyen en la arquitectura básica del sistema; por lo que se decide hacer su desarrollo en el primer ciclo.
CU-2	Actualizar expediente	
CU-3	Modificar estado	
CU-4	Gestionar plantilla	

Casos de uso del segundo ciclo.

Cód	Nombre de caso de uso	Justificación de la selección.
CU-5	Gestionar permisos	Como este caso de uso tiene menos complejidad que los anteriores y no es de tanta prioridad se decide dejar su desarrollo para el segundo ciclo.

Diagrama de casos de uso.

Los diagramas de casos de uso son importantes para visualizar, especificar y documentar el comportamiento de un elemento. Estos diagramas facilitan que los sistemas, subsistemas y clases sean abordables y comprensibles, al presentar una vista externa de cómo pueden utilizarse estos elementos en un contexto dado. Los diagramas de casos de uso también son importantes, para probar sistemas ejecutables a través de ingeniería directa y para comprender sistemas ejecutables a través de ingeniería inversa.

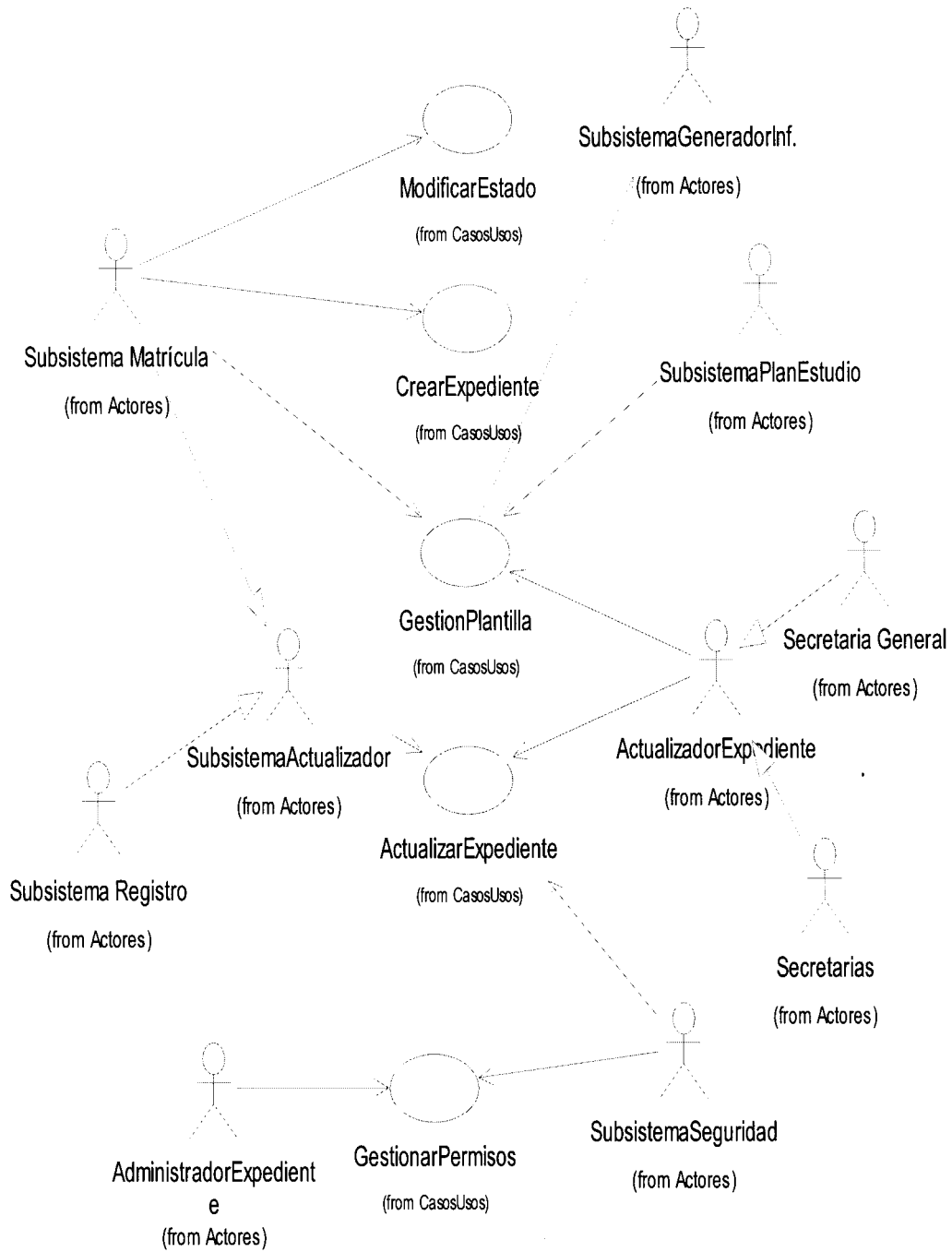


Diagrama de casos de uso.

Expansión de los casos de uso.

A través de la expansión de los casos de uso se describe paso a paso la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema [ALV00]. En el Anexo 1 se muestra la expansión de cada uno de los casos de uso del primer ciclo.

Conclusiones.

Este capítulo ha sido fundamental para determinar el objeto de estudio, de automatización, la información que se maneja, sistema propuesto, Se han detallado los procesos que forman parte del proceso de desarrollo de la aplicación, haciendo énfasis a través los requerimientos del sistema y la explicación detallada de los casos de uso, de cada uno de los pasos a seguir para obtener un producto con una calidad garantizada.

Capítulo 3. Análisis y diseño del Sistema.

Introducción

El análisis y diseño del sistema es una de los mecanismos más importantes dentro del proceso unificado del desarrollo de un software. Se consideran más cercanos al acceso de datos y a la arquitectura física de la aplicación.

El análisis forma parte del proceso de desarrollo de software, cuyo propósito primario es formular el modelo del dominio del problema. [LAR99]

El diseño es la etapa del proceso de desarrollo donde se decide cómo se llevará a cabo el sistema. A través de esta fase, se toman decisiones estratégicas y tácticas para cumplir los requerimientos funcionales y de calidad de un sistema. Con él se responde la pregunta de cómo hacer [ARZ02].

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.

Modelo de clases de análisis.

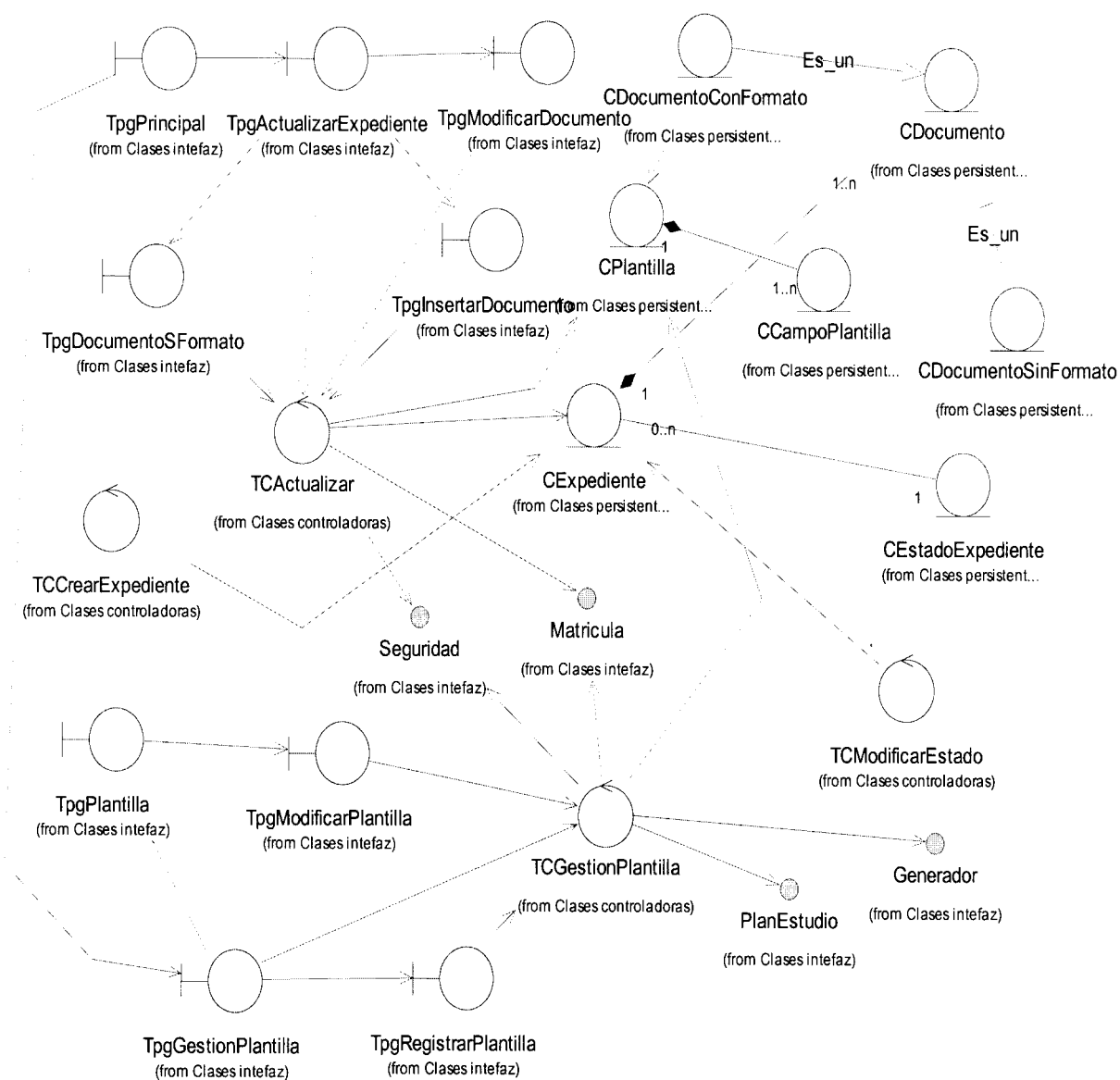


Fig 3.1 Diagrama de clases del análisis.

Diagramas de secuencia del sistema.

Con la idea de dar una visión gráfica de las interacciones de los actores con el sistema, se utilizan los diagramas de secuencia del sistema (DSS), los cuales muestran qué hace el sistema ante el medio, sin explicar el cómo. [ALV00]

Para cada uno de los casos de uso se define un diagrama de secuencia del sistema. Los diagramas de secuencias del sistema de cada caso de usos se muestran en el **Anexo # 2**.

Diagramas de clases del diseño.

Los diagramas de clases pertenecientes a los casos de usos podemos encontrarlos en el **Anexo # 3**.

Descripción de las clases.

En el **Anexo # 4**, se muestra las descripciones detalladas de las clases que componen los diagramas de clases de análisis y del diseño, con sus atributos y métodos.

Diseño de la Base de Datos (BD).

Una vez definidas las clases del sistema se puede determinar qué clases requieren que la información que poseen se mantenga a los largo del tiempo, entonces, para obtener las mismas se toman a las clases entidad que están involucradas en el sistema y se realiza el modelado de la Base de Datos.

En el **Anexo # 5** se muestra el diseño de la base de datos del sistema propuesto a través del diagrama entidad-relación.

Descripción de las tablas.

Podemos encontrar la descripción de las clases en el **Anexo # 6**.

Tratamiento de errores

El sistema hace uso de las potencialidades que brinda ASP.NET para la validación de formularios. El marco de trabajo de los formularios Web incluye un conjunto de controles de servidor de validación que permiten de un modo sencillo y potente comprobar errores en los formularios de entrada y, en caso necesario, mostrar mensajes al usuario.

Los controles de validación se agregan a una página de formularios Web con otros controles de servidor. Existen controles para tipos concretos de validación, como la comprobación de intervalos o la coincidencia de modelos, además de CustomValidator, que se asegura de que un usuario no omita un campo de entrada.

Nombre del control	Descripción
CustomValidator	Se asegura de que el usuario no omita ninguna entrada.
ValidationSummary	Muestra los errores de validación en forma de resumen para todos los validadores de la página.

Tabla 5. Controles de validación

Seguridad

El acceso a las diferentes secciones de la información será en dependencia de los derechos que cada uno posea, el Subsistema Seguridad es el encargado de mantener la seguridad del sistema Akademos. A la página principal de la aplicación solo tienen acceso aquellas personas que puedan gestionar el expediente, a las demás páginas será por el nivel de privilegio que tengan los usuarios. A la página de gestión de permiso solo se puede acceder con una cuenta de administración, que en este caso la tendrán los administradores del sistema.

Conclusiones:

En este capítulo se han mostrado los resultados de la etapa de análisis y diseño, expandiéndose los casos de usos determinados para el primer ciclo de desarrollo del sistema. Además de realizar los diagramas de secuencias del sistema donde se representan las operaciones del mismo.

Conclusiones.

Una vez culminada la investigación y desarrollo del sistema, se llegó a la conclusión de que el objetivo fundamental de éste fue logrado. La automatización del primer ciclo del “Modulo Expediente virtual”, que permite garantizar un mejor registro, control, análisis y acceso a la información que se tiene almacenado en los expedientes.

Se considera además que el presente trabajo ha cumplido con todos los objetivos propuestos:

- Se identificaron todas las necesidades de los clientes.
- Los procesos y especificaciones del sistema han sido descritos.
- Se alcanzó una definición de los conceptos más significativos en el dominio del problema.
- Se definió el comportamiento del sistema en su interacción con el medio, esto se realizó mediante los Diagramas de secuencias.
- Al definirse las clases y sus relaciones en el Diagrama de clases del análisis, se obtuvo el diseño de la Base de Datos para implementar el núcleo central.
- La principal característica que se logró del sistema es su flexibilidad, a través de plantillas, para crear documentos según las necesidades de los usuarios.
- Con la implementación del primer ciclo de desarrollo queda automatizado el proceso de gestión de documentos.

Teniendo en cuenta lo anterior, el subsistema Expediente se convierte en una herramienta muy poderosa y útil para la gestión de expediente de la universidad, este facilita en gran medida el trabajo de los empleados encargados de la administración del mismo, garantizando mayor rapidez en los procesos de selección, adquisición y control de documentos del expediente.

Recomendaciones.

Se recomienda a las personas que van a dar continuidad a este trabajo:

- Concluir con la etapa de prueba del primer ciclo, y desarrollar el segundo ciclo de desarrollo completo.
- Investigar sobre las firmas digitales.

Referencias Bibliográficas.

[ABC] “ABC de Oracle” en <http://www.programatium.com/oracle.htm>

[ALV00] - Álvarez, Sofía, Hernández Anaisa. Metodología para el desarrollo de aplicaciones con tecnología Orientada a Objetos utilizando notación UML. La Habana, 2000

[ARZ02] - Arza, Lizandra, Ortiz, Aeleen. SOSOFT, herramienta para el Soporte de Software. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de la Habana, Junio 2002.

[BAS]: “Bases de Datos” en <http://www.abcdatos.com/tutoriales/tutorial/16455.html>

[BOC00] - BOCH Grady, RUMBAUGH James, JACOBSON Ivar. “El lenguaje unificado de modelado. Manual de referencia”.2000. Addison Wesley.

[CON]: CONALLEN, Jim. “Modeling Web Applications with UML”, en <http://www.conallen.org/whitepapers/webapps/ModelingWebApplications.htm>

[INEI] - Instituto Nacional de Estadística e Informática. Lima. Arquitectura Cliente – Servidor. Metodologías informáticas, Perú.

<http://www.inei.gob.pe/cpi/ibancopub/libfree/lib616/index.html> (20/5/04)

[IWFP]: “Introduction to Web Forms Pages”, MSDN, Abril.

[LAR99] - Larman, C. UML y Patrones. Prentice Hall, México, 1999.

[MOR] - MORAN, Brian. Análisis comparativo entre Microsoft® .NET y Sun® J2EE

[QRJ] Quintana Rodríguez, Julio T. “Sistemas distribuidos con componentes”, en <http://www.uv.mx/iiesca/revista4/distribuidos.htm>

[SPH] -SALINAS Caro, Patricio y Histchfeld K, Nancy Tutorial de UML

[TUML] - Tutorial de UML <http://www.dcc.uchile.cl/~psalinas/uml>

Bibliografía.

1. C. Larman, UML y patrones, introducción al análisis y diseño orientado a objetos, Segunda edición, Prentice-Hall, 2002.
2. Pressman, Roger S. Ingeniería de Software, un enfoque práctico. Cuarta Edición.
3. Diagramas de Casos de Uso. <http://www.vico.org/MuestrarioDiagCU.pdf>.
4. 10 razones para utilizar Office InfoPath 2003.
<http://www.microsoft.com/spain/Office/infopath/top10.asp>
5. Arquitectura de Microsoft Office InfoPath 2003
http://www.microsoft.com/spanish/msdn/articulos/archivo/011003/voices/odc_inArch.asp
6. Educación y NTIC. Desarrollo de proyectos para la extensión de la actividad docente universitaria basados en AGORA
<http://www.rediris.es/rediris/boletin/50-51/ponencia2.html>
7. Módulos GAUSS
8. http://www.semicrol.es/gestion_universitaria2/gauss/gauss.html#
05.07.2004.
9. Desarrollo de Software – Sistemas. Sistema Académico para Facultades
<http://www.cnc.una.py/desoft/sacad.html> 08.07. 2004.
10. Sistema de Información Académica <http://www.siaun.unal.edu.co/>
08.07. 2004.
11. Manual de ASP <http://www.webestilo.com/asp/> 23.06.2004.
12. Sistema de información de gestión académica.
<http://www.siga.utfsm.cl/> 05.07.2004..
13. S.I.G.A. Principal. Sistema Integrado de Gestión Académica.

- <http://www.dara.es/siga/> 05.07.2004.
14. GAT Universidades. Gestión Académica para universidades.
<http://www.uoc.edu/ideasolutions/b/menu131.htm> 08.07. 2004.
15. Curso de SQL <http://www.cybercursos.net/sql/sql.html> 08.07. 2004.
16. Tutorial de SQL <http://www.desarrolloweb.com/manuales/9/> 08.07. 2004.
17. Estructuras de Oracle <http://www.programacion.com/tutorial.php?id=oracle>
08.07. 2004.

Glosario de Términos.

Actor: Alguien o algo, fuera del sistema o negocio que interactúa con el sistema o negocio.

AGORA: Aplicación de Gestión de Organización Académica.

ASP: (*Active Server Page*) Es una tecnología de programación Web con la cual se logra interactuar con el usuario del sitio. ASP provee al programador de una gran cantidad de componentes y objetos con los cuales se puede manejar de manera sencilla la interacción del Navegador y el servidor Web. Lenguajes de scripts como JavaScript o VBScript se usan para manipular estos objetos.

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

Cliente: Una persona u organización, interna o externa a la organización productora que toma responsabilidad financiera por el sistema. El cliente es el último destinatario del producto desarrollado y sus artefactos.

CLR: (*Common Language Runtime*). Es el núcleo de la plataforma .NET. Es el motor encargado de gestionar la ejecución de las aplicaciones para ella desarrolladas y a las que ofrece numerosos servicios que simplifican su desarrollo y favorecen su fiabilidad y seguridad.

DER: Diagrama entidad relación.

DSS: Diagrama de secuencias del sistema.

FTP: Protocolo de transmisión de ficheros.

GAT Universidades : Gestión Académica para Universidades.

GAUSS: Gestión Académica Universitaria sobre Sistemas Informáticos

GESTACAD: Sistema de Gestión Académica.

HTML: *HyperText Markup Language* (Lenguaje de Marcado de Hipertexto) Lenguaje en el que se escriben las páginas a las que se accede a través de navegadores *WWW*. Admite componentes hipertextuales y multimedia.

HTTP: Protocolo de transferencia de datos a través de los navegadores web.(HiperText Transfer Protocol).

Interfaz: Frontera convencional entre dos sistemas o dos unidades, que permite intercambio de informaciones.

Internet: Es la mayor red de computadoras que existe, ya que se compone de infinitas redes pequeñas a lo largo de todo el planeta, interconectadas entre sí. Esta permite a los usuarios compartir información de todo tipo.

Intranet: (*Intranet, Intranet*). -- Red propia de una organización, diseñada y desarrollada siguiendo los protocolos propios de Internet, en particular el protocolo *TCP/IP*. Puede tratarse de una red aislada, es decir no conectada a Internet

ON LINE: En línea o sea que puede ser accedido por la Web.

Proceso: Secuencia de actividades invocadas para producir un producto de software.

Protocolo: Descripción formal de formatos de mensaje y de reglas que dos ordenadores deben seguir para intercambiar dichos mensajes. Un protocolo puede describir detalles de bajo nivel de las interfaces máquina-a-máquina o intercambios de alto nivel entre programas de asignación de recursos.

Rol: Papel, cometido o función que tiene o desempeña que interpreta un actor

RUP: El Proceso Unificado Rational (RUP) es una metodología de desarrollo para la programación orientada a objetos. Según Rational (diseñadores de Rose Rational y el Idioma Modelado Unificado), RUP está como un mentor en línea que mantiene pautas, plantillas, y ejemplos de todos los aspectos y fases de desarrollo del programa. RUP es un software comprensivo que diseña herramientas que combinan los aspectos procesales de desarrollo (como las fases definidas, técnicas, y prácticas) con otros componentes de desarrollo (como los

documentos, modelos, manuales, el código, y así sucesivamente) dentro de un armazón unificándose.

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

Servicios Web: (*Web Services*). Son aplicaciones independientes de la plataforma que pueden ser fácilmente publicadas, localizadas e invocadas mediante protocolos Web estándar, como XML, SOAP, UDDI o WSDL.

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

Software:(*Componentes lógicos, programas, software*). -- Programas o elementos lógicos que hacen funcionar un ordenador o una red, o que se ejecutan en ellos, en contraposición con los componentes físicos del ordenador o la red.

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

UCI: Universidad de las Ciencias Informáticas.

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

UML: Lenguaje unificado de modelado -Unified Modeling Language.

URL: (*Uniform Resource Locator*) -- (Localizador Uniforme de Recursos) Sistema unificado de identificación de recursos en la red. Las direcciones se componen de protocolo, FQDN y dirección local del documento dentro del servidor. Este tipo de direcciones permite identificar objetos *WWW, Gopher, FTP, News,...* Ejemplos de URL son: <http://cubasi.cu> o <ftp://ftp.ati.es>.

Usuario: Persona que usa ordinariamente una cosa.

XML: *eXtensible Markup Language* (Lenguaje Extensible de Marcado). Metalenguaje que sirve para crear lenguajes. No es una extensión ni un componente de *HTML*.

.NET: .NET es una plataforma de servicios Web XML que se compone de: una plataforma de software para generar experiencias .NET, un modelo de programación y herramientas para generar e integrar servicios Web XML y un conjunto de servicios Web XML programables. Es una forma de permitir que los usuarios interactúen con una amplia gama de dispositivos inteligentes mediante la Web, asegurando al mismo tiempo que el usuario controla la interacción, más que la aplicación.

.NET Framework: El .NET Framework de Microsoft es un ambiente para desarrollar, implementar y ejecutar servicios Web XML y otras aplicaciones. Consiste de tres partes principales: el tiempo de ejecución del lenguaje común, los tipos de Framework, y ASP.NET. Como una infraestructura complementaria, el .NET Compact Framework de Microsoft, es un conjunto de interfases de programación que permiten a los desarrolladores apuntar a dispositivos móviles como teléfonos inteligentes y PDA's.

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.

Anexos.

Anexo #1. Casos de uso expandidos.

Caso de uso	Crear expediente	
Actores	Subsistema Matrícula	
Propósito	Crear un expediente para un estudiante de nuevo ingreso.	
Resume	El subsistema Matrícula solicita crear un expediente para un estudiante, brindando los datos necesarios para la creación del mismo.	
Referencias	RF2.	
Acción del actor	Respuesta del sistema	
1. El caso de uso inicia cuando el subsistema Matrícula solicita crear un expediente para un estudiante de nuevo ingreso, brindando el Id del mismo.	2. El sistema verifica que no tenga asociado ya un expediente, crea el expediente para el este estudiante, devolviendo un valor según los resultados de la operación.	
Precondición	El estudiante debe existir.	
Poscondición	Se ha creado un expediente.	
Caminos Alternativos	Si en el paso 2 el sistema detecta que el estudiante tiene un expediente asociado, entonces devuelve un valor de error.	

Caso de uso	Modificar estado del expediente	
Actores	Subsistema Matrícula	
Propósito	Modificar el estado del expediente de un determinado estudiante.	
Resumen: La función de este caso de uso es cambiar el estado de un expediente.		
Referencias	RF 4	
Diagrama de transición de estados da la entidad Expediente: Ver Anexo # 6		
Acción del actor	Respuesta del sistema	
1. El caso de uso da inicio cuando el subsistema Matrícula solicita modificar el estado de un expediente, brindando el Id del estudiante al cual se le modificará el estado del expediente, Id del nuevo estado.	2. El sistema valida que el tipo de estado exista, que sea posible modificar el estado actual del expediente, modifica el estado del expediente, crea una incidencia y devuelve un valor según el resultado de la operación.	
Precondición Debe existir un expediente asignado para este estudiante.		
Poscondición: Se ha modificado el estado del expediente.		
Caminos Alternativos Si en el paso 2 el sistema detecta que el tipo de estado no es válido o que no sea posible modificar el estado, entonces devuelve un valor de error.		

Caso de Uso	Actualizar expediente	
Actores	SubsistemaActualizador, Actualizador,	
Propósito	Actualizar el expediente de un estudiante.	
Resumen: Este caso de uso permite insertar, modificar o eliminar un documento en el expediente de un determinado estudiante.		
Referencias	RF 3	
Si el actor es SubsistemaActualizador ver Sección "Subsistema". Si el actor es Actualizador ver Sección "Actor Humano".		
Sección "Actor Humano"		
Acción del actor	Respuesta del sistema	
1. El caso de uso comienza cuando el actor solicita actualizar el expediente, especificando el carné del estudiante al que se le actualizará el expediente.	2. El sistema valida que el expediente pueda actualizarse, muestra la página "Actualizar expediente" con un listado de los documentos que tiene asociado el expediente, un listado de los nombres de las plantillas existentes, no se mostrarán las plantilla que solo permiten un documento y ya el expediente lo tiene asociado. Se le brindan las posibilidades de insertar un nuevo documento(A), modificar uno existente (B), SinFormato(C), Eliminar(D)).	
3. El Actualizador escoge la opción deseada, puede seleccionar un documento o un conjunto de ellos si la opción que va a escoger es Eliminar.	4. Si la opción seleccionada es: A: Ver sección "Insertar". B: (Ver sección "Modificar") C: Ver sección "Documentos sin formato". D: Se hace una pregunta de confirmación, se eliminan los documentos.	

Sección: "Insertar"	
	5. El sistema le muestra la página "Insertar Documentos" con la plantilla seleccionada y las opciones de Insertar(A), Cancelar (B).
6. El actor llena los datos correspondientes a los campos de la plantilla.	7. Si el sistema escoge: A: El sistema obtiene los datos necesarios y almacena los mismos. B: El sistema le muestra la página "Actualizar expediente"
Sección: "Modificar"	
	8. El sistema, le muestra la página "Modificar documento" con el documento seleccionado por el actor y las opciones Modificar(A), Eliminar (B).
9. El actor hace los cambios requeridos en el documento y escoge una opción..	10. Si la opción escogida por el actor es: A: Se verifican los datos, se almacenan los nuevos datos, luego se crea una incidencia para el expediente B: Se le muestra la página de "Actualizar expediente".
Sección Documento sin formato	
	11. El sistema le muestra la página "Documento sin formato", con las opciones de insertar(A), cancelar (B).
12. El actor especifica el nombre del documento, y dirección donde se encuentra el mismo.	13. Si el actor escoge: A: El sistema almacena el documento, devolviendo un valor según el éxito de la operación. B: El sistema le muestra la página "Actualizar expediente".

Sección "Subsistema"	
14. El subsistema actualizador pide al nuestro sistema, actualizar un expediente, pasando por parámetros CIEstudiente, ID de la plantilla por la que llenó y los datos del documento.	15. El sistema verifica que se pueda crear el documento en función del máximo permitido y almacena los datos referentes a dicho documento.
Precondición Debe existir un expediente asignado para este estudiante.	
Poscondición: Se ha actualizado el expediente.	
Caminos Alternativos	

Caso de Uso	Gestión plantilla
Actores	Actualizador
Propósito	Almacenar o actualizar las plantillas creadas por el usuario.
Resume El objetivo fundamental de este caso de uso es registrar, Modificar o eliminar plantillas confeccionadas por el usuario a través de la aplicación InfoPath determinada para la misma.	
Referencias	RF 1
Acción del actor	Respuesta del sistema
1: El caso de uso comienza cuando el Actualizador solicita la opción de gestionar las platillas.	El sistema le muestra la página "Gestión de plantillas", esta contiene un listado de con los nombres de las plantillas que se encuentran activas y al plan de estudio a que pertenece cada una, además de las

	posibilidades de Modificar(A), Registrar (B) o eliminar(C).
3 El usuario escoge la opción deseada, en caso de que desee eliminar puede seleccionar más de una plantilla.	4.Si el Actualizador escoge: A: Ver sección de "Modificar". B: Ver sección de "Registrar". C: Se hace una pregunta de confirmación y en caso afirmativo se eliminan las plantillas seleccionadas por el actor, devolviendo un valor de acuerdo al éxito de la operación realizada, se crea una incidencia en el expediente.
Sección "Registrar".	
Acción del actor	Respuesta del sistema
	5. El sistema le muestra la página "Registrar plantilla", mostrándole un listado con los diferentes planes de estudios que hay hasta el momento, el actor debe especificar nombre de la plantilla a insertar, dirección donde se encuentra la misma, plan de estudio al que pertenece, con la opción de Registrar(A) o Cancelar (B).
6. El actualizador especifica el nombre, dirección de la plantilla, plan de estudio al que esta pertenece y escoge la opción deseada.	7.Si el actualizador escoge: A: El sistema busca si hay coincidencia en el nombre con otra existente, si existe, le pregunta al usuario si realmente desea sustituir la nueva plantilla por la existente, si la respuesta

	<p>es afirmativa, la nueva plantilla se pone como activa, se almacenan los datos correspondientes, crea todos los permisos sobre la plantilla para el usuario que está haciendo uso del sistema, se crea una incidencia.</p> <p>Solicita al subsistema Reporteador registrar un reporte para la plantilla, brindando los datos requeridos.</p>
Sección "Modificar"	
	<p>8. El sistema le muestra la página "Plantilla" que contiene el formato de la plantilla escogida por el actor y este decide si realmente quiere modificar la plantilla, en caso de una respuesta afirmativa el sistema le muestra la Pagina "Modificar plantilla", el actor debe especificar dirección donde se encuentra la plantilla nueva, y se le brindan las siguientes opciones, Aceptar(A) o Cancelar (B).</p>
<p>9. El Actualizador inserta la dirección donde se encuentra la plantilla y escoge una opción.</p>	<p>A: El sistema busca en la dirección especificada la nueva plantilla, activa esta y desactiva la anterior, crea una incidencia de modificación.</p> <p>B: Se le muestra la página "Gestionar plantilla."</p>
<p>Precondición: Debe estar creada la plantilla.</p>	
<p>Poscondición: Se ha registrado o actualizado una plantilla.</p>	

Caminos alternativos:

1. Si en el paso 4 el sistema determina que el camino no es valido devuelve un mensaje de error.

Anexo #2. Diagrama de secuencias del sistema.

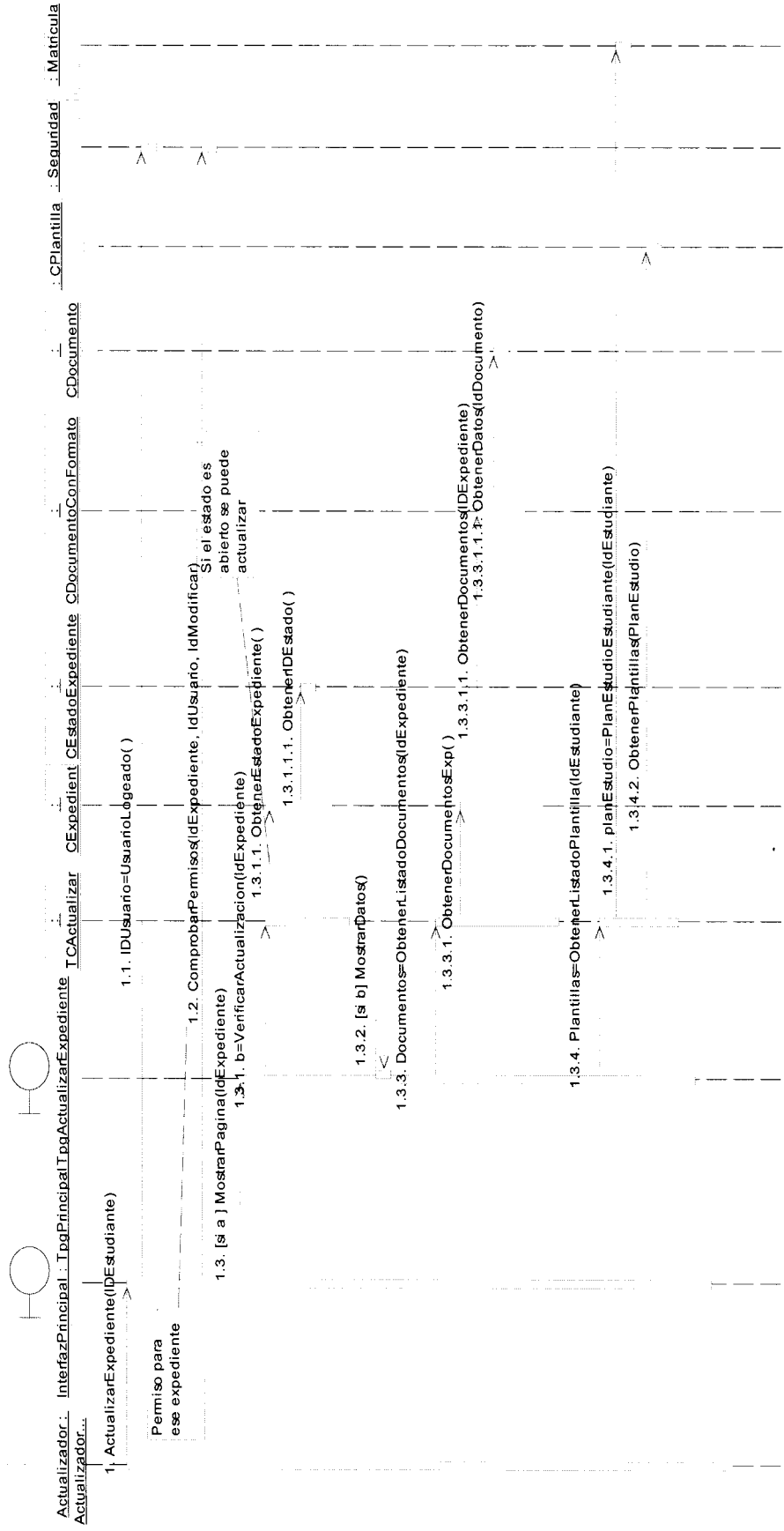


Diagrama de secuencia del caso de uso "Actualizar expediente": Sección "Actor humano".

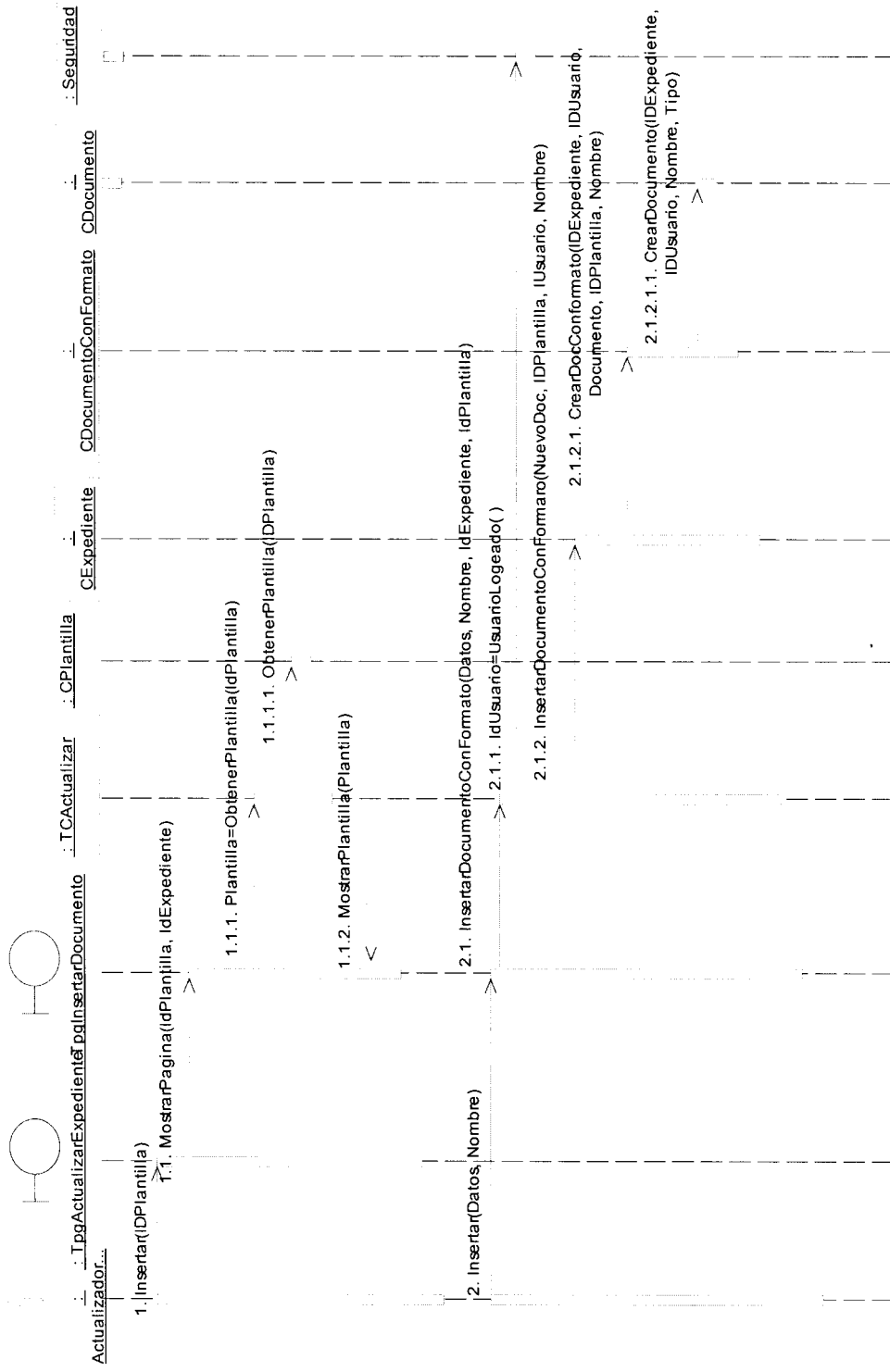


Diagrama de secuencia del caso de uso "Actualizar expediente": Sección "Actor humano": Sección Insertar.

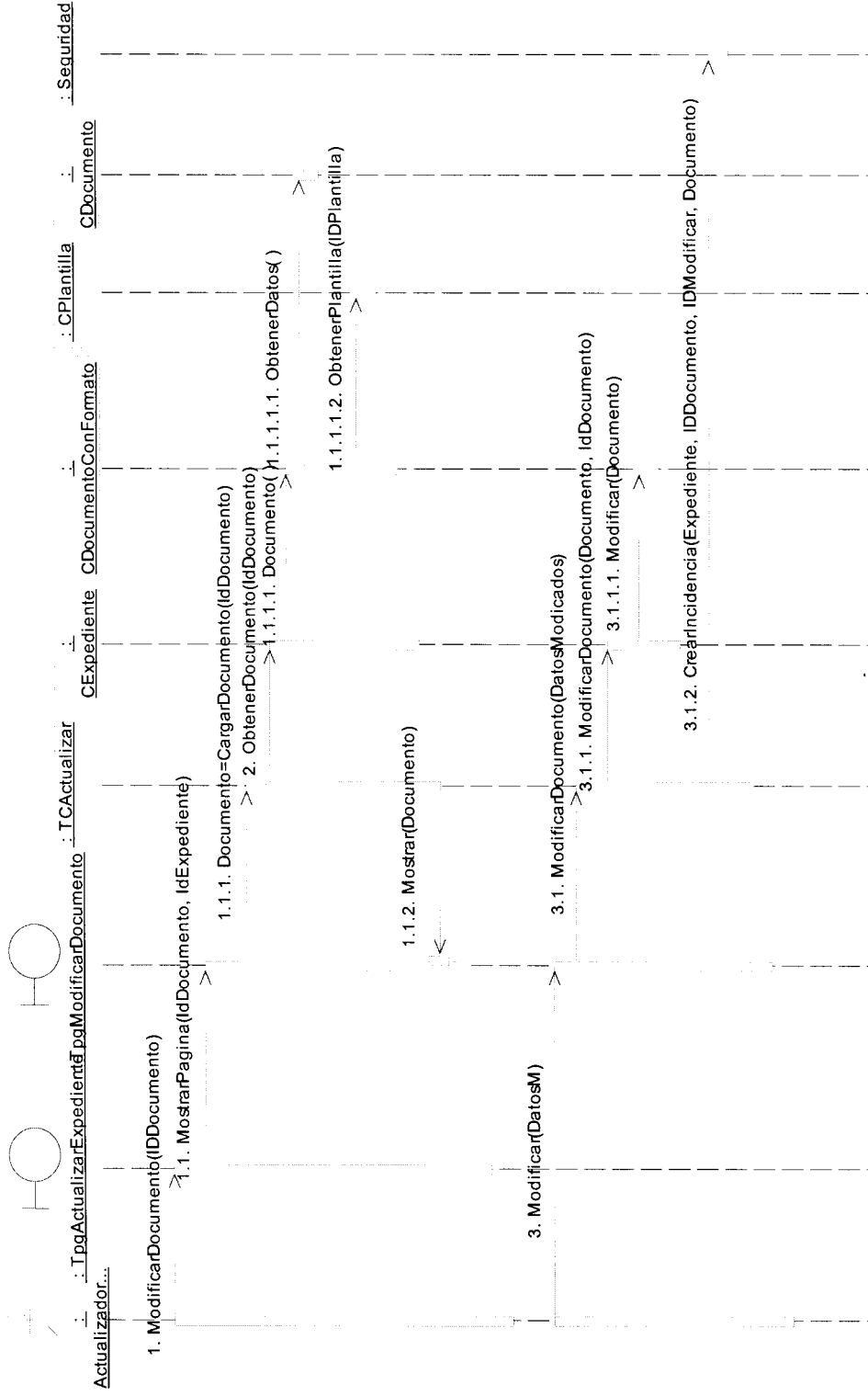


Diagrama de secuencia del caso de uso "Actualizar expediente": Sección "Actor humano". Sección Modificar.

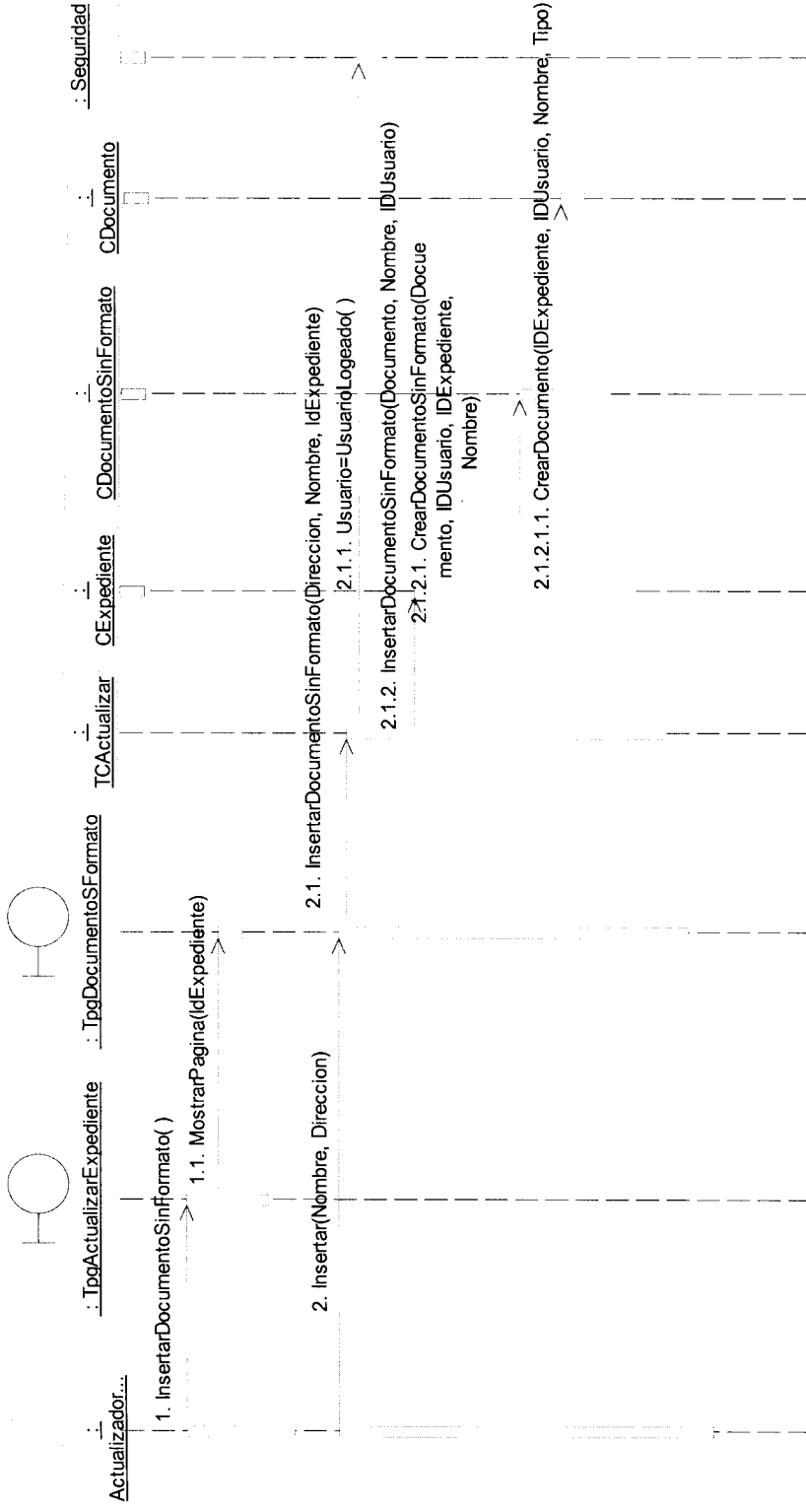


Diagrama de secuencia del caso de uso "Actualizar expediente": Sección "Actor humano". Sección DocumentoSinFormato.

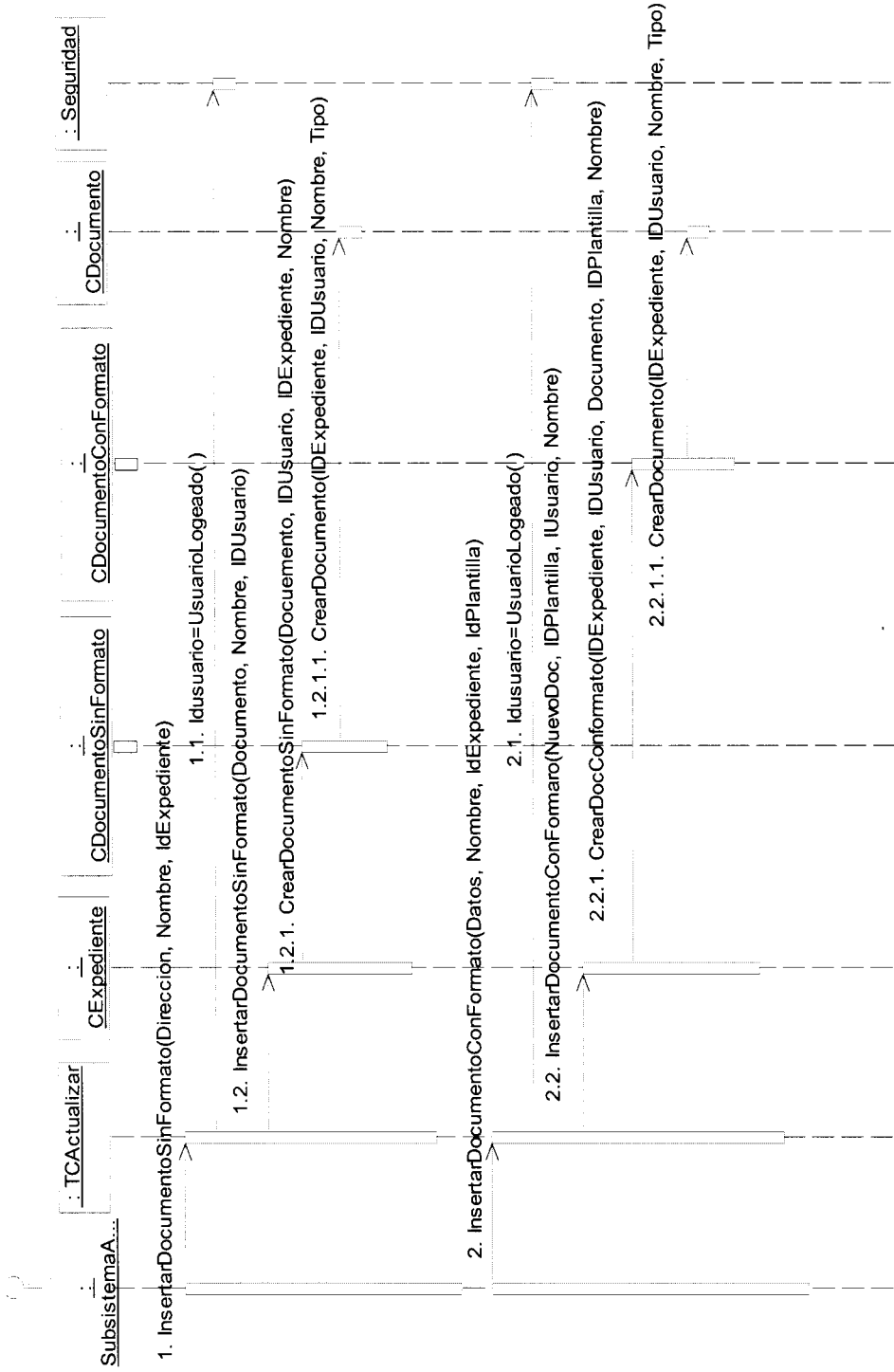


Diagrama de secuencia del caso de uso "Actualizar expediente": Sección "Subsistema".

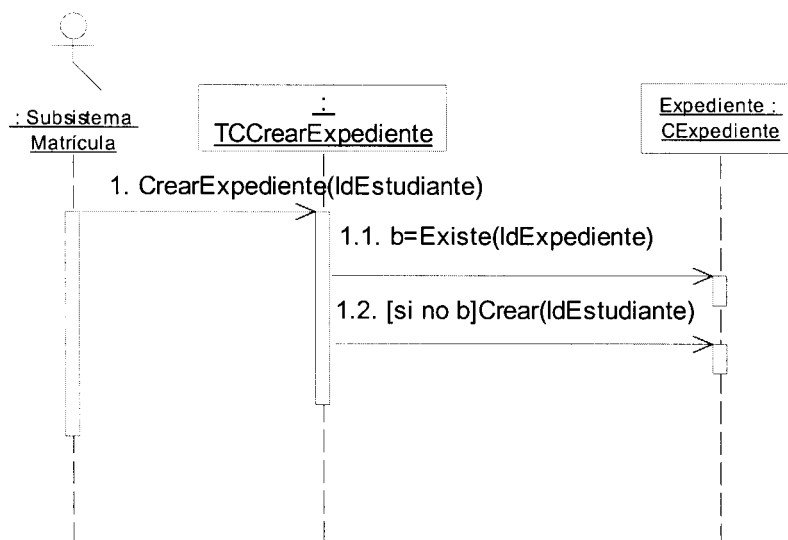


Diagrama de secuencia perteneciente al caso de uso “Crear expediente”.

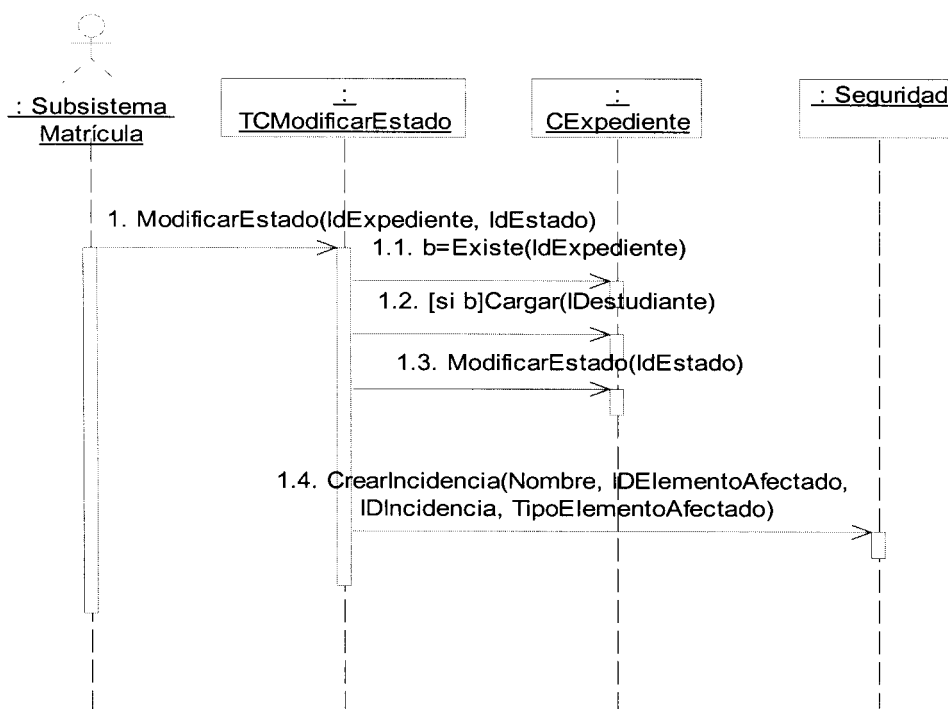


Diagrama de secuencia perteneciente al caso de uso “Modificar estado expediente”.

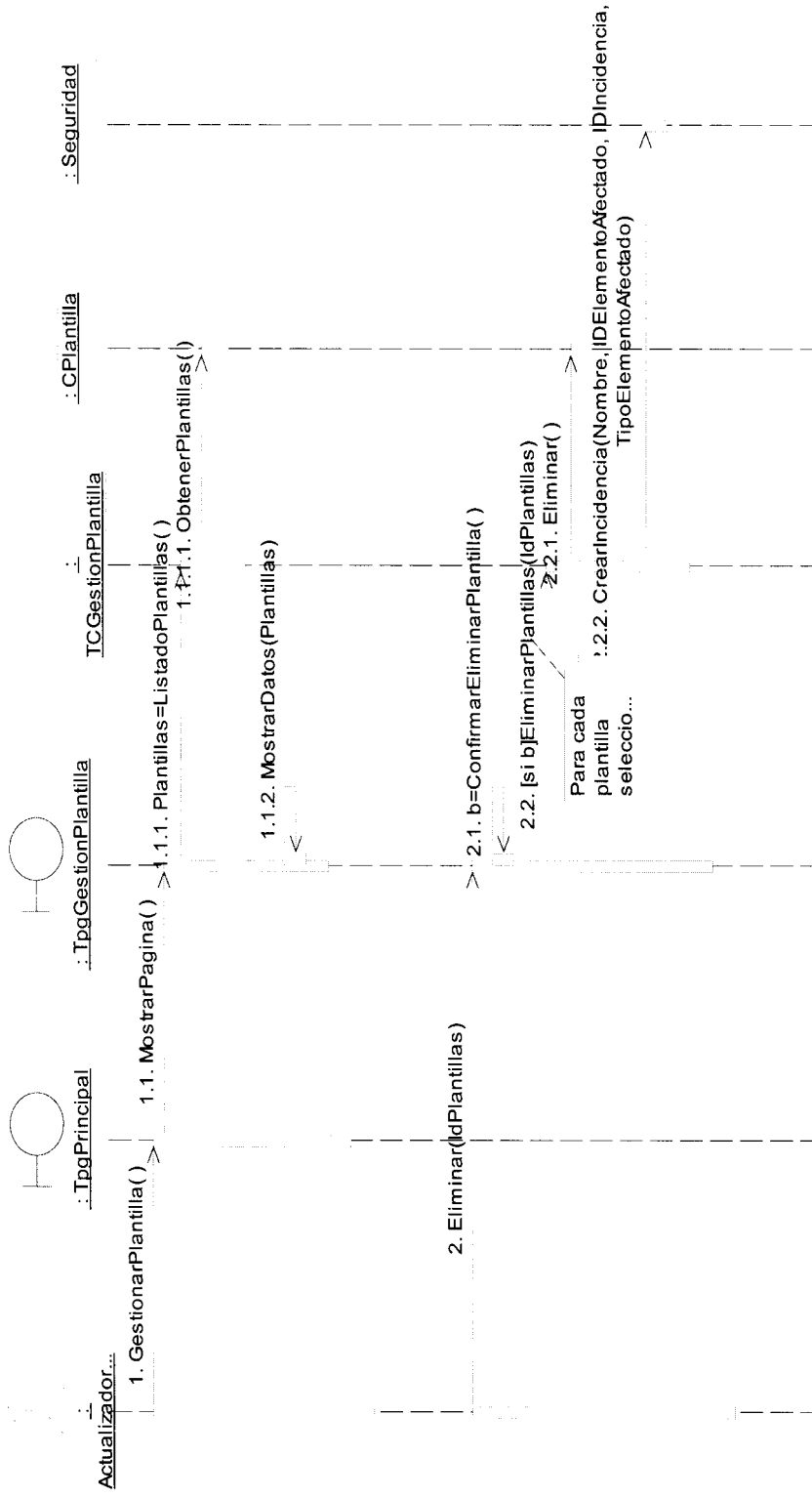


Diagrama de secuencia perteneciente al caso de uso "Gestionar plantilla".

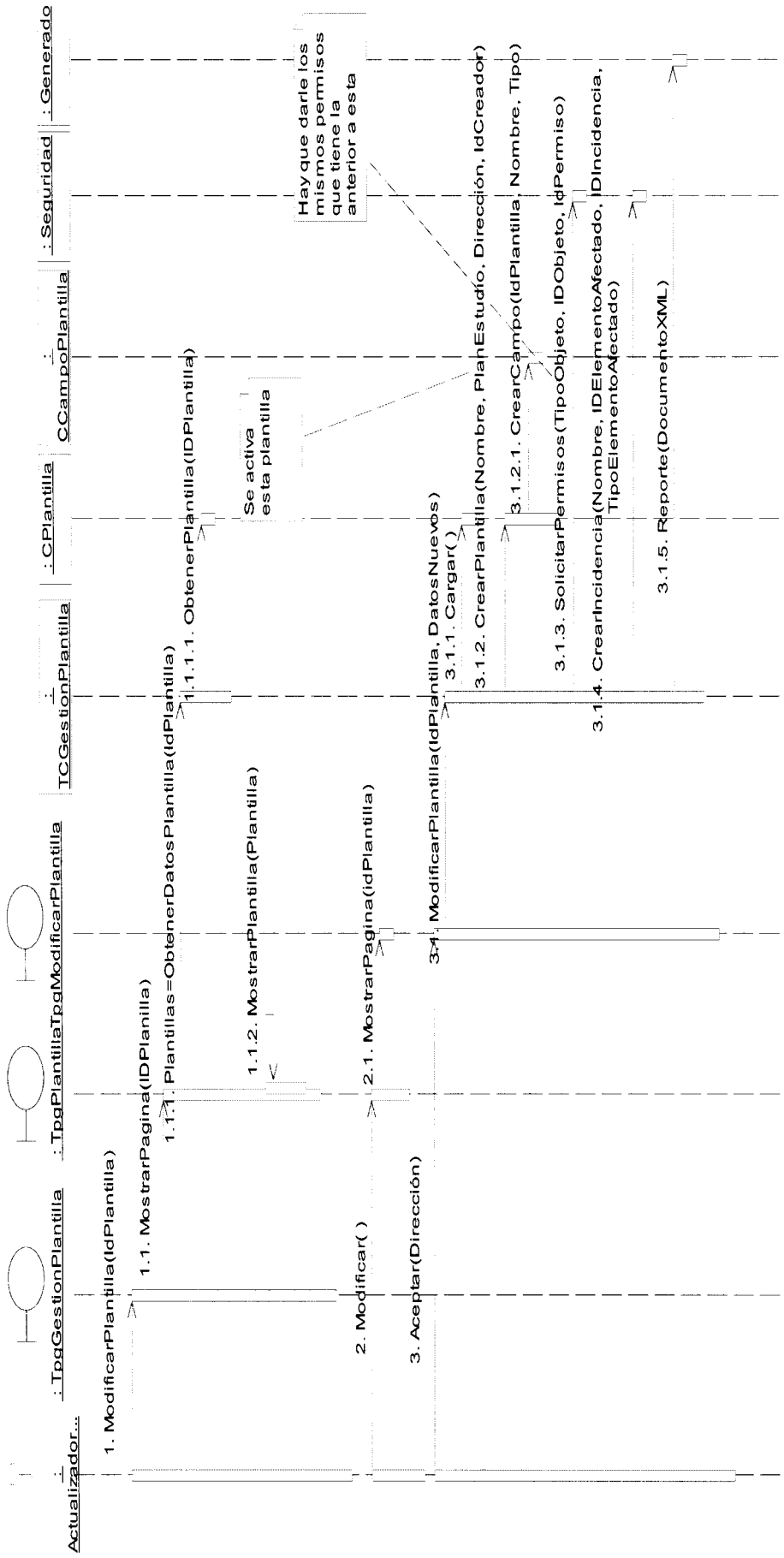


Diagrama de secuencia perteneciente al caso de uso "Gestionar plantilla", sección "Modificar plantilla".

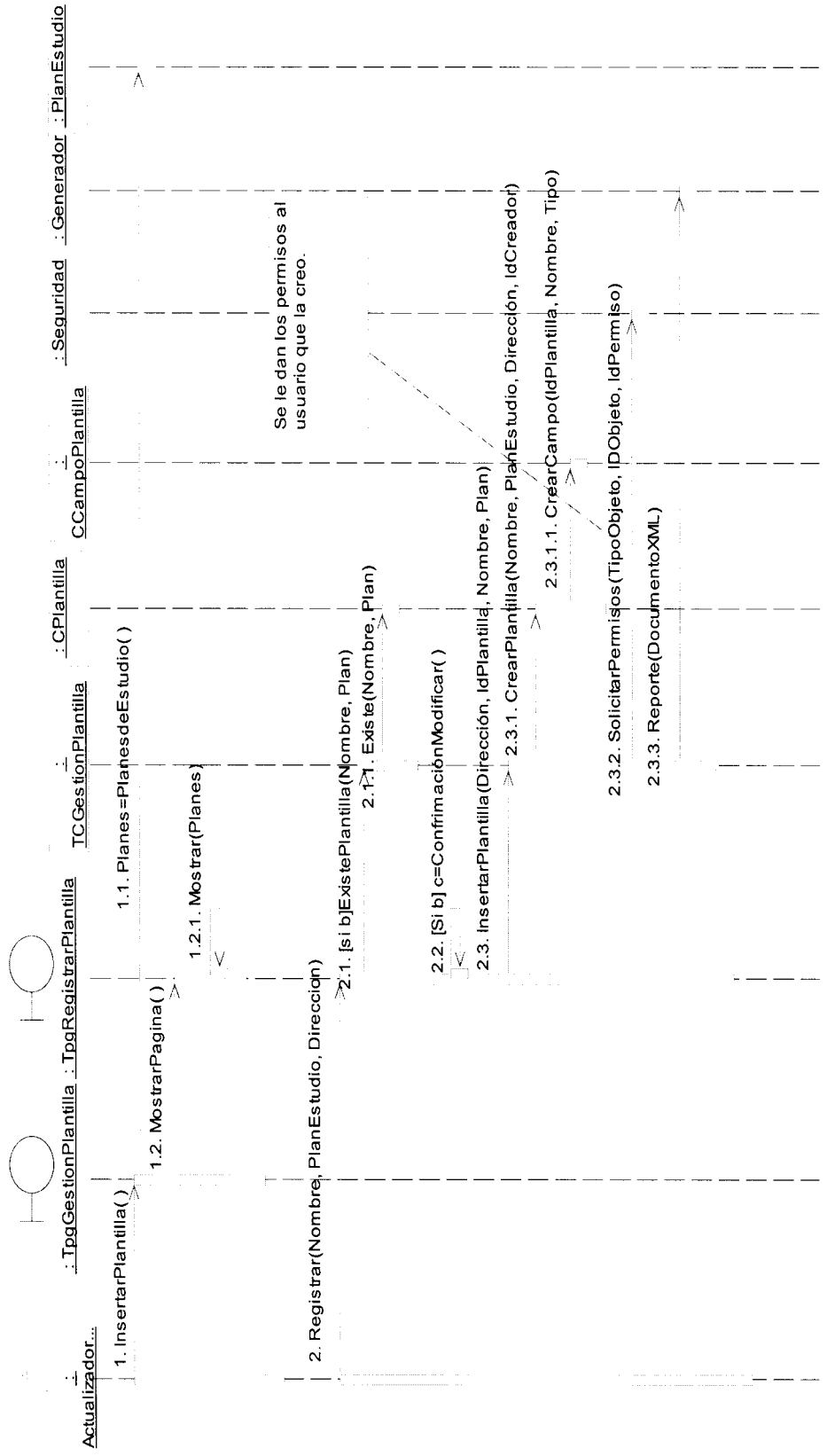


Diagrama de secuencia perteneciente al caso de uso "Gestionar plantilla", sección "Registrar plantilla".

Anexo#3 : Diagrama de clases.

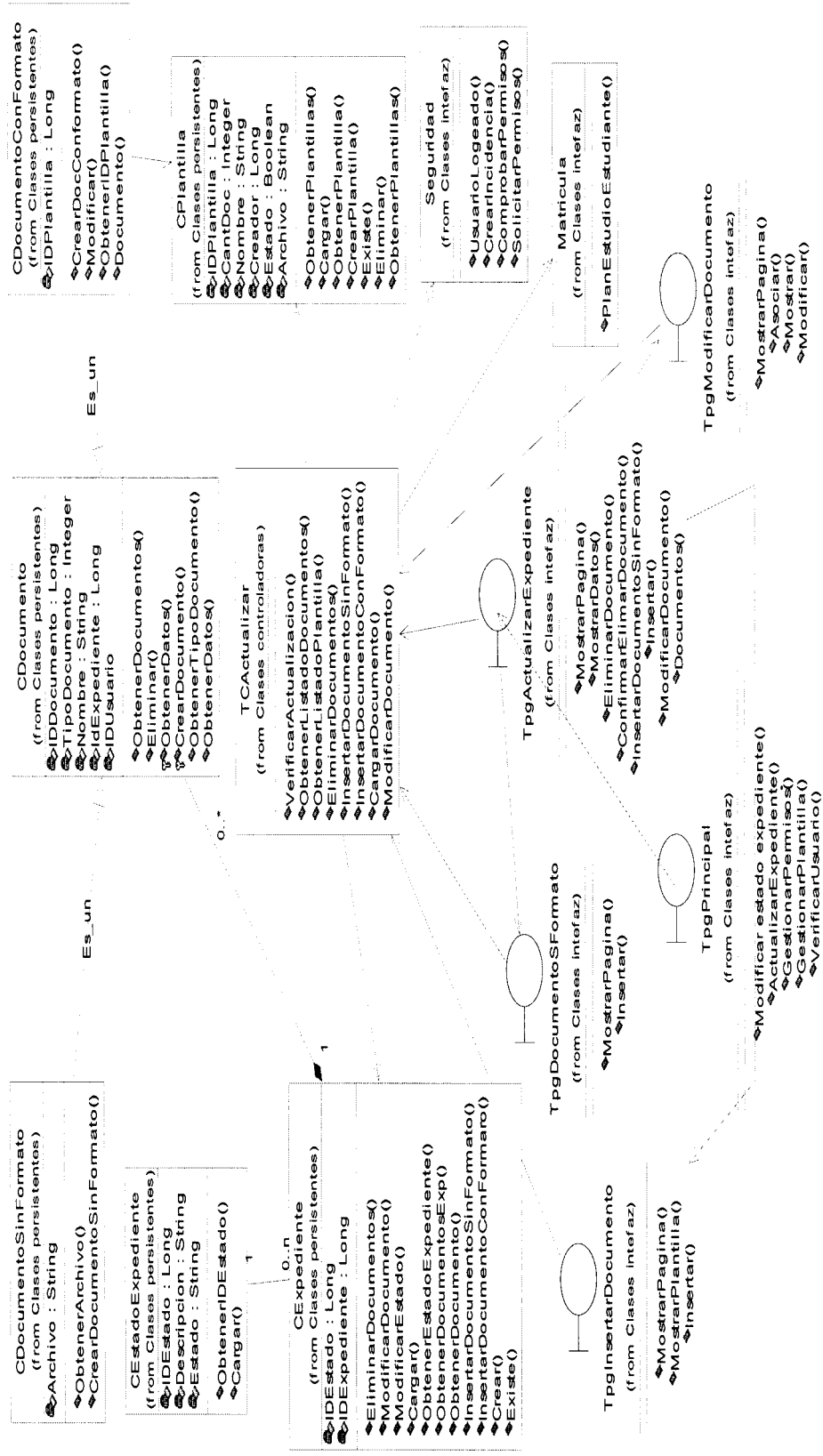


Diagrama de clases del caso uso "Actualizar expediente".

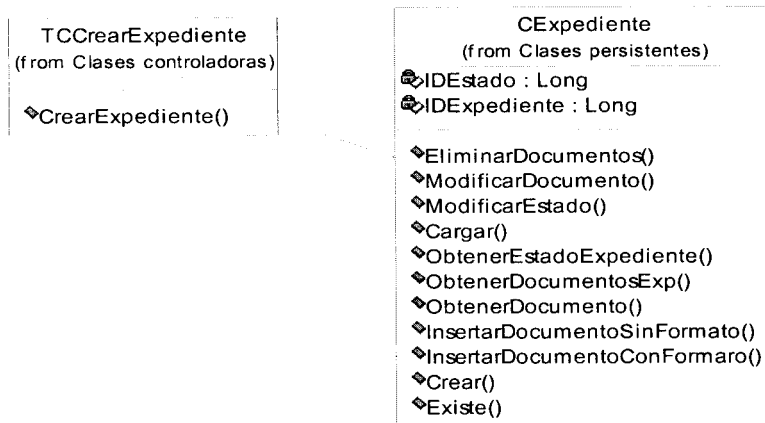


Diagrama de clases del caso de uso "Crear expediente".

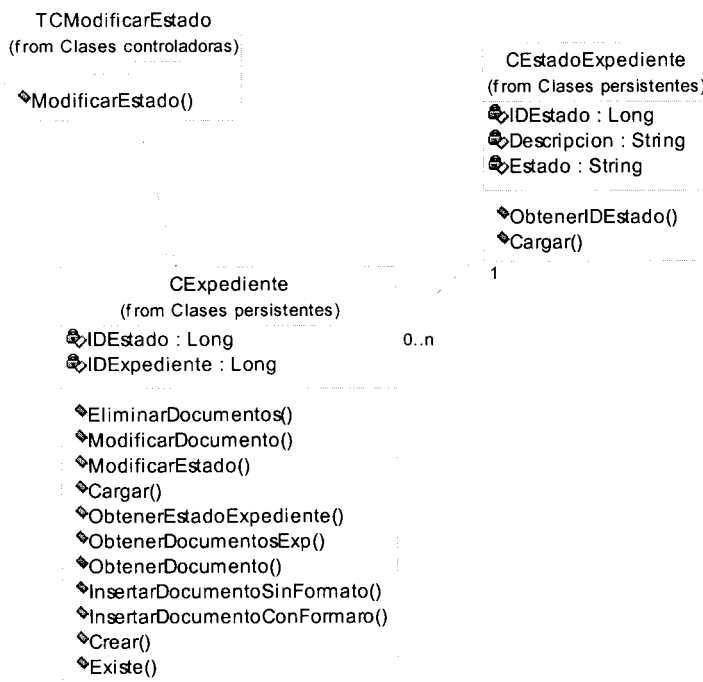


Diagrama de clases del caso de uso "Modificar estado".

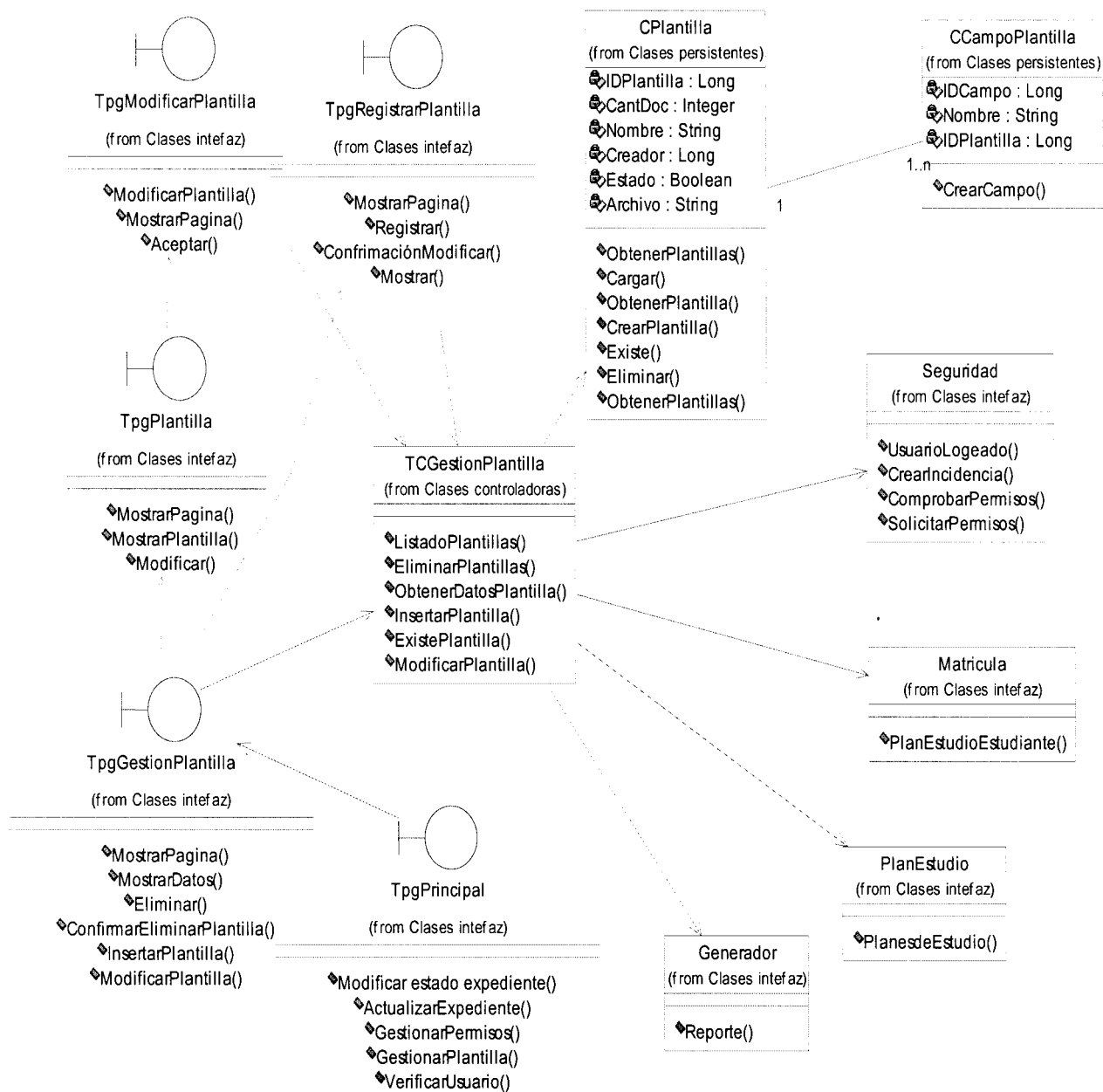


Diagrama de clases del caso de uso "Gestionar plantillas".

Anexo #4: Descripción de las clases del análisis y diseño.

Nombre: CExpediente	
Tipo de clase: Entidad.	
Atributo	Tipo
IdEstado	Int
IdExpediente	Int
Estado	CEstado
IdEstudiante	Int
ListaDocumentos	CDocumentos []
Responsabilidades	
Nombre:	EliminarDocumentos(IDDocumentos)
Descripción:	Su función es eliminar los documentos pasados por parámetros.
Nombre:	ModificarDocumento(Documento, IdDocumento)
Descripción:	Modifica los datos del documento especificado por el segundo parámetro.
Nombre:	ModificarEstado(NuevoEstado)
Descripción	Modifica el estado del expediente
Nombre:	Cargar(IdExpediente)
Descripción	Inicializa los atributos de la clase.
Nombre:	ObtenerEstadoExpediente
Descripción	Devuelve el atributo estado, que seria el estado en que se encuentra el expediente.
Nombre:	ObtenerDocumentosExp
Descripción	Devuelve todos los documentos asociados al expediente.
Nombre:	ObtenerDocumento(IdDocumento)
Descripción	Devuelve un documento específico.
Nombre:	InsertarDocumentoConFormaro(NuevoDoc, IDPlantilla, IUsuario, Nombre)
Descripción	Inserta un nuevo documento con formato al expediente.

Nombre:	InsertarDocumentoSinFormato(Documento,Nombre, IDUsuario)
Descripción	Inserta un nuevo documento sin formato al expediente.
Nombre:	Crear(IdEstudiante)
Descripción	Crea un expediente en estado de pre matricula para un estudiante.
Nombre:	Existe(IdExpediente)
Descripción	Determina si existe el expediente pasado por parámetro.

Nombre: CDocumento	
Tipo de clase : Entidad	
Atributo	Tipo
IdDocumento	Int
Nombre	String
TipoDocumento	int
IdExpediente	Int
Responsabilidades.	
Nombre:	Eliminar(IdDocumento)
Descripción:	Elimina el documento.
Nombre:	ObtenerDatos
Descripción:	Devuelve los valores de los atributos.
Nombre:	CrearDocumento(IDExpediente, IDUsuario, Nombre, Tipo)
Descripción:	Crea un documento.
Nombre:	ObtenerTipoDocumento(IdDocumento)
Descripción:	Devuelve el tipo del documento.

Nombre: CDocumentoSinFormato	
Tipo de clase : Entidad	
Atributo	Tipo
IdDocumento	Int

Direccion	String
Responsabilidades	
Nombre:	CrearDocumentoSinFormato(Documento, IDUsuario, IDExpediente, Nombre)
Descripción:	Crea un nuevo documento.
Nombre:	ObtenerArchivo()
Descripción:	Devuelve el archivo del documento.

Nombre: CDocumentoConFormato	
Tipo de clase: Entidad	
Atributo	Tipo
IdDocumento	Int
IdPlantilla	Int
Responsabilidades.	
Nombre:	CrearDocConFormato(IDExpediente, IDUsuario, Documento, IDPlantilla, Nombre)
Descripción:	Crea un documento con formato.
Nombre:	ObtenerDocumentos(IDExpediente)
Descripción:	Devuelve los documento asociados al expediente.
Nombre:	Modificar(NuevoDocumento)
Descripción:	Modifica los datos del documento.
Nombre:	ObtenerIdPlantilla()
Descripción:	Devuelve el identificador de la plantilla.
Nombre	Documento()
Descripción	Devuelve el archivo que fue creado a través de la plantilla.

Nombre: CPlantilla	
Tipo de clase : Entidad	
Atributo	Tipo
IdPlantilla	Int

Direccion	String
Estado	Int
CantidadDoc	Int
Nombre	String
ListaCampos	CCamposPlantilla[]
Responsabilidades	
Nombre:	ObtenerPlantillas()
Descripción:	Devuelve todas las plantillas activas.
Nombre:	Cargar
Descripción:	Inicializa los atributos.
Nombre:	ObtenerPlantilla(IdPlantilla)
Descripción:	Devuelve el archivo confeccionado a través del InfoPath.
Nombre:	CrearPlantilla(Nombre, PlanEstudio, Dirección, IdCreador)
Descripción:	Crea una plantilla.
Nombre:	Existe(Nombre, Plan)
Descripción:	Devuelve verdadero si existe una plantilla con ese nombre para ese plan de estudio.
Nombre:	Eliminar
Descripción:	Elimina la plantilla.
Nombre	ObtenerPlantillas(PlanEstudio)
Descripción	Devuelve las plantillas pertenecientes a ese plan de estudio.

Nombre: CCampoPlantilla	
Tipo de clase : Entidad	
Atributo	Tipo
IdPlantilla	Int
IdCampo	Int
Nombre	String
Tipo	String

Responsabilidades.	
Nombre:	CrearCampo(IdPlantilla, Nombre, Tipo)
Descripción:	Crea un campo perteneciente a una plantilla.

Nombre: CEstadoExpediente	
Tipo de clase :Entidad	
Atributo	Tipo
IdEstado	Int
Estado	String
Descripción	String
Responsabilidad:	
Nombre:	ObtenerIdEstado()
Descripción:	Devuelve el Identificador del estado.
Nombre	Cargar(IdEstado)
Descripción	Inicializa los atributos.

Nombre: TCAActualizar	
Tipo de clase: Controladora	
Atributo	Tipo
Expediente.	CExpediente.
Responsabilidades.	
Nombre:	VerificarActualizacion(IdExpediente)
Descripción	Esta operación permite saber si el expediente está en un estado en el que se pueda actualizar, se puede actualizar el expediente si y solo si se encuentra en el estado abierto.
Nombre	ObtenerListadoDocumentos(IdExpediente)
Descripción	Dado un expediente obtiene el listado de los documentos asociados al mismo.
Nombre	ObtenerListadoPlantilla(IdEstudiante)

Descripción	Obtiene un listado con las plantillas por las cuales se le puede asociar un documento al expediente del estudiante, estas plantillas deben pertenecer la plan de estudio a que está matriculado el estudiante..
Nombre	EliminarDocumentos(IdDocumentos)
Descripción	Debe eliminar los documentos pasados por parámetros.
Nombre	InsertarDocumentoSinFormato(Direccion, Nombre, IdExpediente).
Descripción	Inserta un documento sin formato.
Nombre	InsertarDocumentoConFormato(Datos,Nombre, IdExpediente, IdPlantilla)
Descripción	Inserta un documento con formato.
Nombre	CargarDocumento(IdDocumento)
Descripción	Obtiene un documento asociado a su plantilla.
Nombre	ModificarDocumento(DatosModicados)
Descripción	Modifica los datos de un documento con formato.

Nombre: TCCrearExpediente	
Tipo de clase : Controladora.	
Atributo	Tipo
Expediente	CExpediente
Responsabilidades	
Nombre:	CrearExpediente(IdEstudiante)
Descripción:	Crea un expediente para el estudiante pasado por parámetro.

Nombre: TCGestionPlantilla	
Tipo de clase : Controladora	
Atributo	Tipo
Plantilla	CPlantilla

Responsabilidades	
Nombre:	ListadoPlantillas()
Descripción:	Devuelve un listado de las plantillas activas.
Nombre:	EliminarPlantillas(IdPlantillas)
Descripción:	Elimina las plantillas pasadas por parámetros.
Nombre:	ModificarPlantilla(IdPlantilla, DatosNuevos)
Descripción:	Modifica una plantilla con los nuevos datos pasados por parámetros, para esto crea una nueva plantilla con los datos específicos de la que se quiere modificar y se activa la nueva.
Nombre:	ExistePlantilla(Nombre, Plan)
Descripción:	Verifica si existe una plantilla para ese plan de estudio con ese nombre.
Nombre:	InsertarPlantilla(Dirección, IdPlantilla, Nombre, Plan)
Descripción:	Crea una nueva plantilla.

Nombre: TCModificarEstado	
Tipo de clase : Controladora	
Atributo	Tipo
Expediente	CExpediente.
Responsabilidades	
Nombre:	ModificarEstado(IdExpediente, IdEstado)
Descripción:	Modifica el estado de un expediente, ver transacción de estado en el anexo4.

Nombre: TpgActualizarExpediente	
Tipo de clase : Interfaz	
Atributo	Tipo
CControlAtualizar	TCActualizar
Responsabilidades.	

Nombre:	MostrarPagina(IdEstudiante)
Descripción:	Si se puede actualizar el expediente, se obtienen los documentos que tiene asociado el expediente, y las plantillas por las que aún se le puede asignar un documentos de ese tipo al expediente, se muestran los datos.
Nombre	Insertar(IDPlantilla)
Descripción	Se muestra la página TpgInsertarDocumento.
Nombre	ModificarDocumento(IDDocumento).
Descripción	Muestra la página TpgModificarDocumento.
Nombre	InsertarDocumentoSinFormato().
Descripción	Muestra la página TpgDocumentoSFormato.
Nombre	EliminarDocumento(Documentos).
Descripción	Se verifica que el usuario quiera eliminar realmente los documentos seleccionados, se llama al método eliminar de la clase TCActualizar.

Nombre: TpgDocumentoSFormato	
Tipo de clase : Interfaz	
Atributo	Tipo
Actualizar	TCActualizar
Responsabilidades.	
Nombre:	Insertar(Nombre, Direccion)
Descripción:	Llama al método InsertarDocumentoSinFormato de la clase TCActualizar con sus previos parámetros.
Nombre	MostrarPagina(IdExpediente)
Descripción:	Muestra la página con los datos requeridos.

Nombre: TpgGestionPlantilla.	
Tipo de clase : Interfaz.	
Atributo	Tipo
CControlPlantilla	TCGestionPlantilla
Responsabilidades	
Nombre:	MostrarPagina()
Descripción:	Obtiene un listado de las plantillas y muestra sus nombres.
Nombre:	Eliminar(IdPlantillas)
Descripción:	Verifica que el usuario quiera realmente el eliminar los documentos seleccionados, llama al método EliminarPlantillas de la clase TCGestionPlantilla.
Nombre:	ModificarPlantilla(IdPlantilla)
Descripción:	Muestra la página TpgPlantilla.
Nombre:	InsertarPlantilla()
Descripción:	Muestra la página TpgRegistrarPlantilla.

Nombre: TpgInsertarDocumento	
Tipo de clase : Interfaz.	
Atributo	Tipo
CControlActualizar	TCActualizar
Responsabilidades	
Nombre:	MostrarPagina(IdPlantilla, IdExpediente)
Descripción:	Muestra la plantilla para que el usuario rellene la misma.
Nombre:	Insertar(Datos, Nombre)
Descripción:	Llama al método InsertarDocumentoConFormato perteneciente a la clase CTActualizar.
Nombre:	
Descripción:	

Nombre: TpgModificarDocumento

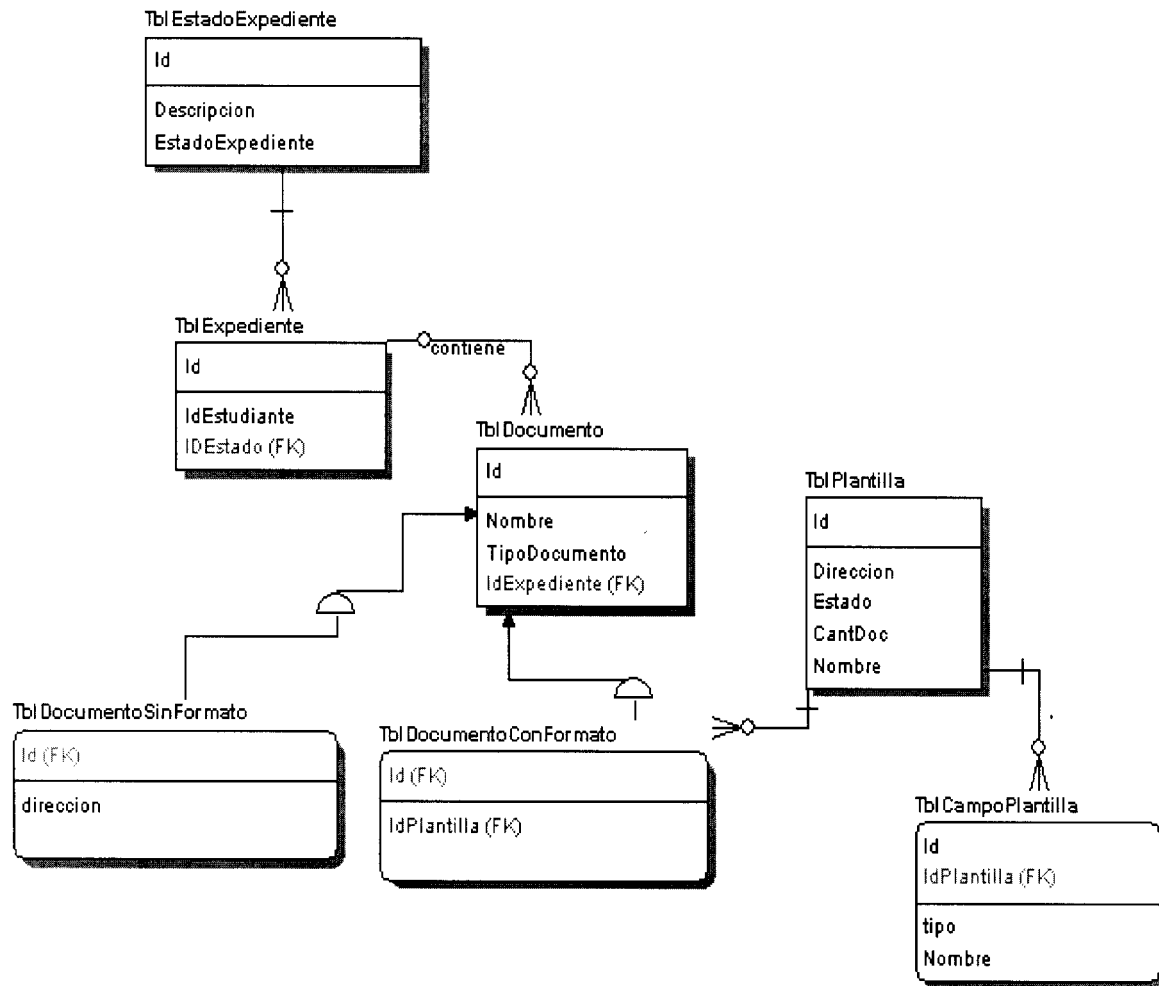
Tipo de clase : Interfaz.	
Atributo	Tipo
CControlActualizar	TCActualizar
Responsabilidades	
Nombre:	MostrarPagina(IdDocumento, IdExpediente)
Descripción:	Muestra un documento para que el usuario lo modifique.
Nombre:	Modificar(DatosM)
Descripción:	Llama al método ModificarDocumento de la clases TCActualizar.

Nombre: TpgPlantilla	
Tipo de clase : Interfaz.	
Atributo	Tipo
CControlPlantilla	TCGestionPlantilla
Responsabilidades	
Nombre:	MostrarPagina(IDPlanilla)
Descripción:	Muestra la plantilla.
Nombre:	Modificar
Descripción:	Muestra la página TpgModificarPlantilla

Nombre: TpgModificarPlantilla	
Tipo de clase : Interfaz.	
Atributo	Tipo
CControlPlantilla	TCGestionPlantilla
Responsabilidades:	
Nombre:	MostrarPagina(idPlantilla)
Descripción:	Muestra la página.
Nombre:	Aceptar(Dirección)
Descripción:	Llama al método ModificarPlantilla de la clase TCGestiónPlantilla.

Nombre: TpgPrincipal	
Tipo de clase : Interfaz.	
Responsabilidades:	
Nombre:	ActualizarExpediente(IDEstudiante)
Descripción:	Verifica si el usuario puede actualizar el expediente, muestra la página TpgActualizarExpediente.
Nombre:	GestionarPlantilla()
Descripción:	Muestra la página TpgGesionPlantilla

Nombre: TpgRegistrarPlantilla	
Tipo de clase : Interfaz.	
Atributo	Tipo
CControlPlantilla	TCGestionPlantilla
Responsabilidades:	
Nombre:	MostrarPagina()
Descripción:	Muestra la página con los datos previos.
Nombre:	Registrar(Nombre, PlanEstudio, Direccion)
Descripción:	Verifica para ese plan de estudio si existe una plantilla con el mismo nombre, en caso afirmativo se le pregunta al usuario si desea modificarlo, se llama al método InsertarPlantilla pertenecientes a la clase TCGestionPlantilla

Anexo #5: Diagrama de Entidad Relación (DER) de la BD.

Anexo #6: Descripción de las tablas.

Nombre: TbExpediente		
Descripción: Esta tabla será la encargada de almacenar los expedientes del estudiante.		
Atributo	Tipo	Descripción
Id	unique	Este atributo es el identificador del expediente.
IEstudiante	Integer	Identificador del expediente al que pertenece el expediente.
IdEstado	Integer	Identificador del estado en que se encuentra el expediente.

Nombre: TblDocumento		
Descripción: Guardará parte de la información de los documentos.		
Atributo	Tipo	Descripción
Id	unique	Identificador del documento.
IdExpediente	Integer	Identificador del expediente al que pertenece
Nombre	nvarchar	Nombre del documento.

Nombre: TblDocumentoConFormato

Descripción: Almacena los datos específicos de los documentos con formatos.		
Atributo	Tipo	Descripción
Id	Integer	Identificador del documento.
IdPlantilla	Integer	Identificador de la plantilla por el que fue confeccionado.

Nombre: TblDocumentoSinFormato		
Descripción: Se almacenan los datos específicos de los documentos sin formatos.		
Atributo	Tipo	Descripción
Id	Integer	Identificador del documento.
Dirección	nvarchar	Dirección donde se encuentra el documento

Nombre: TblPlantilla		
Descripción: Es la encargada de almacenar a los datos de las plantillas.		
Atributo	Tipo	Descripción
Id	unique	Identificador de la plantilla
Nombre	nvarchar	Nombre de la plantilla
Estado	Integer	Estado en que se encuentra la plantilla.

		1 activa, 0 desactivada.
Direccion	nvarchar	Dirección donde se encuentra la plantilla.
CantDoc	Integer	Cantidad de documentos que puede tener asociado un expediente mediante esta plantilla.

Nombre: TblCampoPlantilla		
Descripción: Almacena los campos de cada plantilla.		
Atributo	Tipo	Descripción
Id	Unique	Identificador del campo.
IdPlantilla	Integer	Identificador de la plantilla de la cual el forma un campo.
Nombre	nvarchar	Nombre del campo.
Tipo	nvarchar	Tipo del campo.

Nombre: TblEstadoPlantilla		
Descripción: En esta tabla se guardan los estados en los que puede encontrarse el expediente.		
Atributo	Tipo	Descripción
Id	Integer	Identificador del estado.

EstadoExpediente	nvarchar	Estado en que se puede encontrar un expediente.
Descripción	nvarchar	Un abreve descripción del estado.

Anexo #7: Transacciones del estado del expediente.

