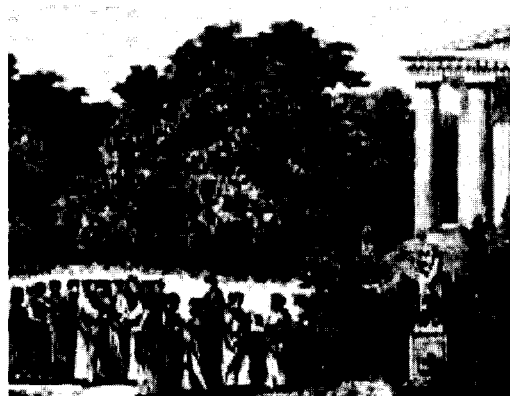


003.7
DOR
S
TD 0033-04-01

TD-0033-04-01

UNIVERSIDAD DE LA HABANA
FACULTAD DE MATEMÁTICA COMPUTACIÓN
CARRERA CIENCIA DE LA COMPUTACIÓN

SISTEMA AUTOMATIZADO PARA LA GESTIÓN ACADÉMICA. SUBSISTEMA “GENERADOR DE REPORTES”



**AUTOR: ARISMAYDA DORADO RISCO
TUTOR: ING. EMIL LIMA VALDÉS**

Dedicatoria

Por la ignorancia nos equivocamos,
y por las equivocaciones aprendemos.

Proverbio Romano

A mis abuelos
A mis tíos
A mis hermanos
A mi papito dondequiera que estés
A mi familia toda

Agradecimientos

Agradecimientos

A mis abuelos por dedicarme su vida, por tantos años de empeño y esfuerzo, por hacerme lo que soy.

A mis tíos y hermanos por ayudarme a realizar este sueño y estar presentes en todo momento, y con su sacrificio permanente contribuir a la culminación de mis estudios.

A mi papá por quererme, aunque ya no este conmigo.

A toda mi familia.

A Anniet, mi hermanita negra por ser mi amiga.

A mi tutor Emil Lima por toda la paciencia y el esfuerzo dedicado.

A Karel Verdecia, Nasario, Enrique del Valle, David Silva, Rolando, Alejandro, Alfredo Mckensi, Jose Celeiro, Ivan, Liuvyn, Frank León, Jhonlier, Pedro, Andión y tantos otros que no me alcanzarían todas las hojas para mencionarlos, muchas gracias por haber dedicado parte de su tiempo a ayudarme cuando presenté dificultades.

A Billy, Yaimé, en fin a todos mis compañeros, por siempre estar presentes en estos cinco años compartiendo no solo alegrías, también ratos amargos.

A Damila, Glenda, Frank, Idalmis, Orlenis, Arnaldo, Ramos, Jorge Arias, Juan Carlos, Leo, Ivis, Lisandra, Ariadna por ayudarme y hacerme sentir especial, por ser mis amigos.

A Gabriel, Lucho, Amilkar, Aldo, Yadier, Yeimí, Chucho, Sandy, Puebla, Yoemni por haber sido mas que compañeros y por haberme hecho sufrir tanto frío.

A mis estudiantes por comprender que necesitaban ser más independientes.

A Carlos, Arnaldo, Lurdes y Daniel porque juntos entendimos la frase trabajo en equipo, aunque nunca la desarrollamos.

A todos aquellos que no he mencionado, pero que siempre tendrán un pedacito de mi corazón.

A todos, los quiero.

Resumen

Debido al entorno dinámico de los negocios, no es posible anticipar todas las necesidades del mañana. En este sentido se desarrolla el "Generador de Reportes", un sistema diseñado para que el mismo usuario pueda realizar adaptaciones en forma rápida y sencilla sin modificar el código fuente, dándole la posibilidad de crear los reportes que necesite, haciendo más flexible el sistema a los cambios.

Se trata de una herramienta que guía paso a paso al usuario en las operaciones más habituales, supliendo así su posible falta de experiencia y conocimientos. También se comporta como un proveedor de servicios para el resto de los subsistemas, brindando y almacenando los informes que estos requieran, permitiéndose con esto un gran ahorro de trabajo para el proyecto completo ya que los desarrolladores no tienen que preocuparse por la elaboración y almacenamiento de los reportes.

Para su desarrollo se siguieron los pasos que propone el Proceso Unificado. Está implementado sobre la plataforma Microsoft .NET, específicamente en el lenguaje C#.NET, con el cual se obtuvo muy buenos resultados.

Índice

Introducción	1
Problema.....	2
Objetivo general y Objetivos específicos.	2
Objeto de Estudio.....	3
Campo de Acción.....	3
Tareas de investigación.	3
Estructuración del trabajo.	4
Capítulo 1. Fundamentación Teórica	5
1.1 Estado del Arte.....	5
1.1.1 Ámbito Internacional.....	5
1.1.2 Estudios en el país.	8
1.2 Herramientas utilizadas para el desarrollo del sistema.	9
1.2.1 La tecnología .NET.....	10
1.2.1.1 Arquitectura Framework.NET.....	10
1.2.2 Lenguajes de programación.	11
1.2.3 ASP.NET.	12
1.2.4 Modelación de aplicaciones Web sobre la plataforma ASP.NET.	15
1.2.5 Programación Multicapas.	17
1.2.6 Servicios Web (Web Services).	18
1.2.6.1 Protocolos que usan los Servicios Web.....	19
1.3 Sistemas Gestores de Base de Datos (SGDB).....	20
1.3.1. Catálogos de sistema de bases de datos.	20
1.3.2. Estructuras de almacenamiento físico y lógico.....	22
1.3.3. Datos de bandas.	23
1.3.4. Registros de transacciones y recuperación automática.....	24
1.3.5. Redes.	25
1.4 RUP.	27
1.4.1 Características del Proceso Unificado.....	28

1.4.2 Rational Rose.....	29
1.5 Conclusiones.....	30
Capítulo 2. Características del Sistema.....	31
2.1 Objeto de estudio.....	31
2.1.1 Situación problemática.....	31
2.1.2 Ubicación.....	31
2.2 Objeto de Automatización.....	32
2.3 Propuesta de Sistema.....	33
2.3.1 Análisis comparativo con otras soluciones existentes con la propuesta... 34	
2.4 Especificación de los requisitos de software.....	35
2.4.1 Dependencias y Relaciones.....	35
2.4.2 Requerimientos funcionales.....	35
2.4.3 Requerimientos no funcionales.....	38
2.5 Definición de los Casos de Uso.....	40
2.5.1 Identificación de los actores.....	40
2.5.2 Listado de casos de uso.....	40
2.5.3 Diagrama de casos de uso.....	43
2.5.4 Casos de uso por ciclos.....	43
2.5.4.1 Casos de uso del núcleo central.....	43
2.5.5 Especificación de casos de uso.....	44
Capítulo 3. Análisis y diseño del sistema.....	45
3.1 Análisis.....	45
3.1.1 Modelo de clases de análisis.....	46
3.2 Diseño.....	46
3.2.1 Diagramas de Secuencia.....	46
3.2.2 Diagramas de Clase.....	47
3.2.3 Descripción de las clases.....	48
3.3 Diseño de BD.....	52
3.3.1 Modelo Relacional.....	52
3.3.2 Descripción de las tablas.....	53
3.4 Conclusiones.....	55
Conclusiones	56

Recomendaciones	58
Referencias Bibliográficas.....	59
Bibliografía.....	60
Glosario de Términos.....	62
Anexos.....	66
Anexo 1. Casos de Uso expandidos	66
Anexo 2. Diagrama de Clases del análisis.....	79
Anexo 3. Diagrama de Clases del diseño.....	77
Anexo 4. Diagramas de Secuencia.....	88

Introducción

En toda empresa siempre ha existido la necesidad de documentar el resultado de los procesos del negocio. Actualmente hay mucha documentación que se genera día a día en diferentes formatos, tales como informes de mercado, listas de productos, albaranes, pedidos o facturas, que no sólo exige que contenga la información que se requiere en un momento dado, sino también que aparezca con un formato coherente y que haga el informe fácil de leer y estructure la información que representa de manera ordenada y lógica.

La mayoría de los sistemas informáticos actuales proveen salidas más o menos avanzadas de la información a través de informes debido a que, aunque el software creado para alguna empresa específica brinde una serie de información predefinida, pudiera ocurrir que no se muestren datos específicos, en un formato deseado. Por eso se crea un asistente de Reportes que facilita al usuario construir sus propias consultas para generar reportes o exportar los datos a otros formatos como Microsoft Excel donde la información puede ser formateada a su gusto.

Los Generadores de Informes surgieron por la necesidad de hacer reportes fáciles y rápidos que se adaptaran 100% a las necesidades del usuario, sin tener que depender de un nuevo desarrollo. También son de gran ayuda para los departamentos de elaboración de software por el ahorro de tiempo tanto en desarrollo como mantenimiento. Estos son programas para crear informes sobre diseño en una amplia variedad de formatos que no son rutinariamente producidos por un sistema de información. Extraen datos de los archivos o de las bases de datos y crean reportes de acuerdo con muchos formatos, proporcionan más control, pueden manejar datos de cálculos y lógica compleja antes de darles la salida. Una herramienta presente en ellos son los asistentes de informes (Wizard), se tratan de herramientas que guían paso a paso al usuario en las operaciones más habituales, supliendo así su posible falta de experiencia y conocimientos. Cada vez cobran más

importancia como una evolución natural de la ayuda para el usuario orientada a tareas, es por esto que los productos intentan integrar más asistentes.

Tradicionalmente los reportes han sido el medio principal para tener información en las manos. Son la manera más rápida de volcar los datos en papel clasificados por alguna información significativa.

Una limitante que siempre ha existido es que el generar un reporte implica manejar algunas habilidades técnicas relacionadas con las bases de datos y las herramientas de software. Los reportes o las consultas son requeridos por cualquier persona dentro de la organización.

Todo lo anteriormente expuesto me ha llevado a plantear el problema cuya solución facilita la obtención de información.

Problema.

¿Cómo disminuir los riesgos del sistema de gestión académica asociados a la aparición de nuevas necesidades de generación de reportes?

Objetivo general y Objetivos específicos.

Desarrollar un sistema automatizado que permita a los usuarios crear de forma inmediata, reportes asociados a la información manejada por el sistema de gestión académica; así como unificar su generación para el resto de los subsistemas que lo necesiten.

A partir de un análisis del objetivo general se derivan los siguientes objetivos específicos para la aplicación:

- Permitir a los usuarios del sistema crear y mostrar sus propios reportes de una manera ágil y rápida (reportes columnares con definición de campos, títulos, formatos y cálculo de totales).

- Desarrollar un proveedor de servicios de reportes para los subsistemas de dicho sistema.
- Ser capaz de proveer de reportes con contenido completamente configurables a los demás subsistemas del Sistema Académico.
- Permitir que los reportes se consulten en pantalla, se envíen a una impresora determinada o se almacenen como un archivo de disco.

Objeto de Estudio.

Se estudiaron dos asistentes de informes, el del Access y Cristal report.Net, un generador que trae incluida la plataforma .NET. De los estudios realizados se determinó que el asistente de informe del Access tiene un mejor destino para un usuario del cual desconocemos su nivel de conocimiento acerca de la base de datos, y como además ofrece una interfaz más amigable, escogimos este asistente como objeto de estudio.

Campo de Acción.

Generadores de Reportes.

Tareas de investigación.

- Búsqueda en Internet sobre el tema y viables propuestas a desarrollar.
- Adiestrarse en el formato de trabajo de los asistentes de informes de las herramientas de Microsoft Access y Cristal Report.net.
- Análisis y diseño del subsistema “Generador de Reportes” del sistema “Gestión Académica”.
- Implementación del subsistema “Generador de Reportes”.

- Redacción del documento para la continuidad de su puesta a punto.

Estructuración del trabajo.

El presente documento consta de una introducción, tres capítulos y varios anexos, que incluye todo lo relacionado con el trabajo investigativo; así como el análisis y el diseño de la herramienta que se propone.

El capítulo 1 recoge el análisis de la información existente acerca del tema a tratar y las tendencias actuales que existen en el mundo. También incluye una descripción del lenguaje de programación a utilizar tanto para la modelación como la implementación.

El capítulo 2 describe aspectos del sistema como: el objeto de estudio, el entorno de trabajo en que se desarrolla, la especificación de los detalles de la construcción de la herramienta, una propuesta de solución, requerimientos funcionales y no funcionales y los casos de uso.

El capítulo 3 muestra la expansión de los casos de uso, el modelo conceptual, los diagramas de secuencia del sistema. Se plantean detalles relacionados con el diseño que se propone, los diagramas de interacción, el modelo de clases y la base de datos.

Capítulo 1. Fundamentación Teórica

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

Han sido implementados muchos sistemas de gestión académica, que de una forma u otra responden a las exigencias, cada día más crecientes, de responsabilidad, calidad y rapidez en todo proceso docente; de estructuras muy variadas donde cada uno le da el enfoque de la actividad que necesita informatizar. Se han aplicado fundamentalmente las tecnologías Web, a pesar de que con la puesta en marcha de la telefonía celular como medio de transmisión de datos han aparecido vías para realizar la matrícula sin necesidad de acceder a un sitio Web desde una PC, sino que ahora se puede hacer incluso desde su celular privado.

1.1 Estado del Arte.

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

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

1.1.1 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 admite consultar el historial académico del estudiante en un determinado plan de estudio, modificar o completar sus datos personales y da acceso a cualquier información relacionada con su presencia en la universidad. Incluye la solicitud de convalidaciones, adaptaciones o reconocimiento de créditos, el registro de traslado de expediente, la emisión de todo tipo de certificado y listado relacionado. La matrícula puede realizarse por la secretaría o por el sistema a través de la 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.

1.1.2 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.

1.2 Herramientas utilizadas para el desarrollo del sistema.

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

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

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

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

1.2.1 La tecnología .NET.

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

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

1.2.1.1 Arquitectura Framework.NET.

La arquitectura .NET (.NET Framework) es el modelo de programación de la plataforma .NET para construir y ejecutar los servicios .NET. El objetivo de esta arquitectura es la de reducir la complejidad en el desarrollo de este tipo de aplicaciones, permitiendo a los desarrolladores centrarse en escribir la lógica específica del servicio a desarrollar.

1.2.2 Lenguajes de programación.

Lenguaje C#

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

Las principales características que identifican al lenguaje C# son:

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

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

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

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

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

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

1.2.3 ASP.NET. [IASP]

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.

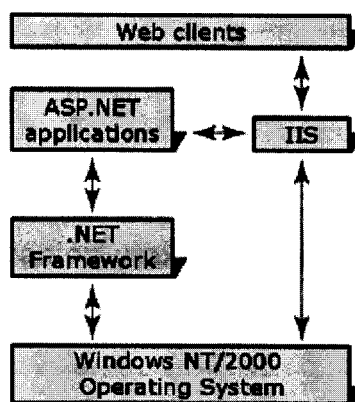


Fig. 1 Arquitectura ASP.NET.

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 [1]:

- 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.

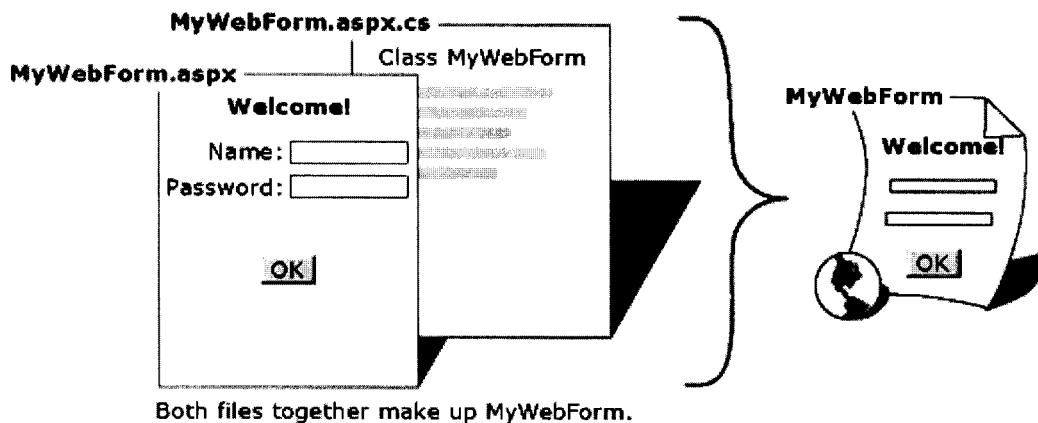


Fig. 2 WebForms.

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.

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

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

a los diagramas de una nueva semántica propia del problema que se está modelando.

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

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

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

- Eliminar la programación en el cliente. Aunque esto supone un deterioro del desempeño de la aplicación, elimina la necesidad de escribir código específico para cada uno de los navegadores más utilizados.
- Utilización intensiva de controles en el servidor. Esta medida apoya la anterior y permite dejarle a estos la generación del código específico para cada navegador.

- No utilización de elementos ajenos a la plataforma como applets java y componentes ActiveX en el cliente.
- No utilización de marcos.

1.2.5 Programación Multicapas.

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

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

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

Presentación.

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

Reglas del Negocio.

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

Acceso a Datos.

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



Figura 3. Aplicaciones de tres capas

Ventajas del modelo.

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

1.2.6 Servicios Web (Web Services).

Los Web Services son componentes software que permiten a los usuarios usar aplicaciones de negocio que comparten datos con otros programas modulares, vía Internet. Son aplicaciones independientes de la plataforma que pueden ser fácilmente publicadas, localizadas e invocadas mediante protocolos Web estándar,

como XML, SOAP, UDDI o WSDL. El objetivo final es la creación de un directorio de online de Web Services, que pueda ser localizado de un modo sencillo y que tenga una alta fiabilidad.

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

1.2.6.1 Protocolos que usan los Servicios Web.

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

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

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

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

1.3 Sistemas Gestores de Base de Datos (SGDB).

Un Sistema de Gestión de Bases de Datos (SGBD) Consiste en un conjunto de programas, procedimientos y lenguajes que nos proporcionan las herramientas necesarias para trabajar con una base de datos. Incorporar una serie de funciones que nos permita definir los registros, sus campos, sus relaciones, insertar, suprimir, modificar y consultar los datos [3]. 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 [4].

1.3.1. Catálogos de sistema de bases de datos.

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

- Objetos de la base de datos (tablas, índices, procedimientos almacenados, vistas, desencadenadores, etc.).
- Restricciones.

- Usuarios y permisos.
- Tipos de datos definidos por el usuario.
- Definiciones de réplicas.
- Archivos utilizados por la base de datos.

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

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

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

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

1.3.2. Estructuras de almacenamiento físico y lógico.

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

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

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

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

SQL Server da formato a los archivos en unidades internas denominadas *páginas*. El tamaño de cada página es fijo: 8.192 bytes (8 KB). Las páginas están organizadas en

extensiones fijas de ocho páginas contiguas. Al crear una tabla o un índice en una base de datos de SQL Server, se asigna automáticamente a una página de una extensión. A medida que se amplía una tabla o un índice, SQL Server le asigna automáticamente su propia extensión. De este modo, el almacenamiento de tablas pequeñas e índices es más efectivo que si se asignara la extensión completa, como sucede en Oracle.

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

1.3.3. Datos de bandas.

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

La configuración de una matriz RAID recomendada para SQL Server es RAID 1 (espejo) o RAID 5 (conjuntos de bandas con unidad de paridad adicional, para redundancia). Se recomienda también la opción RAID 10 (reflejo de conjuntos de bandas con paridad), aunque resulta mucho más cara que las dos anteriores. Los conjuntos de bandas son apropiados para distribuir la E/S aleatoria realizada normalmente en los archivos de base de datos.

Los archivos de registro de transacciones deben optimizarse para la entrada y salida secuencial, y asegurarse sobre un punto de error único. Por lo tanto, se recomienda la opción RAID 1 (reflejo) para los registros de transacciones. Al realizar la migración, el tamaño de esta unidad debe ser, al menos, igual a la suma del tamaño de los

registros de recuperación en conexión de Oracle más las tablas de segmentos de inversión de Oracle. Cree uno o varios archivos de registro que utilicen todo el espacio reservado para la unidad lógica. A diferencia de los datos almacenados en grupos de archivos, las entradas de los registros de transacción se guardan siempre en forma secuencial y no se rellenan proporcionalmente.

1.3.4. Registros de transacciones y recuperación automática.

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

Microsoft SQL Server 2000 también efectúa una recuperación de datos automática mediante la comprobación de todas las bases de datos del sistema siempre que se inicia. En primer lugar, comprueba la base de datos **master** y, a continuación, lanza los subprocesos para recuperar el resto de las bases de datos del sistema. El mecanismo de recuperación automática comprueba el archivo de transacciones de cada base de datos de SQL Server. Si este registro contiene transacciones sin confirmar, éstas se deshacen. El mecanismo de recuperación comprueba seguidamente en el registro de transacción las transacciones sin confirmar que aún no se hayan escrito en la base de datos. Si encuentra alguna, ejecuta de nuevo las transacciones y continúa.

Cada registro de transacciones de SQL Server tiene la funcionalidad combinada de un segmento de inversión de Oracle y uno de recuperación en conexión. Cada base de datos cuenta con un registro de transacciones propio que registra todos los cambios realizados en ella y es compartido por todos sus usuarios. Cuando se inicia una transacción que produce una modificación de los datos, se guarda en el registro

un suceso BEGIN TRANSACTION, además del suceso de modificación. Este suceso se utilizará durante la recuperación automática para determinar el punto de inicio de la transacción. Según se reciben las instrucciones de modificación de los datos, se van escribiendo los cambios en el registro de transacciones, antes de hacerlo en la propia base de datos. SQL Server dispone de un mecanismo de control automático que garantiza que las transacciones completas se escriban regularmente de la caché de disco del servidor SQL Server al archivo de registro de transacciones. Un punto de control guarda en la base de datos cada página almacenada en la caché que ha sido modificada desde el último punto de control. Al establecer puntos de control en estas páginas mantenidas en la caché, también conocidas como páginas desfasadas, se asegura que todas las transacciones completas queden grabadas en el disco. Este proceso reduce el tiempo de recuperación necesario si se produjera un error en el sistema, como un corte del suministro eléctrico. Para cambiar este valor, modifique la configuración del intervalo de recuperación mediante el Administrador corporativo de SQL Server o el procedimiento almacenado del sistema **sp_configure** de Transact-SQL.

1.3.5. Redes.

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

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

Específicamente del SQL Server 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.

1.4 RUP.

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

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

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

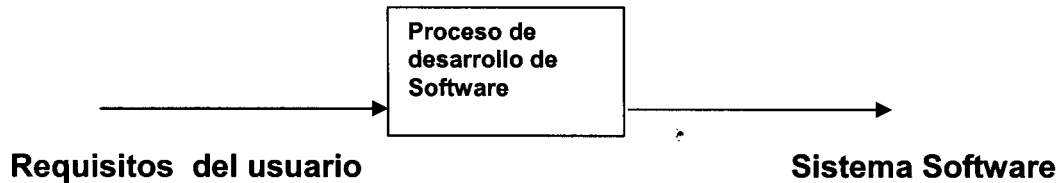


Figura 4. Un proceso de desarrollo de software.

1.4.1 Características del Proceso Unificado.

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

Dirigido por los casos de uso:

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

Estar centrado en la arquitectura:

La arquitectura es una vista del diseño completo con las características más importantes, dejando a un lado los detalles. Esta no solo incluye las necesidades de los usuarios e inversores, sino también otros aspectos técnicos como el hardware, sistema operativo, sistema de gestión de base de datos, protocolos de red; con los que debe coexistir el sistema. En otras palabras, la arquitectura representa la forma del sistema, la cual va madurando en su interacción con los casos de uso hasta llegar a un equilibrio entre funcionalidad y características técnicas.

Ser iterativo e incremental:

La alta complejidad de los sistemas actuales hace que sea factible dividir el proceso de desarrollo en varios mini-proyectos. Cada uno de estos mini-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.

1.4.2 Rational Rose.

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

1.5 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,[5] de ahí que uticemos 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

2.1 Objeto de estudio.

2.1.1 Situación problemática.

La Universidad de las Ciencias Informáticas (UCI) como proyecto estratégico de la nación, hace énfasis en la inteligencia y la creatividad de estudiantes capaces de producir software que sirvan de ingreso a la economía del país. Esto trae consigo, dinámicas transformaciones en su estructura interna.

Para controlar toda la parte académica de la universidad (UCI) se pensó en el desarrollo del “Sistema de Gestión Académica”, el cual es el encargado de gestionar y brindar toda la información necesaria para el cumplimiento de este proceso.

Debido al cambiante mundo de los negocios, si cada módulo de este sistema es el responsable de la notificación de su información, esta puede en un futuro no ser suficiente; por lo que los requerimientos de los usuarios no podrían satisfacerse y surgiría la necesidad de elaborar nuevos reportes accediendo directamente al código fuente de la aplicación, problema que arrastra el sistema que lleva la gestión académica hoy en la universidad.

2.1.2 Ubicación.

La UCI constituye la universidad más joven del país, creada para formar un personal altamente calificado en ingeniería informática, que contribuya en gran medida al desarrollo en ese campo y de la economía del país como se menciono anteriormente.

En estos momentos se está desarrollando en ella el Proyecto UCI-Ciudad digital, que persigue la creación de una micro sociedad completamente informatizada, que pudiera constituir el modelo de sociedad cubana del futuro.

Se han definido diferentes áreas de desarrollo en el proyecto, dentro de las que se encuentra la de Informatización. En ella se ha comenzado la tarea de automatizar todos los procesos de la universidad, dentro de los cuales se encuentra la DIP(Dirección de Informatización) de Gestión Académica, que como bien su nombre lo indica está enfocada al desarrollo de la automatización de la parte académica de la universidad.

2.2 Objeto de Automatización.

El objeto principal a automatizar para evitar que la aplicación caduque es una herramienta flexible que facilite la obtención de información, así como que permita la creación de reportes personalizados y el almacenamiento de estos con mínimos conocimientos de programación por parte del usuario final.

Las ventajas más significativas que ofrece este sistema son:

- Conseguir la máxima autonomía técnica posible para diseñar informes complejos y analizar la información disponible.
- Reducir los costes de elaboración de informes para poder concentrar los esfuerzos en las tareas de análisis, control y propuestas de mejora.
- Aumentar la frecuencia y disponibilidad de la información en cantidad y calidad.
- Diseñar informes dinámicos que permitan el análisis desde múltiples perspectivas.

Para establecer una política para la gestión de la información de una forma integradora, global y efectiva es necesario que esta fluya periódicamente, no sólo a los responsables de la empresa, sino a todas las personas que necesitan información para controlar y mejorar su gestión diaria.

GESTACAD, un sistema desarrollado para la gestión académica, es el vigente actualmente en la universidad, más apegado al negocio de las universidades tradicionales. Al interactuar con este sistema se puede notar que existen problemas con la navegabilidad, el ambiente de trabajo resulta en ocasiones restringido e

inflexible aspectos que recrudescen la búsqueda de información, la interfaz requiere ser más agradable, interesante y atractiva a la vista del usuario, no permite almacenar los planes de estudio, los horarios y los registros.

Este software está muy limitado para las necesidades de un centro de estudios de nivel superior como la UCI, por lo que se necesita más que una actualización, el replanteamiento conceptual del sistema como un todo, o sea, dejar atrás algunos estándares concebidos por el sistema y recomenzar con nuevas ideas que nos permitan brindar los servicios exigidos.

2.3 Propuesta de Sistema.

Desarrollar un simple Generador de Reportes de fácil uso para la creación de informes. El principio de diseño del subsistema está orientado hacia la configuración de salidas de la información dentro del sistema, o sea, que este módulo constituye un generador de reportes en pequeña escala.

Ofrece un asistente de informes (Wizard) para los usuarios del Sistema de Gestión Académica (SGA), mediante el cual se generaran nuevos reportes de acuerdo a sus necesidades, permitiendo personalizar su consulta seleccionando los datos, el orden de presentación y los subtotales. Los reportes pueden consultarse en pantalla, ser enviados a una impresora determinada o almacenarse como un archivo de disco. Además, la información siempre estará segura ya que sólo aquellos usuarios que cuenten con los permisos de acceso apropiados, pueden consultarla.

Provee de servicios de reportes a los subsistemas del SGA, el cual permite que los subsistemas envíen instrucciones que caractericen un reporte, para posteriormente elaborarlo y almacenarlo.

Si se tiene un reporte especial (reporte específico) que se imprime con sistematicidad, se pueden almacenar los valores de sus parámetros una sola vez y posteriormente utilizar esa plantilla cada vez que se requiera dicho informe.

Permite la seguridad de la gestión de la información. Solo el creador de un reporte y el administrador del subsistema tendrán todos los permisos y podrán operar con ellos. Si un usuario tiene autorización a modificar o eliminar un reporte, entonces puede visualizarlo.

Las ventajas operativas son:

- Facilitar toda la información necesaria.
- Es un sistema automático eficaz que permite obtener información en un formato estándar.
- Parametrización. Permite con un solo informe enviar a cada usuario los datos que le corresponden.
- Creación más rápida de reportes. Con los informes dinámicos (tablas dinámicas) se crea un reporte fácilmente que equivale a decenas de informes de papel.
- Posibilidad de exportar los reportes.
- Posibilidad de trabajar con plantillas creadas por los desarrolladores para dar formato a sus reportes.

2.3.1 Análisis comparativo con otras soluciones existentes con la propuesta.

Toda aplicación CRM debe contar con un Generador de Informes que permita que el usuario final , cree, modifique y filtre los reportes según sus necesidades.

RTP Wizard ® un asistente customizador de todas nuestras aplicaciones, estándar o a medida, permite la generación, modificación y filtro de reportes personalizados.

Entre las características operativas más importantes encontramos:

- Libre selección del origen del informe.
- Permite Generar informes personalizados.
- Permite múltiple niveles de agrupamiento en el reporte.
- Múltiples opciones de Ordenamiento.
- Selección de Opciones de Resumen o detalle.
- Selección del Layout del Informe.

- Selección del Diseño del Informe.
- Generación de Informes Gráficos.
- Generación de Etiquetas.
- Permite modificar el Diseño del Informe.
- Configuración de Página de Informe en tiempo de ejecución.
- Selección de Impresora Predeterminada.
- Opciones de Filtrado Dinámico de Informes.
- Definición de Tipos de Informes.
- Posibilidad de definir niveles de seguridad por informes.
- Reportes se combinan con MS Excel, MS Word y MS Graph.

Generador de reportes se ha pensado para que incluya todas estas facilidades y además para que se comporte como un proveedor de servicios, brindando los informes que requieran el resto de los subsistemas, permitiéndose con esto un gran ahorro de trabajo para el proyecto completo ya que los desarrolladores no tienen que preocuparse por la elaboración y almacenamiento de los informes.

2.4 Especificación de los requisitos de software.

2.4.1 Dependencias y Relaciones.

Generador de Reportes para su seguridad depende del subsistema de Seguridad, tanto para proteger el acceso a él como para proteger la información almacenada. Tiene relación con todos los subsistemas del Sistema de gestión Académica pues es un proveedor de servicios de informes.

2.4.2 Requerimientos funcionales.

1. Diseñar reportes a partir de un asistente.

1.1. El asistente tiene una serie de opciones y parámetros que son utilizados para definir la forma en que será ordenado y la manera en que serán seleccionados los datos.

1.1.1. Iniciar el asistente de informes.

1.1.2. Mostrar datos.

1.1.2.1. Mostrar todas las tablas existentes en la base de datos.

- 1.1.3. Permitir seleccionar uno, varios o todos los campos de una tabla escogida.
 - 1.1.4. Permitir deseleccionar uno, varios o todos los campos de una tabla escogida.
 - 1.1.5. Permitir seleccionar campos de más de una tabla
 - 1.1.6. Ordenar o agrupar los datos del informe.
 - 1.1.6.1. Adicionar un nivel de agrupamiento.
 - 1.1.6.1.1. Especificar el campo por el que se desea agrupar.
 - 1.1.6.2. Adicionar regla de ordenación.
 - 1.1.6.2.1. Especificar los campos por los cuales se desea ordenar.
 - 1.1.6.2.2. Especificar el sentido de la ordenación de cada campo, esta puede ser:
 - 1.1.6.2.2.1. Ascendente.
 - 1.1.6.2.2.2. Descendente.
 - 1.1.7. Permitir dar formato al informe.
 - 1.1.8. Permitir dar el resumen de los campos de un informe.
 - 1.1.8.1. El usuario debe especificar el resumen que desea desarrollar, este puede ser:
 - Sumatoria.
 - Promedio.
 - Porcentaje.
 - Mínimo.
 - Máximo.
2. Modificar reportes existentes.
 - 2.1. Insertar/Eliminar un campo.
 - 2.2. Insertar/Eliminar niveles de agrupamiento.
 - 2.3. Insertar/Eliminar reglas de ordenamiento.
 - 2.4. Insertar/Eliminar resúmenes de columnas.
 - 2.5. Insertar/Eliminar un nuevo formato.

3. Eliminar reportes existentes.
 - 3.1. Eliminar el código del reporte.
4. Actualizar reportes.
 - 4.1. Actualiza los datos mostrados.
5. Exportar para Excel.
6. Registrar reportes.
 - 6.1. Los reportes se generan a través de consultas SQL elaboradas por los desarrolladores de cualquiera de los módulos del sistema Gestión Académica.
 - 6.1.1. Obtener la consulta SQL, el título, los parámetros y una plantilla (los dos últimos son opcionales).
 - 6.1.2. Si se envía una plantilla parsearla para establecer el formato deseado.
 - 6.1.3. Almacenar la plantilla como un fichero XML en una dirección predefinida.
 - 6.1.4. Almacenar datos del reporte.
7. Generar Reportes
 - 7.1. Generar el reporte con un formato estándar (plantilla) ya determinado para todos los reportes nuevos.
 - 7.1.1. Las plantillas pueden ser predefinidas por los subsistemas o una tabular predefinida en el Generador de Reportes.
 - 7.2. Mostrar una vista previa del reporte generado.
8. Permitir la seguridad de la información.
 - 8.1. Sólo aquellos usuarios que cuenten con los permisos de acceso apropiados, pueden consultar la información. Los datos y reportes siempre estarán disponibles solo para los usuarios que cuenten con los derechos de acceso apropiados.
9. Permitir filtrar la información.
10. Obtener una vista previa o imprimir el informe.
 - 10.1. Obtener una vista previa.
 - 10.2. Imprimir el informe.
 - 10.2.1. Tipos de Visualización:

- 10.2.1.1. Consultados en pantalla.
- 10.2.1.2. Enviados a una impresora determinada.
- 10.2.1.3. Almacenados como archivos de disco.

2.4.3 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.

Apariencia o interfaz externa

Diseño de interfaz sencillo, fácil de entender y manejar para el rápido desenvolvimiento de los usuarios; así como poco contenido de 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.

Usabilidad

El sistema en general se desarrolla con el objetivo de facilitar el trabajo manual realizado hasta el momento y obtener la mayor cantidad de información almacenada.

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.

Rendimiento

El sistema debe garantizar la mayor eficiencia posible en cuanto al tratamiento de la información, de manera que la velocidad de procesamiento de la misma sea la máxima posible. También se debe garantizar la consistencia y disponibilidad de ella

en todo momento, por lo que se requiere además un tiempo de recuperación mínimo.

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.

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 SQL Server, 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.

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.

Ayuda y documentación en línea

Para informar a los usuarios todo lo referente a como trabaja el sistema se incorporará una ayuda que le permitirá orientarse.

2.5 Definición de los Casos de Uso.

Se identifican los actores y los casos de uso para:

- Delimitar el sistema de su entorno.
- Esbozar quién y qué (actores) interactúan con el sistema, y que funcionalidad (casos de uso) se espera del sistema.
- Capturar y definir un glosario de términos comunes esenciales para la creación de descripciones detalladas de las funcionalidades del sistema (es decir de los casos de uso).

2.5.1 Identificación de los actores.

Los actores son el entorno del sistema. No todos los actores representan a personas. Pueden ser actores otros sistemas o hardware externo que interactúe con el sistema. Es un modelo del sistema que contiene actores, casos de uso y sus relaciones.

Actores	Justificación
Usuario	Representa a cualquier usuario humano del sistema.
Subsistema	Representa a cualquier componente software que necesite hacer uso de los servicios del subsistema Generador de Reportes.

2.5.2 Listado de casos de uso.

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

CU-DR	Diseñar Reporte.
Actores	Usuario.
Descripción Proporciona la capacidad de crear un reporte como se desee, con formato, orden, agrupación, cálculo de totales.	
Referencias: RF1, RF9.	

CU-RR	Registrar Reporte.
Actores	Subsistema.
Descripción Proporciona la capacidad de registrar nuevos reportes a partir de plantillas tanto predefinidas (Tabulares) como enviadas por los subsistemas.	
Referencias: RF6.	

CU-VR	Visualizar Reporte.
Actores	Subsistema.
Descripción Visualizar un reporte. Se permitirá imprimirlo, almacenarlo o exportar los datos. Solo si es un reporte con parámetros (lo que se almacena son los valores de los parámetros para un reporte específico).	
Referencias: RF5, RF10.	

CU-GR	Gestionar Reporte.
Actores	Usuario.
Resumen	
<p>Se permitirá visualizar todos los reportes almacenados previamente sobre los cuales el actor logeado tiene acceso de visualización pudiendo seleccionar cualquiera de ellos para obtener los datos de interés e imprimirlos o exportarlos. Si el actor tiene además permisos de modificación (eliminación) podrá realizar estas operaciones. También se le permite la creación de nuevos reportes a través del asistente.</p>	
Referencias: RF2, RF3, RF4, RF7.	

CU-GPR	Gestionar Permisos para los Reportes.
Actores	Usuario, Subsistema.
Resumen	
<p>Gestiona los permisos que los actores tendrán sobre los reportes, los cuales pueden ser de visualización, modificación y eliminación. Solo puede establecer permisos sobre un reporte su creador o un administrador del subsistema "Generador de Reportes".</p>	
Referencias: RF8.	

2.5.3 Diagrama de casos de uso.

El modelo de casos de uso ayuda al cliente, a los usuarios y a los desarrolladores a llegar a un acuerdo sobre como utilizar el sistema.

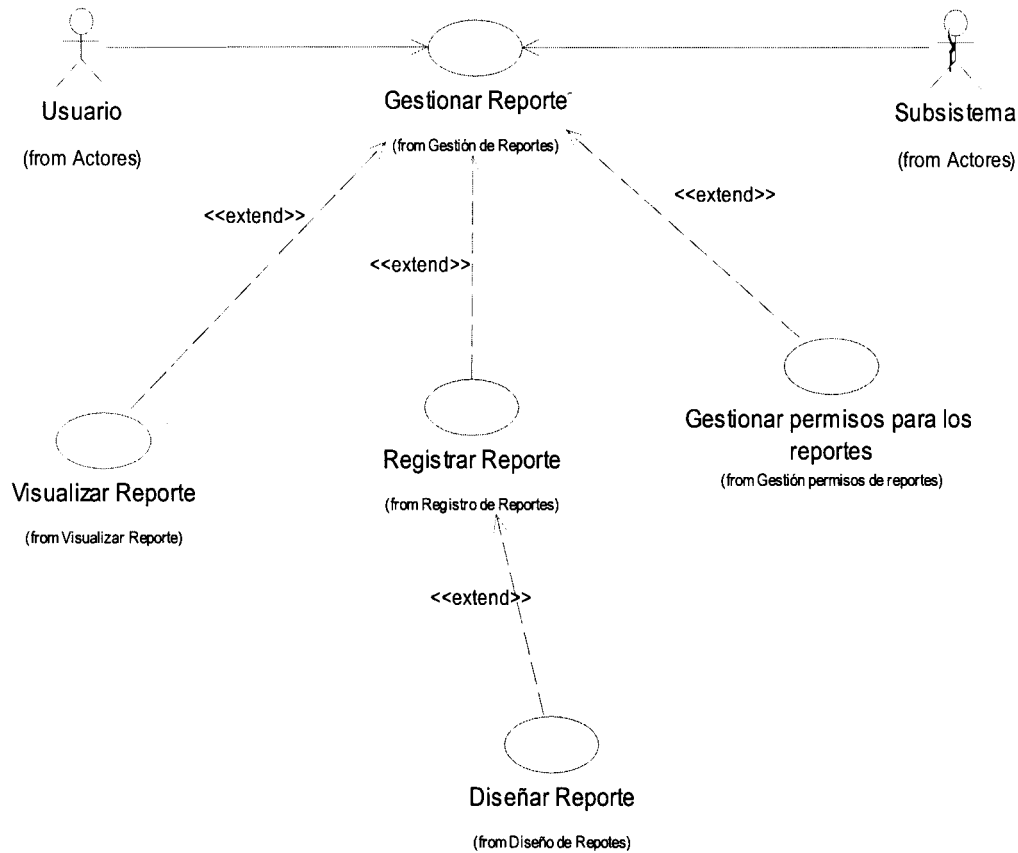


Fig 5. Diagrama de Casos de Uso.

2.5.4 Casos de uso por ciclos.

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.

2.5.4.1 Casos de uso del núcleo central.

Para la selección de los casos que deben iniciar el primer ciclo de desarrollo o núcleo central es necesario identificar cuáles son los que influyen de manera determinante en la arquitectura básica del sistema.

Código	Nombre de caso de uso.	Paquete	Justificación de la selección.
CU-DR	Diseñar Reporte	Diseño de Reportes	Son los casos de uso de mayor complejidad y representan los procesos primarios que se requieren para el funcionamiento del sistema e influyen en la arquitectura básica.
CU-GR	Gestionar Reporte	Gestión de Reportes	
CU-RR	Registrar Reportes	Registro de Reportes	
CU-VR	Visualizar Reporte	Visualizar Reporte	
CU-GPR	Gestionar Permisos para los Reportes	Gestionar permisos de reportes	

2.5.5 Especificación de 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] .

Ver Anexo 1.

Capítulo 3. Análisis y diseño del sistema

Para crear una aplicación de software hay que describir el problema y las necesidades o requerimientos: en qué consiste el conflicto y qué debe hacerse.

El análisis se centra en una investigación del problema, no en la manera de definir una solución. Para desarrollar una aplicación, también es necesario contar con descripciones detalladas y de alto nivel de la solución lógica y saber como satisface los requerimientos y las restricciones. El diseño pone de relieve una solución lógica: como el sistema cumple con los requerimientos.

Durante el análisis se procura ante todo identificar y describir los objetos dentro del dominio del sistema. Durante el diseño se procura definir los objetos lógicos del software que finalmente serán implementados en un lenguaje de programación. [UMLP]

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.

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

3.1 Análisis.

Un modelo de análisis ofrece una especificación más precisa de los requisitos que la que tenemos como resultado de la captura de estos, incluyendo al modelo de casos de uso. Los estructura de un modo que facilita su comprensión, preparación, modificación, en general, su mantenimiento.

3.1.1 Modelo de clases de análisis.

Una clase del análisis representa una abstracción de una y/o subsistemas del diseño del sistema. [BOCOO]

Existen tres estereotipos (Interfaz, Control y Entidad) estandarizados en UML y se utilizan para ayudar a los desarrolladores a distinguir el ámbito de las diferentes clases.

Ver Anexo 2.

3.2 Diseño.

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]

Adquirir una comprensión en profundidad de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia, tecnologías de interfaz de usuario.

3.2.1 Diagramas de Secuencia.

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

Forma parte del modelado dinámico del sistema. Se modelan las llamadas entre clases desde un punto concreto. Es útil para observar la vida de los objetos, identificar llamadas a realizar o posibles errores del modelado estático, que imposibiliten el flujo de información o de invocaciones entre los componentes.

Muestra cómo los objetos se comunican unos con otros en una secuencia de tiempo, qué sucede en cada momento, y para ello contienen objetos con sus ciclos de vida y los mensajes que se envían entre ellos ordenados secuencialmente. [ALV00]

Ver Anexo 4.

3.2.2 Diagramas de Clase.

El modelo conceptual muestra los conceptos básicos del sistema, sus partes y relaciones. Se realiza a través de un diagrama de clases de UML simplificado, en el cual se representan las clases preliminares, las asociaciones preliminares entre las clases, y los atributos de las clases. [ALV00].

Forma parte de la vista estática del sistema. En el se definirán las características de cada una de las clases, interfaces, colaboraciones y relaciones de dependencia y generalización.

Ver Anexo 3.

3.2.3 Descripción de las clases.

Nombre: TControlGestionarReporte
Controladora
Para cada responsabilidad:
Nombre: ConParámetro(RUI: Integer)
Descripción: Manda a verifica si el reporte invocado tiene parámetros.
Nombre: RepEspecifico(RUI: Integer)
Descripción: Manda a verifica si el reporte invocado es específico.
Nombre: getPermisosUsuarios(RUI:Integer)
Descripción: Pide todos los usuarios que tienen permisos de visualización sobre un reporte con sus respectivos permisos.
Nombre: EliminarPermiso(RUI: Integer)
Descripción: Manda a eliminar los permisos de todos los usuarios sobre un reporte.
Nombre: Adicionar(UUI: Integer, Nombre: String, RUI: Integer, Tipo: String)
Descripción: Le pide al Subsistema Seguridad que almacene un nuevo permiso del tipo especificado para un usuario dado en un reporte escogido.
Nombre: Modificar(UUI: Integer, Nombre: String, RUI: Integer, Tipo: String)
Descripción: Le pide al Subsistema Seguridad modificar los permisos que tiene un usuario sobre un reporte.
Nombre: RegistrarReporte (Nombre String, Titulo: String, SQL: String, Parámetros: String, Plantilla: String)
Descripción: Le pide a TControlRegistrarReporte que lo registre.
Nombre: getReportes
Descripción: Pide todos los reportes sobre los cuales el usuario logeado tiene permiso de visualización.
Nombre: EliminarReporte(RUI: Integer)
Descripción: Le pide a TReporte que elimine el reporte especificado.
Nombre: getPlantilla(PUI: Integer)
Descripción: Pide el identificador de la plantilla que se utilizará para elaborar

el reporte.
Nombre: getParametros(RUI: Integer)
Descripción: Le pide a TReporte los parámetros del reporte especificado.

Nombre: TControlRegistrarReporte
Controladora
Para cada responsabilidad:
Nombre: ParsearPlantilla(Plantilla: String, Titulo: String)
Descripción: Parsea la plantilla.
Nombre: Almacenar xml()
Descripción: Almacena la plantilla como un fichero XML

Nombre: TControlEsquema
Controladora
Para cada responsabilidad:
Nombre: getTablas
Descripción: Le pide a TEsquema todas las tablas de la base de datos.
Nombre: getCampos(Tabla: String)
Descripción: Le pide a TEsquema todos los campos de una tabla específica.
Nombre: ElaborarSentenciaSQL(Tablas, campos)
Descripción: Le pide a TEsquema que elabore una sentencia SQL a partir de las Tablas y Campos seleccionados.
Nombre: SentenciaSQLOrden(SQL: String, CamposTipoOrden:)
Descripción: Le manda a TEsquema que adicione una cláusula a la consulta elaborada anteriormente para ordenar los datos según los campos escogidos.
Nombre: setDatos(SQL: String, Titulo: String)
Descripción: Le pide a TEsquema que almacene los datos del reporte.

Controladora
Para cada responsabilidad:
Nombre: getUsers()
Descripción: Devuelve todos los usuarios que tienen acceso al sistema y no tienen permisos sobre el reporte especificado.
Nombre: Agregar(Usuario: Integer, Tipo: String, Nombre: String, RUI: Integer)
Descripción: Adiciona un nuevo permiso del tipo especificado sobre el reporte seleccionado para el usuario escogido.
Nombre: getPermisosUsuarios(RUI: Integer)
Descripción: Devuelve todos los usuarios que tienen permisos sobre un reporte especificado.
Nombre: EliminarPermisosUsuarios(RUI: Integer)
Descripción: Elimina los permisos de un usuario en un reporte especificado.

Nombre: TReporte
Entidad
Para cada responsabilidad:
Nombre: ConParametro(RUI: Integer)
Descripción: Verifica si el reporte especificado tiene parámetros.
Nombre: Especifico(RUI: Integer)
Descripción: Verifica si el reporte suministrado es un reporte específico.
Nombre: setDatos(Nombre: String, Titulo: String, SQL: String, PUI: String)
Descripción: Almacena los datos del reporte.
Nombre: getReportes()
Descripción: Devuelve todos los reportes existentes.
Nombre: EliminarReportesEsp(RUI: Integer)
Descripción: Elimina el reporte específico especificado.

Nombre: setParametros(RUI: Integer, Parámetros: String)
Descripción: Almacena los parámetros de un reporte.

Nombre: TEsquema
Entidad
Para cada responsabilidad:
Nombre: getTablas()
Descripción: Devuelve todas las tablas que tiene la base de datos.
Nombre: getCampos(tablas: String)
Descripción: Devuelve todos los campos de una tabla.
Nombre: ElaborarSentenciaSQL(Tablas, Campos)
Descripción: Elabora una sentencia SQL con las tablas y los campos escogidos por el usuario.
Nombre: SentenciaSQLOrden(SQL: String, CamposTipoOrden)
Descripción: Le agrega a la sentencia SQL la cláusula para el ordenamiento por campos específicos y el tipo de ordenamiento por campo, por defecto ascendente.
Nombre: setdatos(SQL: String, Titulo: String)
Descripción: Almacena los datos del reporte.

Nombre: TParametro	
Entidad	
Atributo	Tipo
Nombre	Cadena
Tipo	Cadena
Para cada responsabilidad:	
Nombre: setParametro(RUI: Integer, parametros: String)	
Descripción: Almacena los valores de los parámetros de un reporte especificado.	
Nombre: getParametro(RUI)	
Descripción: Devuelve los parámetros de un reporte especificado.	

3.3 Diseño de BD.

Las tablas de la base de datos se originan a partir del modelo conceptual. La base de datos del sistema es sencilla, y se utiliza solo para almacenar algunos de los aspectos relacionados con la configuración de los Reportes.

3.3.1 Modelo Relacional.

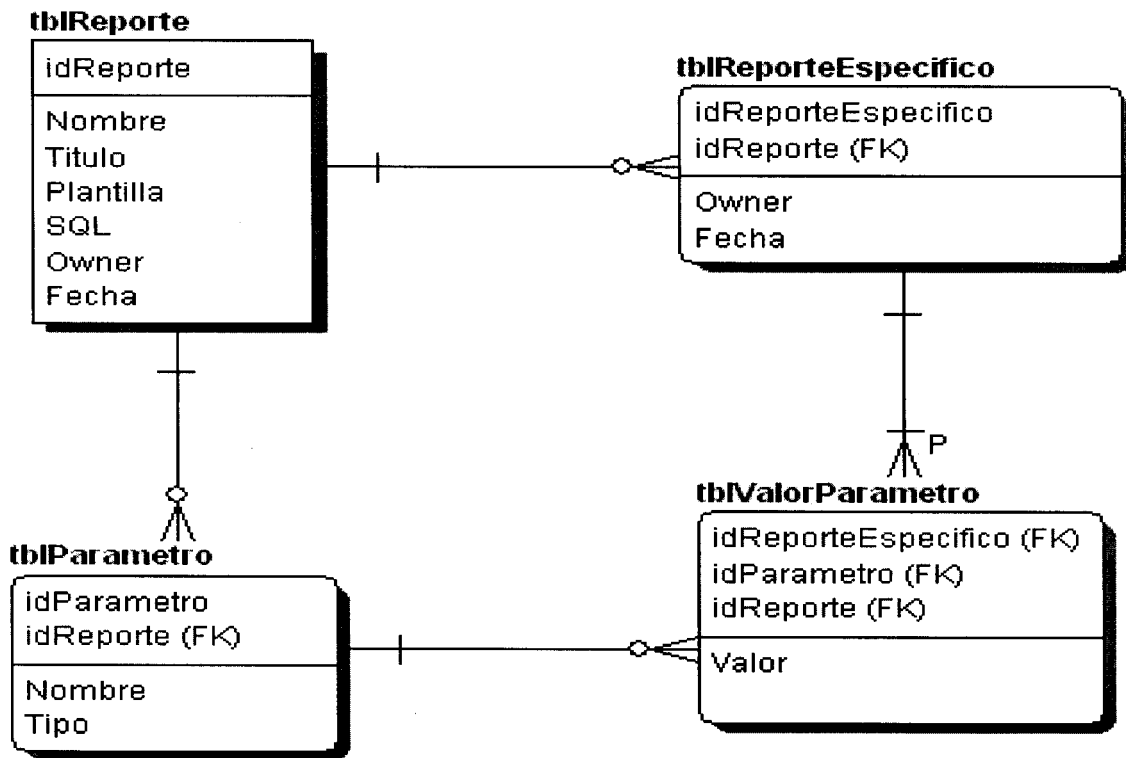


Fig. 6 Modelo Relacional.

3.3.2 Descripción de las tablas.

Nombre: tblReporte		
Descripción: Almacena los datos generales de los reportes.		
Atributo	Tipo	Descripción
idReporte	Entero	Identificador del reporte.
Nombre	Cadena	Nombre del reporte.
Titulo	Cadena	Titulo del reporte.
Plantilla	Cadena	Nombre de la plantilla que se usará para generar el reporte.
SQL	Cadena	Sentencia SQL asociada al reporte para su generación.
Owner	Entero	El usuario que registró el reporte.
Fecha	DateTime	La fecha en que fue registrado el reporte.

Nombre: tblParametro		
Descripción: Almacena los datos generales de los parámetros de un reporte.		
Atributo	Tipo	Descripción
idParametro	Entero	Identificador del parámetro.
idReporte	Entero	Identificador del reporte al cual pertenece el parámetro.
Nombre	Cadena	Nombre del parámetro.
Tipo	Cadena	Tipo de datos parámetro.

Nombre: tblReporteEspecifico		
Descripción: Almacena los datos generales de los reportes más solicitados por el usuario, dándole valores a sus parámetros para agilizar su trabajo.		
Atributo	Tipo	Descripción
idReporteEspecifico	Entero	Identificador del reporte específico.
idReporte	Entero	Identificador del reporte al cual pertenece.
Owner	Cadena	Usuario creador del reporte.
Fecha	DateTime	Fecha de elaboración del reporte.

Nombre: tblValorParametro		
Descripción: Almacena los datos generales de los valores de los parámetros de un reporte específico.		
Atributo	Tipo	Descripción
idReporteEspecifico	Entero	Identificador del Reporte Especifico al cual pertenece.
idParametro	Entero	Identificador del parámetro al cual se le especificará el valor.
idReporte	Entero	Identificador del reporte al cual pertenece el reporte específico.
Valor	Cadena	Valor del parámetro de un reporte específico.

3.4 Conclusiones

Se finaliza la etapa de análisis y diseño del sistema, habiéndose expandido 5 casos de uso y se confeccionaron sus respectivos diagramas de interacción del sistema. Estos diagramas se graficaron mediante diagramas de secuencia. Con todo lo planteado se tiene una idea más precisa sobre los elementos constitutivos del sistema que se propone.

Conclusiones

El subsistema “Generador de Reportes” constituye un intento de elaborar un modelo flexible de automatización de la gestión y prestación de servicios, particularizando su funcionamiento en la UCI, donde se implementan las funciones principales de la gestión y los servicios. Esta característica le permite adaptarse con facilidad a variaciones en la estructura del flujo informativo, incluyendo la estructura administrativa o la forma en que se brindan los servicios.

Para lograr tal objetivo se implementó un asistente de informe con características similares a las del asistente de informe de Microsoft Access el cual se considera de fácil uso y bastante flexible, que son un par de características a las cuales se le otorga gran importancia en el proyecto.

El proveedor de servicios de reportes para los subsistemas del sistema de gestión académica está destinado a la elaboración y almacenamiento de los reportes. La elaboración que es una acción cooperativa entre el proveedor y los subsistemas se logra mediante estándares a través de plantillas en código HTML o XML y sentencias SQL. Este método es más costoso en cuanto a tiempo de trabajo sin embargo evita el uso del sistema CrystalReport.Net, de modo que no será necesaria la patente de dicho sistema en caso de comercialización del sistema.

Para el desarrollo del trabajo fue necesario el aprendizaje de los Sistemas Gestores de Bases de Datos Microsoft Access y SQL Server, el lenguaje de programación C#.Net, el generador de reporte CrystalReport.Net, y el marco de trabajo ASP.Net, todos de gran utilidad ofreciendo herramientas de gran utilidad y además despertando ideas que han sido incorporadas al sistema.

El trabajo realizado además de no tener término aún, va incorporado al sistema de gestión académica el cual es desarrollado por un grupo de trabajo cuyos miembros hemos sostenido un amplio intercambio para poder coordinar ideas y lograr la

cooperación necesaria, debido a que los subsistemas realizados no son independientes. Por tanto creo que esta es una de las experiencias que además de nueva pues en el transcurso de la carrera no hubo ningún trabajo serio en equipo es muy interesante y da la idea del potencial que tiene y la responsabilidad que se debe asumir.

Recomendaciones

La versión de “Generador de Reporte” presentada en este trabajo de diploma, es solo una versión de prueba de lo que será un gran subsistema integrado para la gestión de los servicios de información en la UCI. Como se puede ver, existe un amplio campo de desarrollo del sistema en el futuro.

Se recomienda:

- Implantar el sistema en la Intranet de la Universidad de las ciencias Informáticas para dar solución inmediata a los problemas detectados en la etapa de estudio preliminar.
- Continuar la implementación y desarrollo del subsistema, dando continuidad al ciclo de desarrollo número dos.
- Divulgar el estudio realizado entre los profesionales de la información.
- Continuar con la investigación para garantizar nuevas y buenas mejoras en futuras versiones del sistema.

Referencias Bibliográficas

- [BOC00] BOCH Grady, RUMBAUGH James, JACOBSON Ivar. "El lenguaje unificado de modelado. Manual de referencia".2000. Addison Wesley.
- [UMLP] C. Larman, UML y patrones, introducción al análisis y diseño orientado a objetos, Segunda edición, Prentice-Hall, 2002.
- [IASP] Introducción ASP.NET.
<http://desarrollo.ethek.com/Desarrollo/contenido.asp?IDContenido=99>
- [ABC] "ABC de Oracle". <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>
- [CON]: CONALLEN, Jim. "Modeling Web Applications with UML".
<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] Quintana Rodríguez, Julio T. "Sistemas distribuidos con componentes", en <http://www.uv.mx/iiesca/revista4/distribuidos.htm>
- [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] 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>
- [5] http://www.semicrol.es/gestion_universitaria2/gauss/gauss.html# 05.07.2004.
- [6] Desarrollo de Software – Sistemas. Sistema Académico para Facultades <http://www.cnc.una.py/desoft/sacad.html> 08.07. 2004.
- [7] Sistema de Información Académica <http://www.siaun.unal.edu.co/> 08.07. 2004.

- [8] Manual de ASP <http://www.webestilo.com/asp/> 23.06.2004.
- [9] Sistema de información de gestión académica. <http://www.siga.utfsm.cl/> 05.07.2004.
- [10] S.I.G.A. Principal. Sistema Integrado de Gestión Académica. <http://www.dara.es/siga/> 05.07.2004.
- [11] GAT Universidades. Gestión Académica para universidades. <http://www.uoc.edu/ideasolutions/b/menu131.htm> 08.07. 2004.
- [12] Curso de SQL <http://www.cybercursos.net/sql/sql.html> 08.07. 2004.
- [13] Tutorial de SQL <http://www.desarrolloweb.com/manuales/9/> 08.07. 2004.
- [14] Estructuras de Oracle <http://www.programacion.com/tutorial.php?id=oracle> 08.07. 2004.
- [15] Bischof, Brian. Crystal Reports .NET Programming. Copyright 2004.
- [16] Sentencias SQL en Visual Basic <http://www.monografias.com/trabajos7/sql/sql.shtml#intro>
- [17] Funciones en SQL 2000 <http://www.sqlmax.com/func1.asp>
- [18] Iniciación al Lenguaje SQL <http://www.lania.mx/biblioteca/seminarios/basedatos/sql.html#sentencia>

- [19] Report-writing services Crystal Reports PFW Systems
<http://www.lawebcrystal.com/crearrept.asp>
- [20] Web Services: Análisis de sus perspectivas partiendo del panorama internacional Abril, 2003 Proyecto "UCI Ciudad Digital"
- [21] Módulo de reporteador
<http://www.procesac.com/productos/visionweb/visionwebrepo.asp>
- [22] 5-Cinco. Sistema Comercial y Contable <http://www.5cinco.com/index.htm>
- [23] Reportes en .Net con Cristalreport
<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art11.asp>
- [24] Informes de MOTORSHOP <http://www.motorshop.es/informes.html>
- [25] Solomon <http://www.microsoft.com/businessSolutions/Solomon/default.msp>
- [26] Report generation in .Net
<http://www.c-sharpcorner.com/Code/2004/Feb/ReportsInDotNet.asp>
- [27] WEB-SERVICES: LA SIGUIENTE GENERACIÓN DE INTERNET
<http://revista.robotiker.com/articulos/articulo62/pagina1.jsp>
- [28] Sant'Anna, Mauro. Reportes en .Net con Cristal Reports

Glosario de Términos

- **Usuario final:** Entendido éste como cualquier persona que requiera un reporte.
- **Informes:** Los informes nos permiten imprimir la información de base de datos en una forma eficaz y además, permite combinar los datos de varias tablas en un documento único.
- **Cascada:** Regla por la cual si un valor en una tabla se borra, ello debe afectar a los valores idénticos de las tablas dependientes.
- **Clave foránea:** Sirve para relacionar lógicamente varias tablas. Es una columna o combinación de columnas de una tabla cuyos valores coinciden con los valores de la clave primaria de otra tabla.
- **Clave primaria:** Columna o columnas utilizadas para identificar de manera única las filas de una tabla.
- **Columna:** También llamado campo o atributo, se encarga de describir una característica del valor de los datos de una tabla.
- **Fila:** También llamada registro o t pula, es un conjunto de valores que representan las ocurrencias de datos de una tabla.
- **Funci n agregada:** Operaci n SQL con un valor de resumen de un conjunto de valores.
- **Integridad referencial:** Operaci n que garantiza que las referencias de una tabla a otra son v lidas. Permite prevenir los problemas de inconsistencia de datos.
- **Join:** Una consulta que extrae datos de dos o m s tablas. La m s sencilla es el llamado producto cartesiano. Otros tipos son: self-join, equi-join, natural inner join, theta join, outer join.
- **Relaci n:** En un sistema relacional una relaci n es una tabla. Cada fila es una ocurrencia diferente para las mismas columnas. Ello constituye una relaci n. En lenguaje com n se entiende por relaci n una uni n o join.
- **SQL (Structured Query Language):** Conjunto est ndar de comandos para gestionar bases de datos relacionales por sus mismas caracter sticas relacionales.
- **Tabla:** Es la estructura de almacenamiento fundamental de una base de datos relacional.

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

- **Artefactos:** Una parte de la información que (1) es producida, modificada, o usada por un proceso, (2) define un área de responsabilidad, y (3) está sujeta al control de versión. Un artefacto puede ser un modelo, un elemento del modelo, o un documento. Un documento puede adjuntar otros documentos.

Una parte de la información que es usada o producida por un proceso de desarrollo del software. Un artefacto puede ser un modelo, una descripción, o un software.

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

- **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.

- **Correo Electrónico:** Servicio que permite enviar y revisar mensajes de forma electrónica, a través de las redes de computadoras, estos mensajes generalmente se envían en formato ASCII y pueden adjuntársele otros archivos.

- **FTP:** “File Transferring Protocol” Protocolo para la Transferencia de Ficheros. Es uno de los servicios Internet más comunes para mover archivos entre un computador y otro. Es habitual usar FTP desde un sitio en Internet para rescatar o enviar archivos. Existen muchos sitios Internet que suministran información gratuita contenida en archivos y que pueden ser accesadas usando el protocolo FTP, usando para el acceso el nombre de cuenta anonymous.

- **Interfaz:** Una colección de operaciones que se usan para especificar el servicio de una clase o de un componente.

Un juego nombrado de operaciones que caracterizan la conducta de un elemento.

La Interfaz hombre-máquina es un canal comunicativo entre el usuario y el ordenador.

- **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:** Es una adaptación de las mismas tecnologías que existen en Internet, para que sean utilizadas dentro de la red interna de una empresa u organización de forma tal que sus miembros puedan intercambiar información de todo tipo, utilizando el Web como interfaz común.
- **.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.
- **Proceso:** Secuencia de actividades invocadas para producir un producto de software.
- **Rol:** Papel, cometido o función que tiene o desempeña, que interpreta un actor.
- **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.
- **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.
- **Software:** Palabra en inglés utilizada para indicar a los programas de computadoras, a las aplicaciones.
- **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.

- **UML:** “Unified Modeling Language” Lenguaje gráfico que brinda un vocabulario y reglas para especificar, construir, visualizar y documentar los artefactos de un sistema utilizando el enfoque orientado a objetos.
- **Usuario:** Persona que usa ordinariamente una cosa.
- **Gestión Académica:** La gestión académica involucra el planteamiento y la programación curricular, la producción de materiales didácticos, la ejecución, la evaluación, articulación de los aprendizajes, la enseñanza y aprendizaje enmarcados en un concepto de exigencia, excelencia académica y mejora continua.

Anexos

Anexo 1. Casos de Uso expandidos

Diseñar Reporte

Caso de Uso	Diseñar Reporte.	
Actores	Usuario.	
Propósito	Crear un informe deseado.	
Resumen		
El caso de uso comienza cuando el Usuario desea obtener alguna información deseada del sistema. Proporciona la capacidad de crear un reporte como se desee, con formato, orden, agrupación, cálculo de totales.		
Referencias: RF1, RF9.		
Acción del actor	Respuesta del sistema	
1. El caso de uso comienza cuando el usuario solicita diseñar un nuevo informe.	2. El sistema muestra una interfaz para la elaboración del informe deseado donde se le presenta al usuario las tablas a las que tiene acceso y de cada una de ellas todos los campos.	
3. El actor escoge los campos de las tablas que desea para elaborar su informe.	4. El sistema muestra una interfaz para la continuación de la elaboración del informe donde se le presenta al usuario los campos escogidos para que agregue algún nivel de agrupamiento si lo desea.	
5. El actor escoge el nivel de	6. El sistema muestra una	

agrupamiento que desea(es opcional).	interfaz para la continuación de la elaboración del informe donde se le permite al usuario escoger que nivel de ordenación e información de resumen desea utilizar para los registros de detalles.
7. El actor escoge el nivel de ordenamiento y los resúmenes que desea aplicarle a su informe(es opcional).	8. El sistema muestra una interfaz para la continuación de la elaboración del informe donde se le presenta al usuario los formatos que puede darle al informe.
9. El actor escoge el nivel de distribución y orientación que desea aplicarle al informe(es opcional).	10. El sistema muestra una interfaz para la continuación de la elaboración del informe donde se le presenta al usuario una lista de distribuciones que puede aplicar a su informe.
11. Escoge la distribución que desea aplicarle al informe(es opcional).	12. El sistema muestra una interfaz para la continuación de la elaboración del informe donde se le muestra al usuario una lista de estilos que puede escoger para aplicarle al informe.
13. Escoge el estilo que desea aplicar al informe(es opcional).	14. El sistema muestra una interfaz para la continuación de la elaboración del informe donde se le pide al usuario la

	entrada del título del informe y si desea filtrar la información.
15. Introduce el título que desea para su informe y los campos por los cuales quiere filtrar la información si lo desea también(opcional).	16. El sistema obtiene el título del informe, lo registra y visualiza: a) Ver Caso de Uso Registrar Reporte. b) Ver Caso de Uso Visualizar Reporte.
Precondición El actor se ha logeado correctamente en el sistema obteniendo como resultado que se muestre la interfaz principal donde se visualizarán las opciones referentes a cada uno de ellos en dependencia del tipo de actor que sea.	
Poscondición La instancia del caso de uso termina cuando se ha registrado el reporte y se le ha mostrado al usuario.	
Caminos Alternativos 1. El actor puede cancelar el caso de uso en el paso que desee. 2. En el paso tres si el actor no selecciona la tabla(s) y los campo(s) de estas que desee para elaborar su informe el sistema debe emitir un mensaje de error. 3. En el paso 15 si el actor no específico el título del informe el sistema debe de emitir un mensaje de error.	

Registrar Reporte

Caso de Uso	Registrar Reporte
Actores	Subsistema.
Propósito	Dar la posibilidad a los subsistemas de registrar sus propios reportes.
Resumen	
Este caso de uso comienza cuando el Subsistema solicita registrar un reporte. Proporciona la capacidad de registrar nuevos reportes a partir de plantillas tanto predefinidas (Tabulares) como enviadas por los subsistemas.	
Referencia: RF6.	
Acción del actor	Respuesta del sistema
1. El caso de uso comienza cuando alguno de los Subsistemas pide registrar un reporte y envían la solicitud con el nombre del reporte, título, la sentencia SQL, los parámetros y la Plantilla (esta última si no va a desarrollar un reporte tabular y los parámetros si no es un reporte estático).	2. El sistema obtiene la solicitud enviada.
	3. El sistema chequea si se desea registrar un reporte que puede ser: a) Ver sección "Registrar Reporte con plantilla". b) Ver sección "Registrar Reporte Tabular".
Sección: Registrar Reporte con plantilla	
	2. El sistema construye el reporte con la plantilla enviada y almacena su URL

	junto con el nombre, Título, la sentencia SQL y los parámetros y envía al subsistema una confirmación de registro.
Sección: Registrar Reporte Tabular	
	2. El sistema construye el reporte con una plantilla tabular predefinida y almacena su URL junto con el nombre, la sentencia SQL y los parámetros y envía al subsistema una confirmación de registro.
Precondición	
Obtener una solicitud para registrar el reporte.	
Poscondición	
La instancia del caso de uso termina cuando se registra el informe.	
Caminos Alternativos	
1. En el paso dos de la sección "Registrar Reportes con Plantilla" ó "Registrar Reporte Tabular" si los datos suministrados no son correctos el sistema debe emitir un mensaje de error.	

Visualizar Reporte

Caso de Uso	Visualizar Reporte
Actores	Subsistema.
Propósito	Mostrar un reporte.
Resumen	
Comienza cuando se desea visualizar algún reporte. Se permitirá imprimirlo o exportar los datos.	
Referencia: RF5, RF10.	
Acción del actor	Respuesta del sistema
1. El caso de uso comienza cuando el Subsistema pide visualizar un reporte y envía la solicitud con el nombre del reporte.	2. El sistema obtiene la solicitud del subsistema y verifica si es un: <ul style="list-style-type: none"> a) Reporte con parámetros. b) Reporte Usual: El sistema obtiene el reporte solicitado y lo visualiza. c) Reporte Específico: El sistema obtiene los valores de los parámetros elabora el reporte y lo visualiza.
	3. Después de mostrado el informe se puede: <ul style="list-style-type: none"> a) Imprimir el informe: El sistema manda a imprimir el informe. b) Exportar los datos: El sistema manda a exportar el informe hacia el formato pedido. c) Almacenar el Informe: El sistema permite almacenar el informe solo si es un reporte con parámetros. El sistema obtiene

	quién creo el reporte, la fecha del sistema y el nombre y lo almacena como un reporte usual.
Sección: Reporte con parámetros	
Parámetros del Reporte	
	2. Muestra una interfaz para que el actor especifique los parámetros deseados.
3. Suministra los parámetros que necesita el reporte.	4. El sistema obtiene los parámetros suministrados y visualiza el reporte.
Precondición Que el informe haya sido registrado con anterioridad.	
Poscondición La instancia del caso de uso termina cuando se muestra el reporte o cuando se imprime, exporta o almacena.	
Caminos Alternativos 1. En el paso cuatro de la sección Reporte con parámetros si el actor no suministro los parámetros correctos mostrar un mensaje de error.	

Gestionar Reporte

Caso de Uso	Gestionar Reporte
Actores	Usuario, Subsistema.
Propósito	Visualizar los reportes almacenados previamente y permitir operar con ellos (Modificarlos, Eliminarlos, Exportarlos e Imprimirlos).
Resumen	
Comienza cuando se desea interactuar con el subsistema. Se permitirá visualizar solo los reportes a los que el actor logeado tiene acceso de visualización pudiendo seleccionar cualquiera de ellos para obtener los datos de interés e imprimirlo o exportar los datos si desea. Si el actor tiene además permisos de modificación (eliminación) podrá realizar estas operaciones.	
Referencia: RF2, RF3, RF4, RF7.	
Acción del actor	Respuesta del sistema
1. El caso de uso comienza cuando el actor decide interactuar con el subsistema reporteador.	2. El sistema verificará los permisos que tiene el actor sobre los reportes y visualizará solo aquellos a los que el actor logeado tiene acceso de visualización, permitiendo: <ul style="list-style-type: none"> a) Ver caso de uso "Visualizar Reporte". b) Eliminar Reporte: El actor escoge los reportes que desea eliminar y el sistema obtiene la petición, elimina los reportes y actualiza la página. c) Ver sección "Diseñar Reporte"

	<p>d) Ver sección "Modificar Reporte".</p> <p>e) Ver caso de uso "Gestionar permisos para los reportes".</p>
Sección: Modificar Informe	
Imagen de la Ventana	
	<p>2. El sistema le muestra una interfaz al actor para que escoja los aspectos del informe que desea modificar (resumen, ordenación, agrupación, eliminación o agregación de campos, título).</p>
<p>3. El actor escoge las modificaciones que desea aplicarle al informe.</p>	<p>4. El sistema construye una nueva cadena reutilizando la elaborada anteriormente pero añadiendo o modificando según los datos suministrados por el actor. Almacena los datos necesarios y visualiza el informe modificado.</p>
<p>Precondición</p> <p>Cualquier actor del sistema que se logee correctamente obteniendo como resultado que se muestre la interfaz principal donde se visualizarán las opciones referentes a cada uno de ellos en dependencia del tipo de actor que sea.</p>	
<p>Poscondición</p> <p>La instancia del caso de uso termina cuando se ha gestionado el reporte (visualizado, eliminado, modificado o cuando se le han gestionado los permisos).</p>	

Caminos Alternativos

1. En el paso dos b) y d) si el actor que inicia la instancia del caso de uso no tiene permiso de modificación (eliminación) en el reporte escogido no se le mostrarán las opciones de modificar (eliminar) el informe.
2. En el paso dos d) si el reporte escogido por el actor no es un reporte tabular no se le mostrará la opción de modificar.
3. En el paso dos si el actor que instancia el caso de uso no tiene permiso de visualizar al menos uno de los reportes del sistema emitir un mensaje aclarando que no puede ver ningún reporte que por favor lo registre previamente.
4. Cuando el actor seleccione un reporte determinar si es su dueño y si no lo es no mostrarle la opción de gestionar los permisos para ese reporte.

Gestionar permisos para los reportes

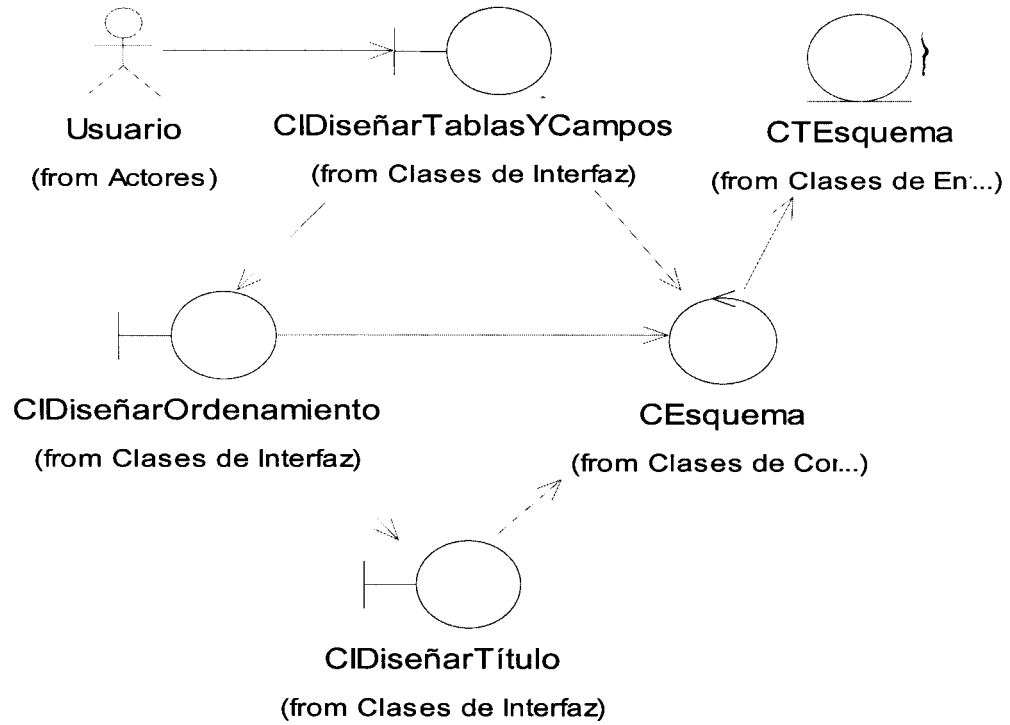
Caso de Uso	Gestionar permisos para los reportes.
Actores	Subsistema.
Propósito	Gestiona los permisos que los actores tendrán sobre los reportes.
Resumen	
Comienza cuando se desean establecer permisos sobre los reportes existentes, así como modificar o eliminar los ya establecidos.	
Referencia: RF8.	
Acción del actor	Respuesta del sistema
1. El caso de uso comienza cuando el subsistema desea gestionar permisos para los reportes o después de haber sido registrado un reporte, establecer sus permisos.	2. El sistema verifica que el actor es el creador del reporte y le muestra una interfaz donde se le presentan todos los usuarios que tienen algún permiso sobre el reporte junto con los respectivos permisos que pueden proporcionárseles, incluidos los que ya posee. Los permisos son: Visualizar, Modificar y Eliminar. Y le permite al actor: a) Ver sección "Añadir un nuevo usuario" b) Eliminar un usuario existente: El actor manda a eliminar a los usuarios que no desea que tengan ningún permiso sobre su informe. El sistema elimina todos los

	<p>permisos que tienen esos usuarios sobre dicho reporte y actualiza la página.</p> <p>c) Aplicar Modificación: El actor escoge los permisos que desea que tenga cada usuario del reporte y el sistema se encarga de almacenar y actualizar estos permisos.</p> <p>d) Aceptar: El sistema se encarga de obtener todas las modificaciones hechas sobre los permisos de los usuarios, almacenarlas y cerrar la ventana.</p> <p>e) Cancelar: El sistema se encarga de cerrar la ventana sin aplicar ninguna modificación.</p>
<p>Sección: Añadir un nuevo usuario</p>	
<p>Añadir Usuario</p>	
	<p>2. El actor previamente habrá escogido a que reporte quiere establecerle un nuevo usuario para definirle permisos sobre este. Se le mostrarán todos los usuarios que pueden acceder al subsistema y todos los roles que tienen los mismos permisos.</p>
<p>3. Escoge un nuevo usuario.</p>	<p>4. Muestra la interfaz anterior:</p>

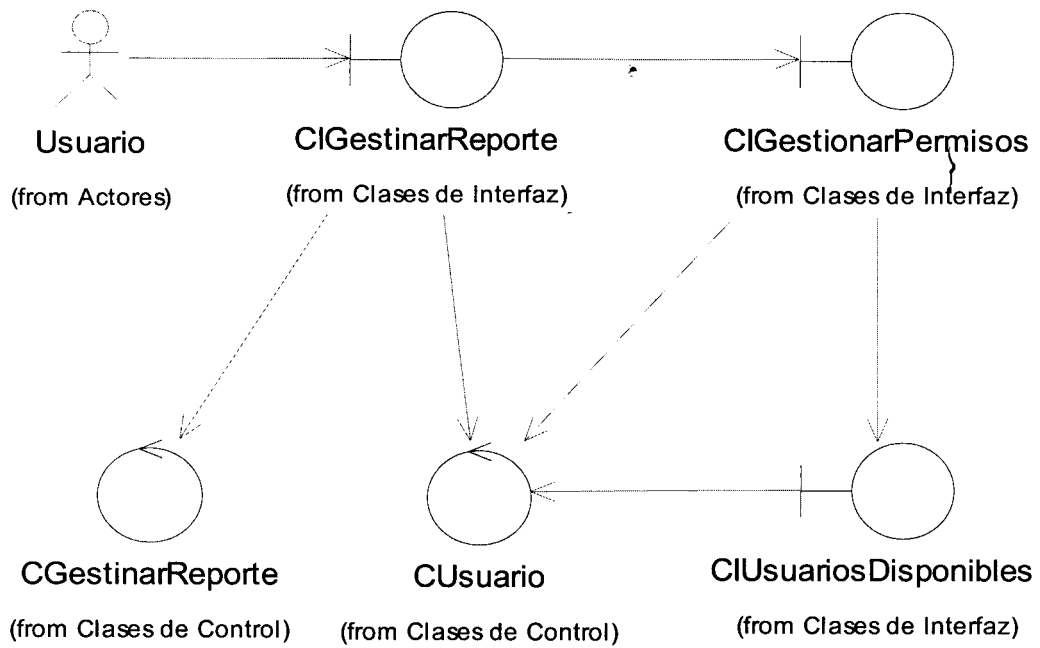
	Visualizar usuarios con permisos.
5. Establece los permisos deseados para el usuario.	6. Obtiene los permisos establecidos y los almacena.
Precondición El actor ha registrado correctamente un reporte y desea establecer sus permisos.	
Poscondición La instancia del caso de uso termina cuando se hayan gestionado los permisos.	
Caminos Alternativos	

Anexos 2. Diagrama de Clases del análisis.

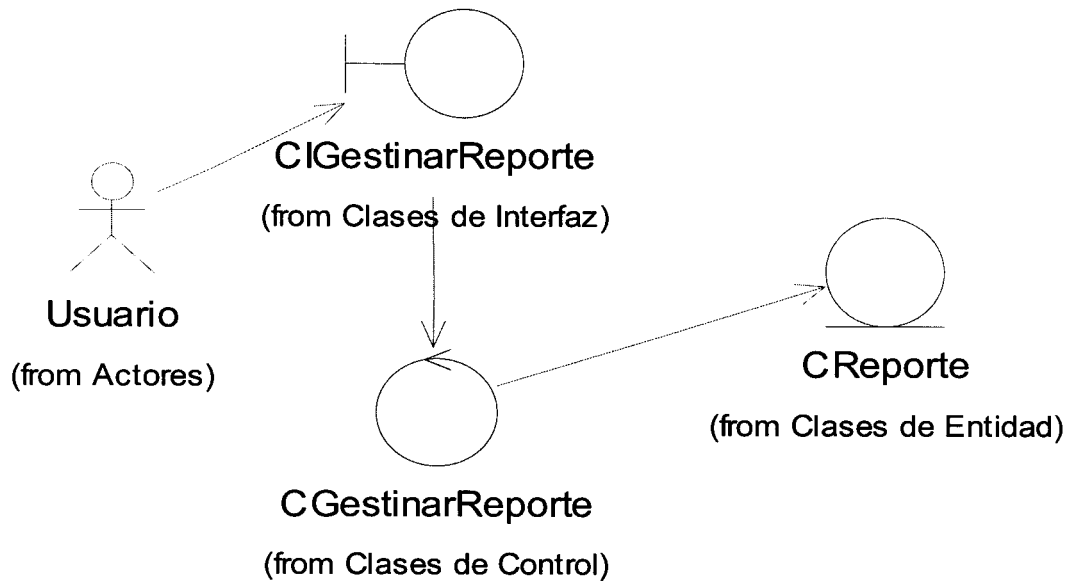
Diseñar Reporte



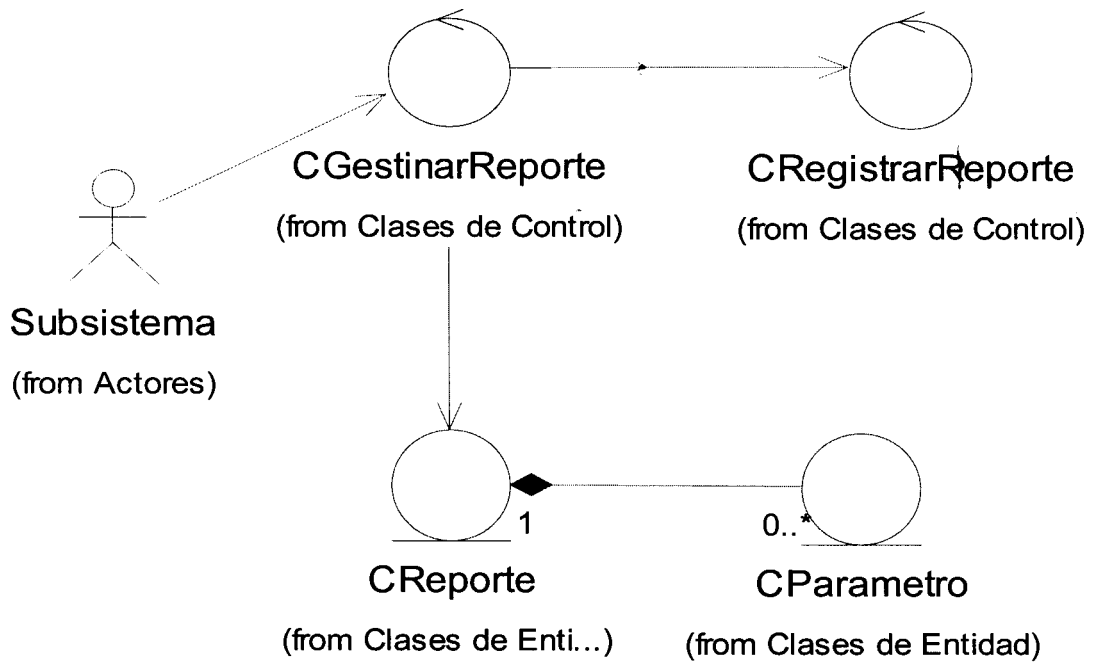
Gestionar Permiso para los Reportes



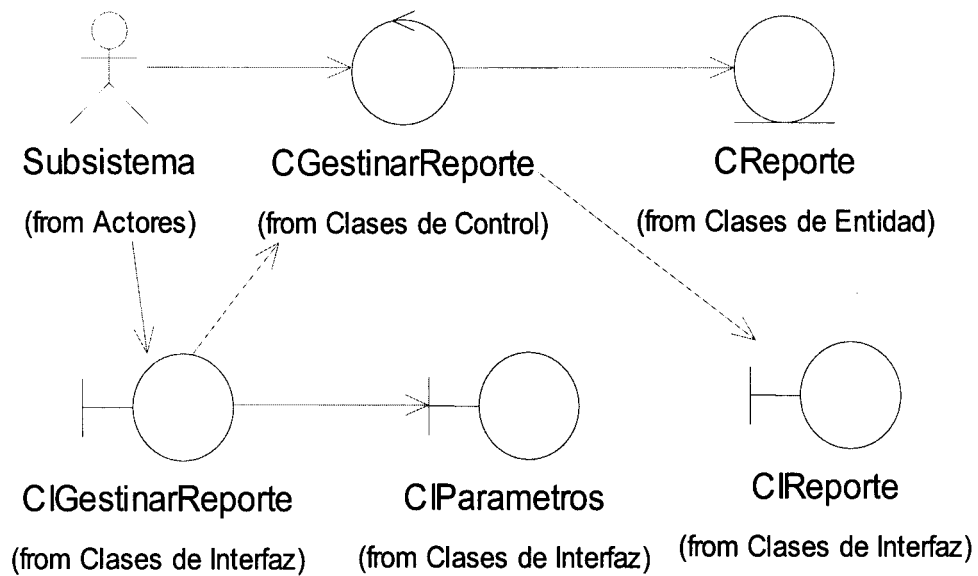
Gestionar Reporte



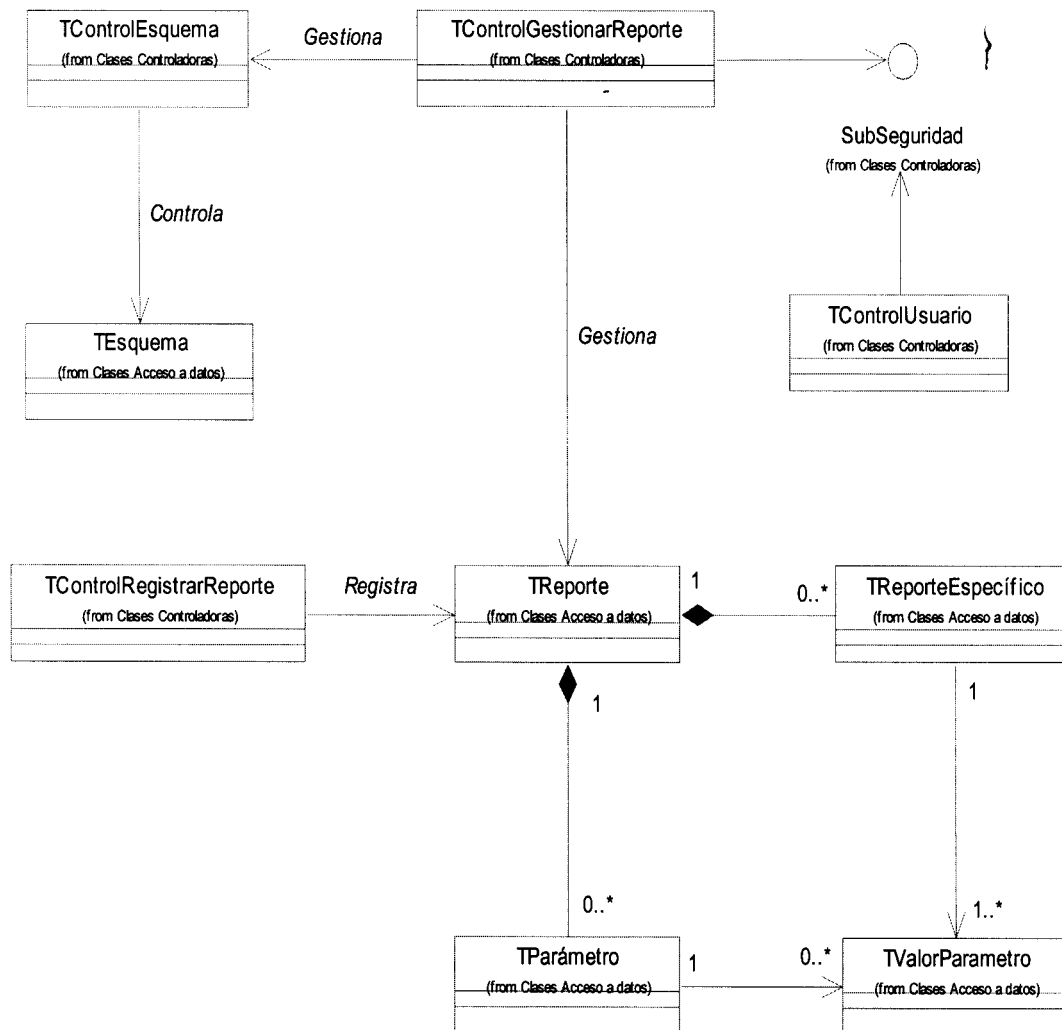
Registrar Reporte



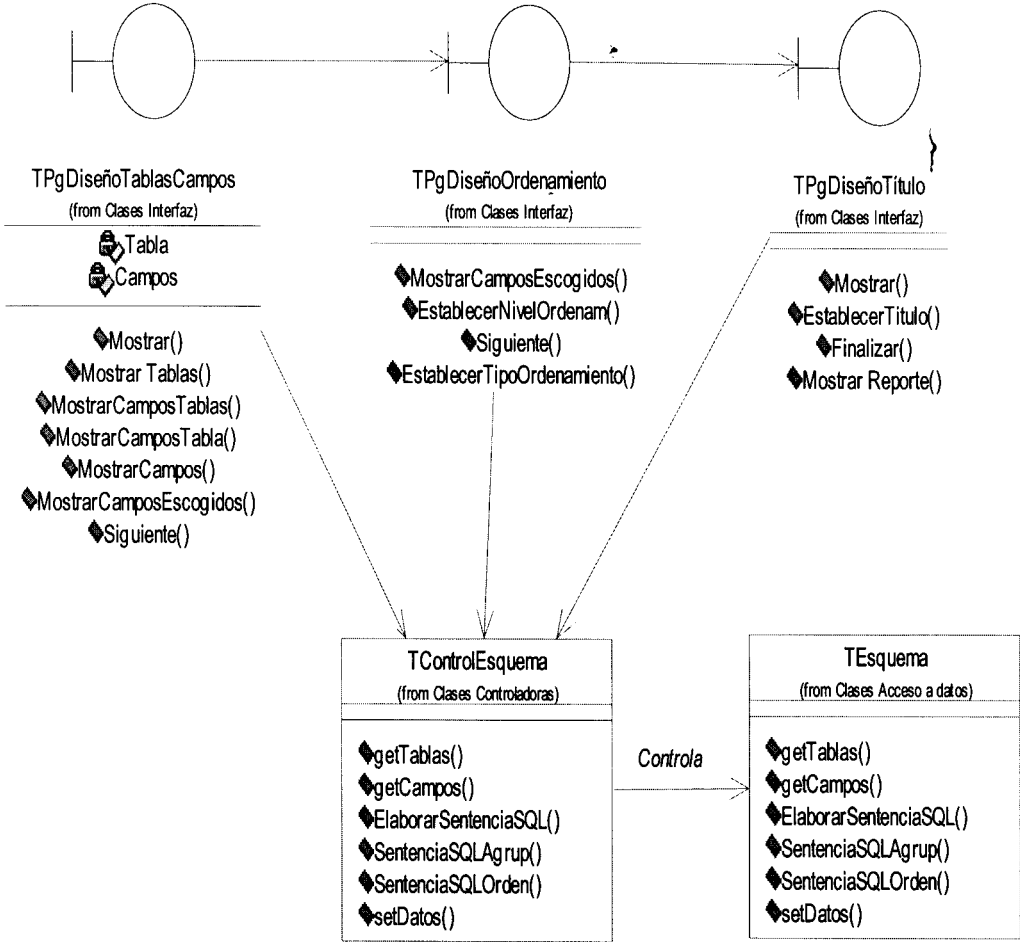
Visualizar Reporte



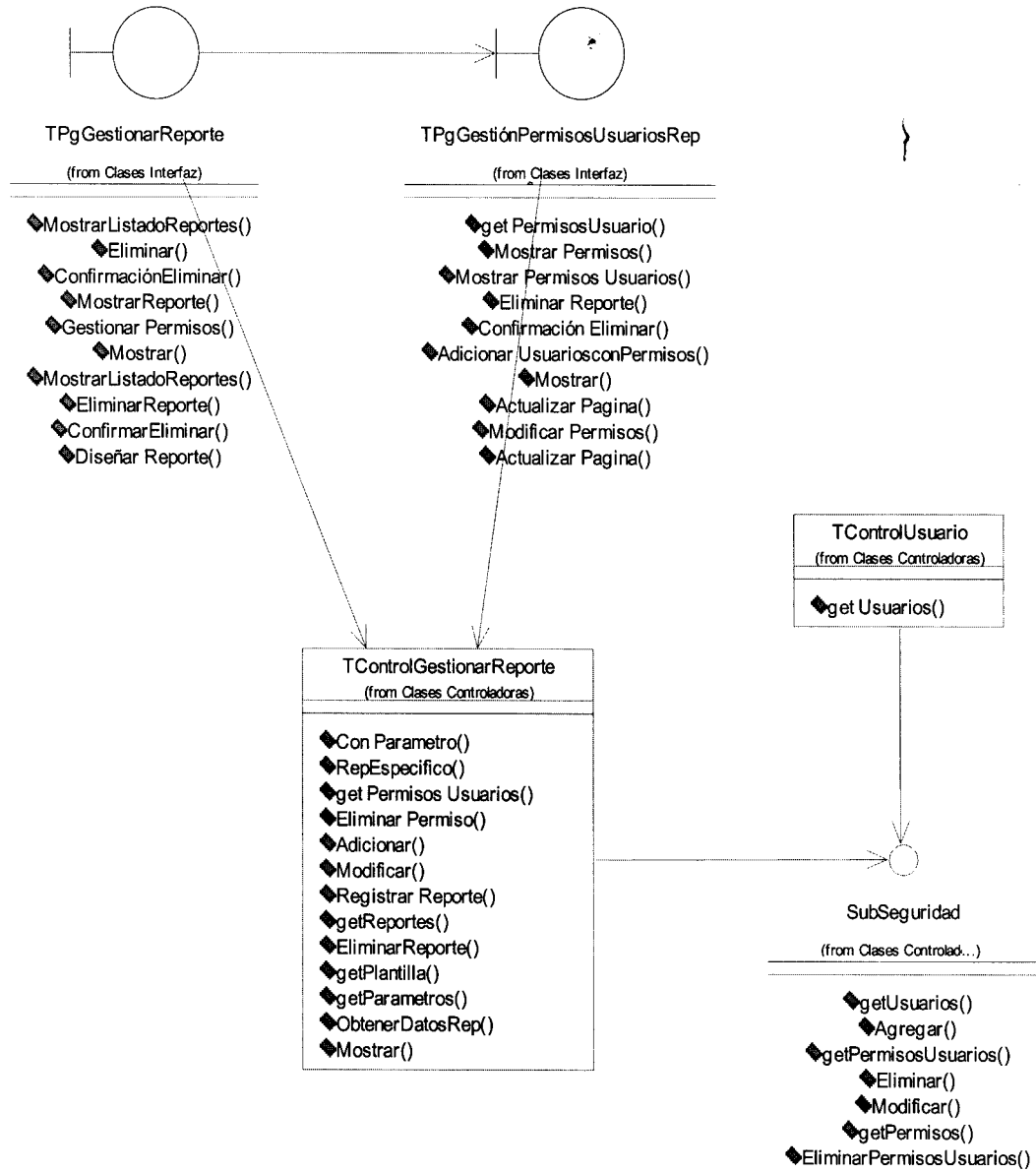
Anexo 3. Diagramas de Clases del Diseño.



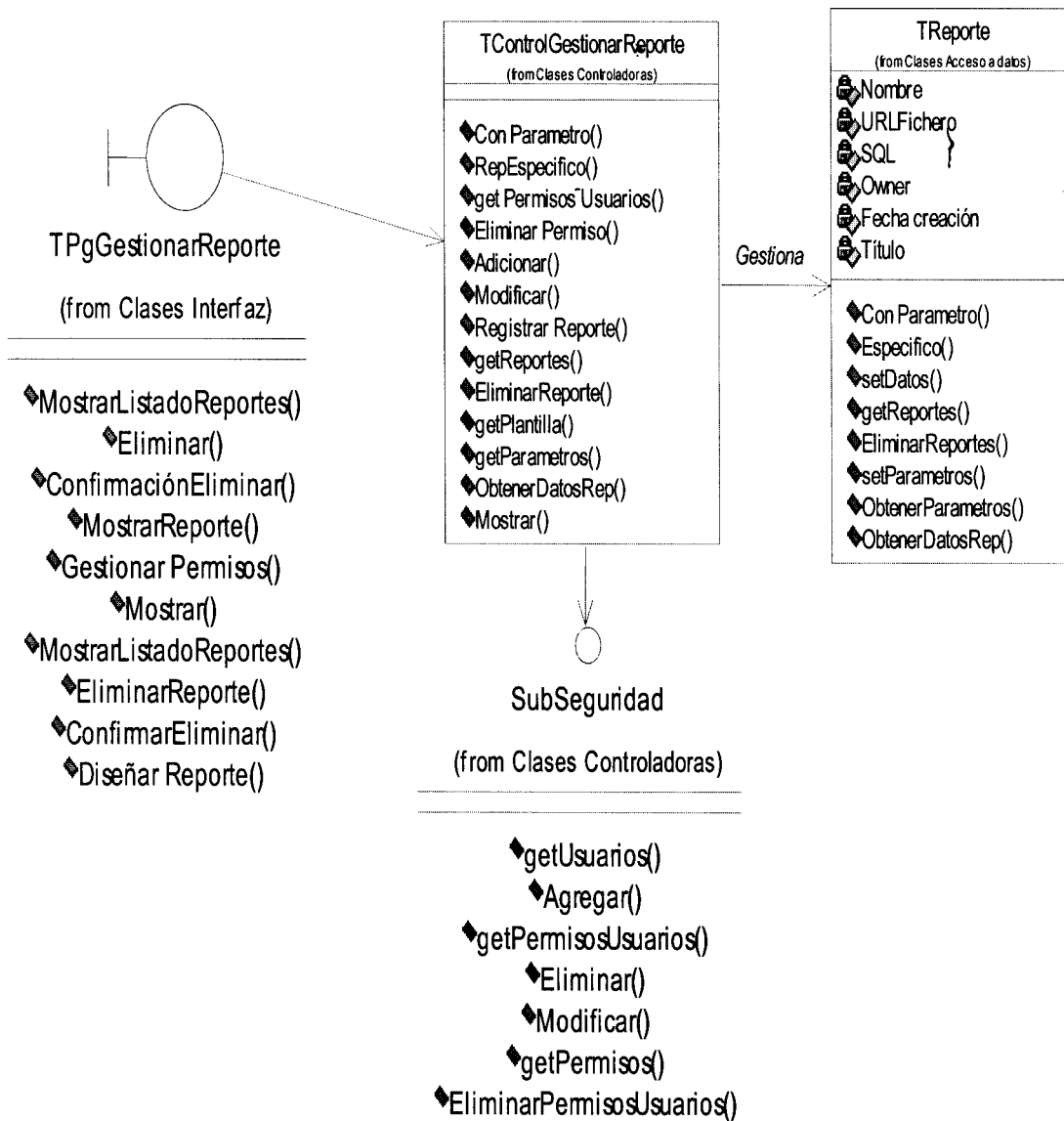
Diseñar Reporte



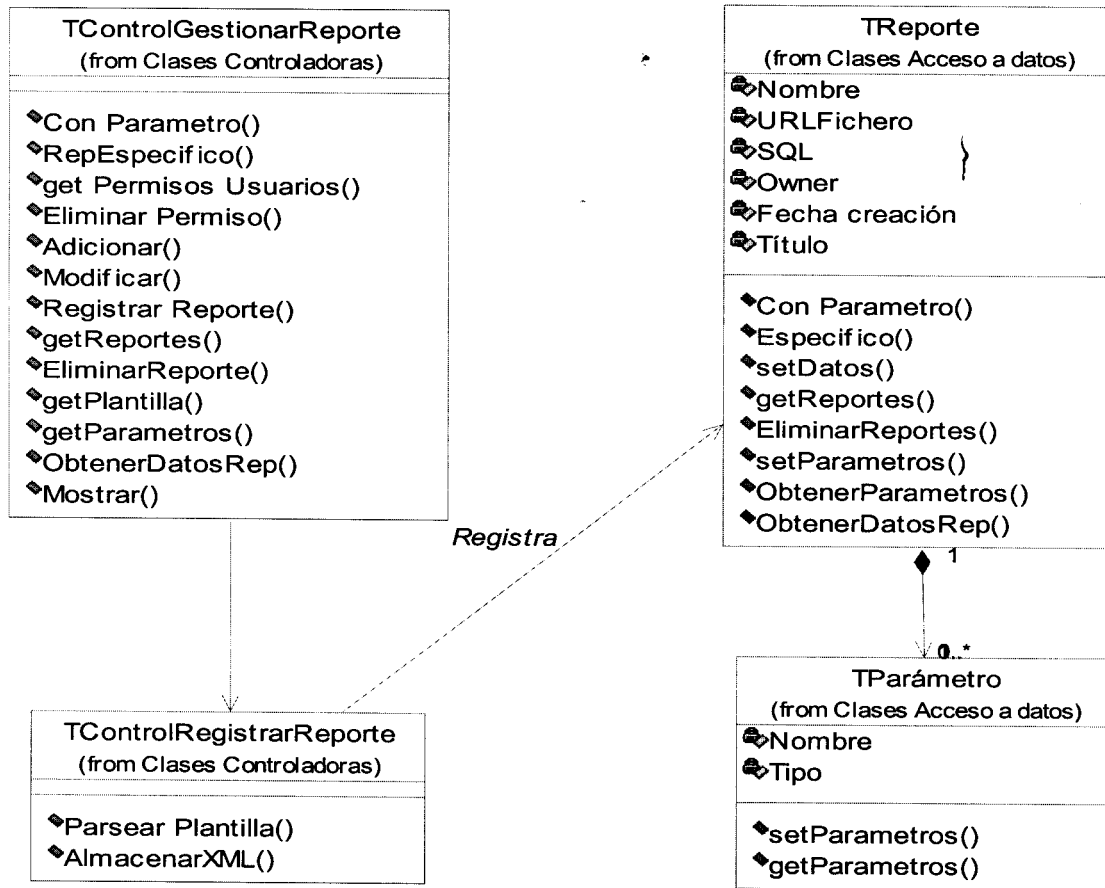
Gestionar Permisos para los Reportes



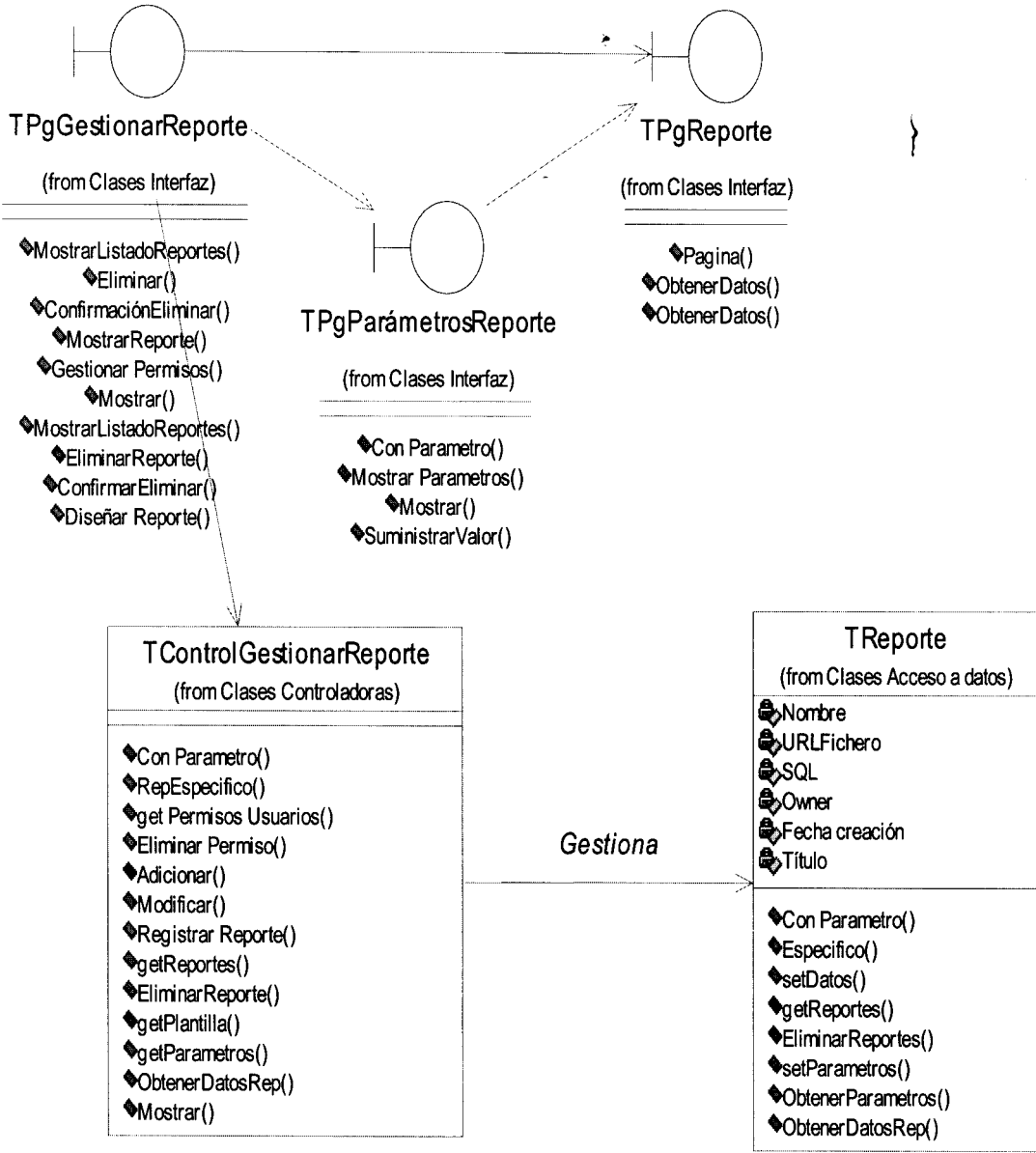
Gestionar Reporte

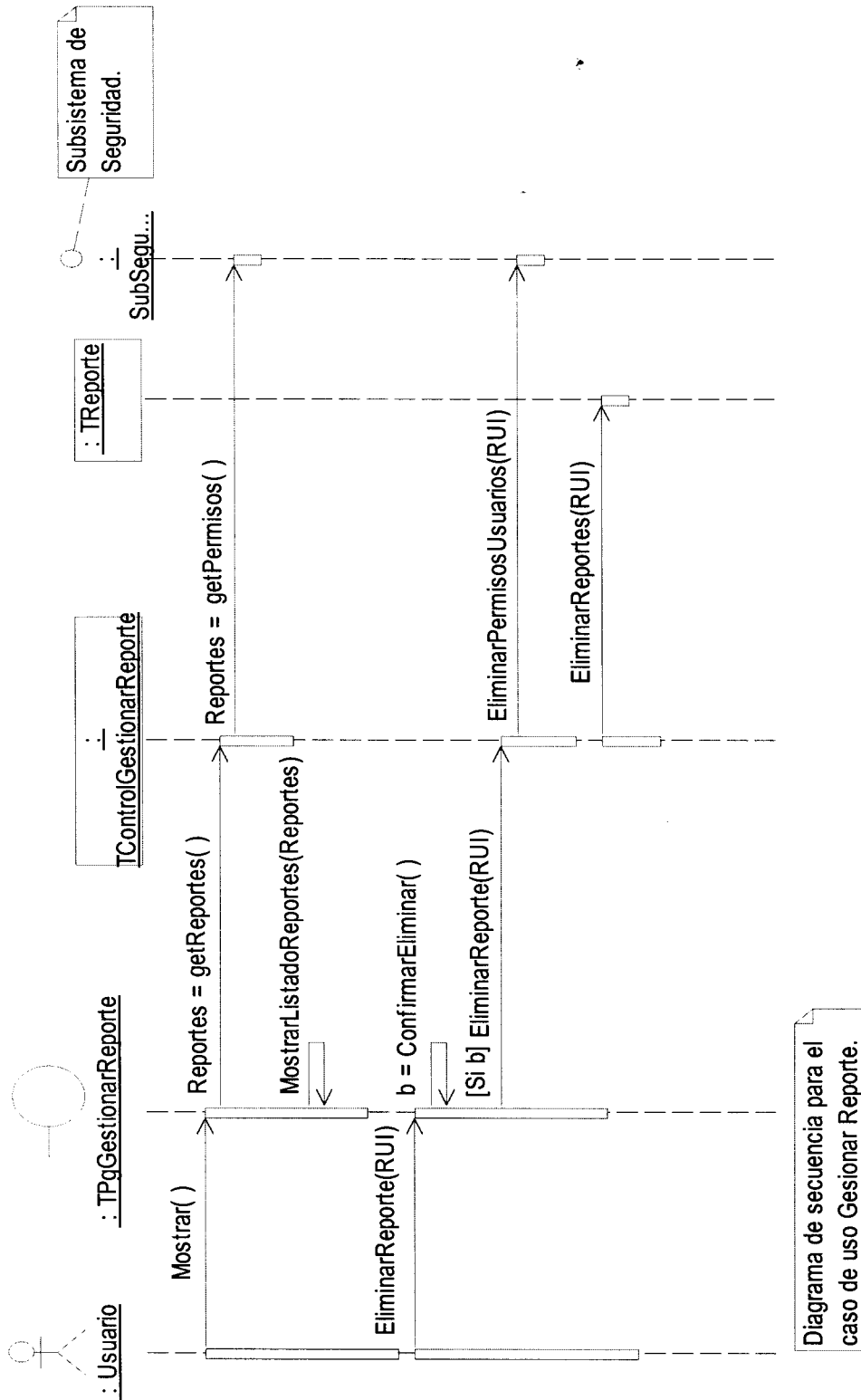


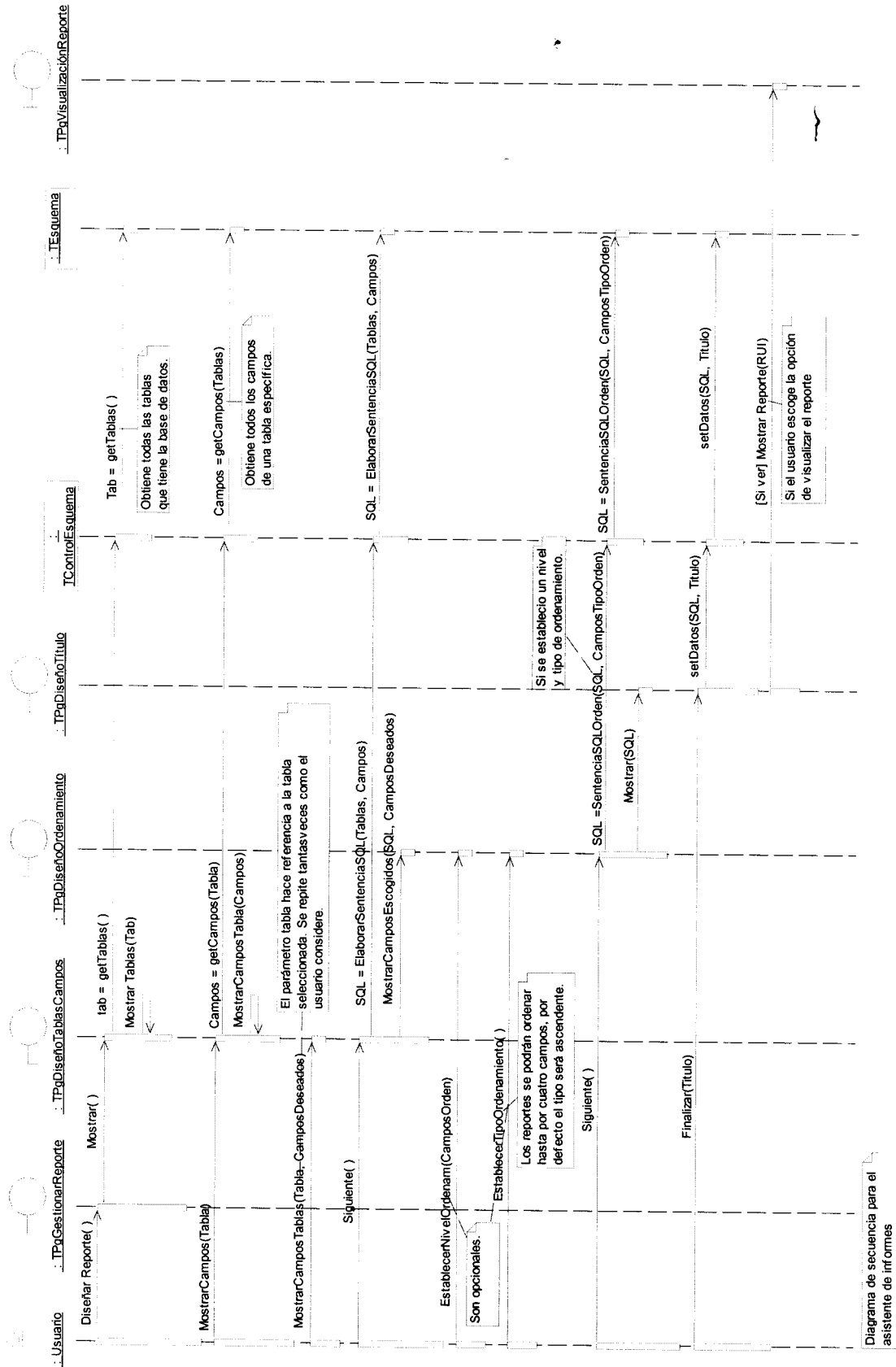
Registrar reporte

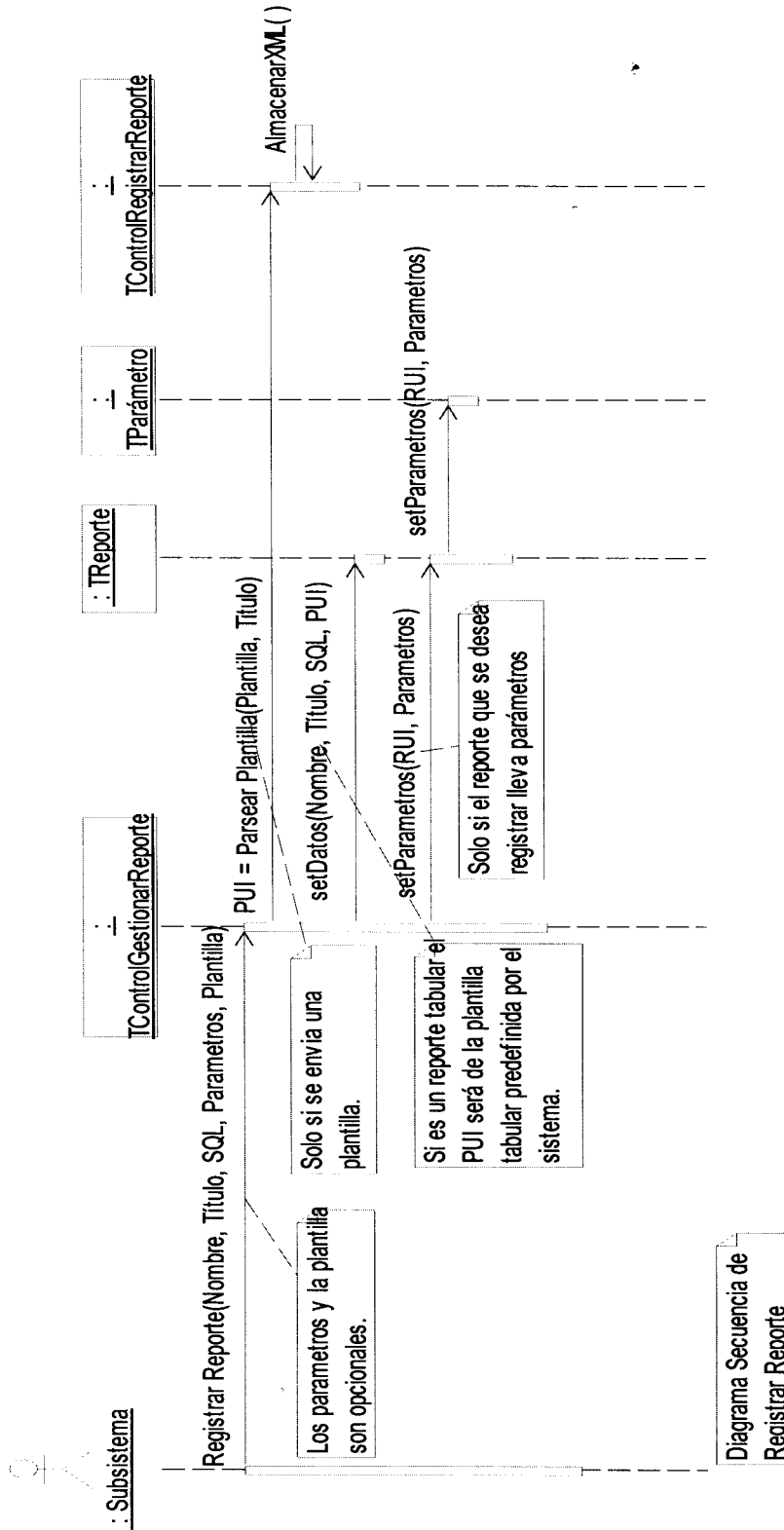


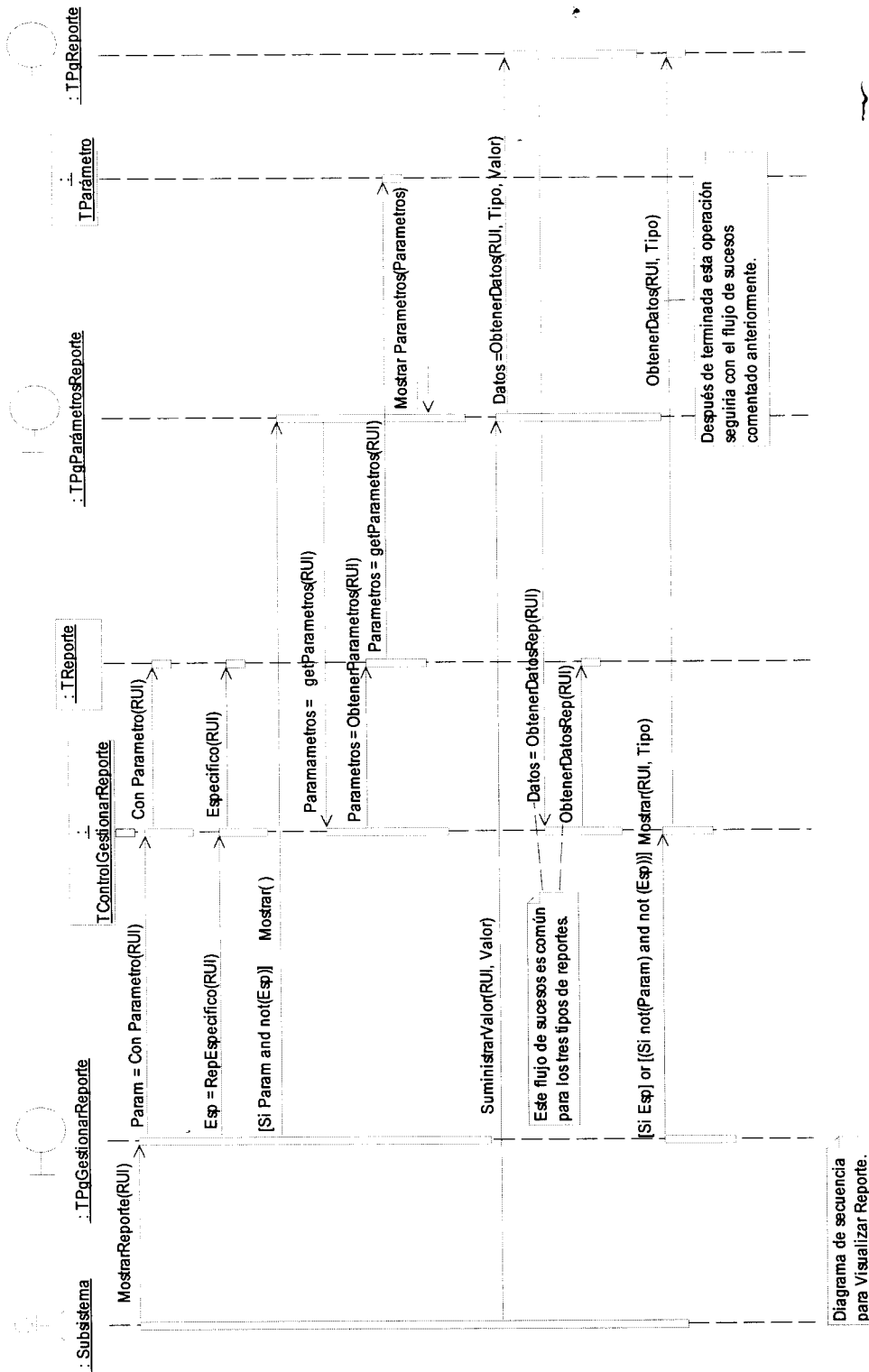
Visualizar Reporte











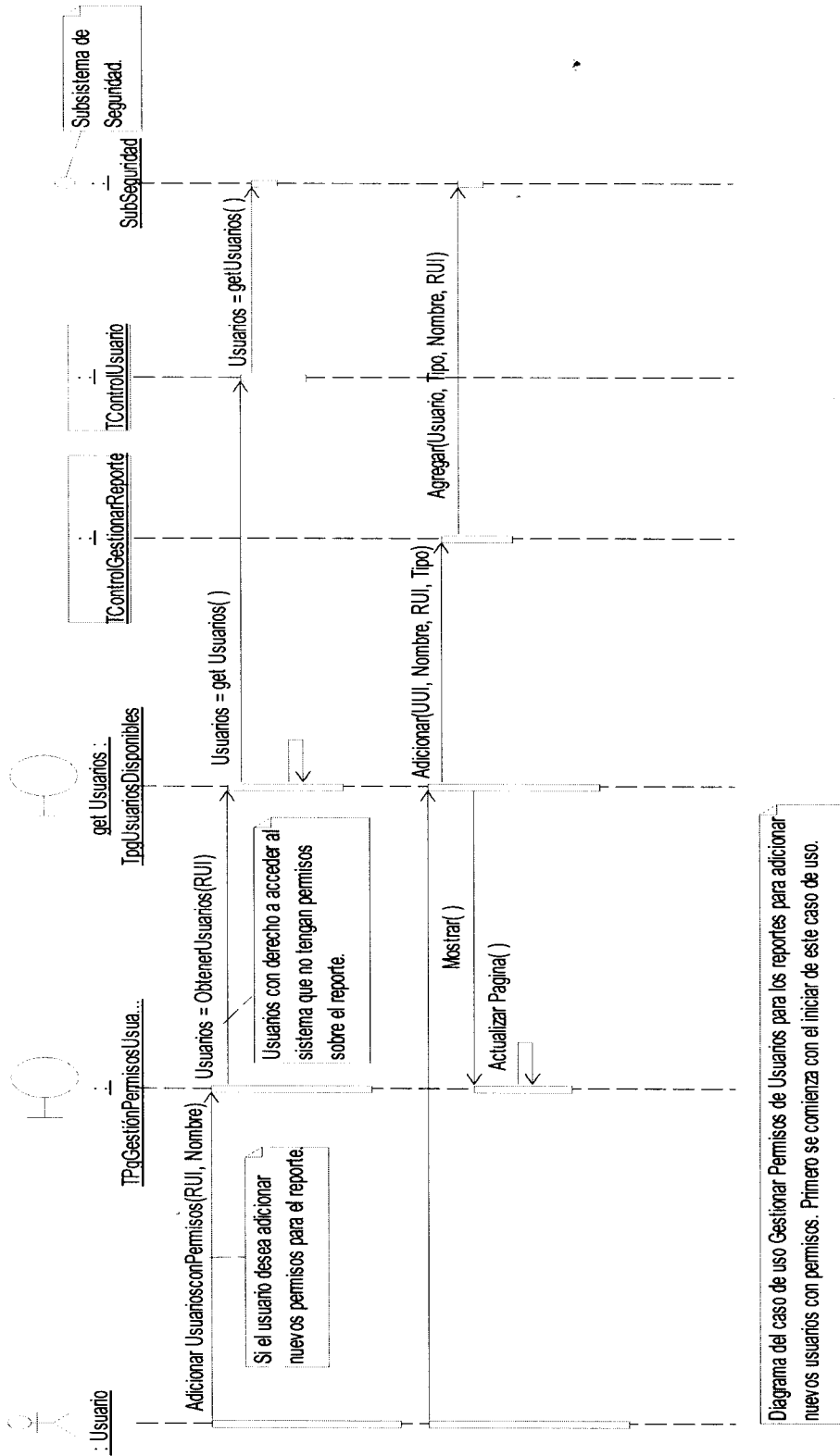


Diagrama del caso de uso Gestionar Permisos de Usuarios para los reportes para adicionar nuevos usuarios con permisos. Primero se comienza con el iniciar de este caso de uso.

