

Instituto Superior Politécnico “José Antonio Echeverría”

**Facultad Industrial
Ingeniería Informática**



**SISTEMA DE GESTIÓN ACADÉMICA: MÓDULO DE
DISEÑO Y DISTRIBUCIÓN DE REPORTE**

Trabajo para optar por el título de Ingeniería Informática

AUTOR: DIAILEN DE ARMAS MEDINA

TUTOR: ING. EMIL LIMA VALDÉS

Ciudad de La Habana

Junio del 2005

DEDICATORIA

*A Jesús, mi fiel amigo,
A quienes me guardan con sus oraciones, mis padres,
A mi nueva familia, mi amado esposo y mi hija.*

AGRADECIMIENTOS

♥ A mi Dios, una vez más compruebo que no tienes limitaciones, cuando todo parece imposible para los hombres, llegas tú y haces cosas increíbles. Gracias Papá por haberme permitido llegar hasta el final.

♥ A mi esposo, gracias mi amor por hacerme reír cuando más lo necesito, por estar siempre a mi lado, por tu cuidado y porque sé que me amas. Gracias porque cuando estoy a solas con Dios me haces decir palabras como...qué lindo es tener una familia.

♥ A mis padres, tengo que agradecerles todo. Han sido mi inspiración para seguir adelante en esta larga carrera, reciban este regalo porque ha sido por ustedes. Gracias por todo el tiempo que me han dedicado desde pequeña, por su apoyo incondicional, su optimismo, su desvelo, su confianza en mí y por todos los trabajos que me hicieron y que me ayudaron a hacer. Los admiro y los amo siempre.

♥ A mi niña, que se ha portado tan bien dentro de mí y que ha soportado todo este meche.

♥ A mi hermana Mirelys, por ayudarme ahora que es cuando más lo necesito, a Marcos por su disposición y a toda mi familia, yo sé que todos me quieren.

♥ A mis hermanos en la fe, ustedes tienen un lugar importante en todo esto. Puedo asegurar que he resistido gracias a que sus oraciones me han sustentado.

♥ A mi tutor, Emil, por haberme ayudado y guiado en la tesis; realmente lo admiro profe. También a Anaisa y la gente de la UCI, ellos saben por qué.

♥ A Tania, Yircy y todos mis amigos del aula, gracias por tantos momento lindos que me regalaron en estos cinco años.

♥ A la gente del cuarto, por soportarme todo este tiempo. He disfrutado mucho con ustedes en estos finales. Las voy a extrañar.

♥ A todas las persona que se han preocupado por mí y han contribuido en mi formación profesional y personal,

Mis Agradecimientos

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Centro de Estudios de Ingeniería de Sistemas (CEIS) y a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los 15 días del mes de junio del 2005.

Firma del Autor

Firma del Tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado Sistema de Gestión Académica: Módulo de Diseño y Distribución de Reportes, fue realizado en la Universidad de las Ciencias Informáticas (UCI). Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

- Totalmente
- Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

- Ahorro de tiempo en la elaboración y distribución de informes.
- Obtención de la información adecuada en un momento determinado.
- Mejora de la calidad de los servicios prestados en cuanto a tiempo de respuesta, procesamiento de datos y centralización y organización de la información
- Seguridad de la información contenida en los reportes.
- Posibilidad de hacer investigaciones de tipo social.

Como resultado de la implantación de este trabajo se reporta un efecto económico que asciende a 1995 dólares.

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____

Representante de la entidad

Cargo

Firma

Cuño

RESUMEN

Debido al dinamismo de los negocios que forman parte de la gestión académica, no es posible prever todas las necesidades de información que tiene cada trabajador relacionado con este aspecto en cualquier centro educativo. Actualmente, es constante el flujo de datos de este tipo que existe en la Universidad de Ciencias Informáticas, y como no se cuenta con un sistema que satisfaga estas necesidades completamente, todo el proceso de obtenerlos y procesarlos se retrasa en gran medida, se torna engorroso y hasta inseguro. Por esta razón, se hace necesario desarrollar el “Módulo de Diseño y Distribución de Reportes” del sistema Akademos. Tiene como objetivo poner a disposición del usuario herramientas que le permitan crear reportes personalizados que se ajusten a sus necesidades de información, exonerándolos de la dependencia a un conjunto de reportes fijos, que puede, no lo satisfagan en un momento determinado. Además podrá hacer llegar la información automáticamente a los diferentes trabajadores que la soliciten, con sólo planificar una vez la distribución de la misma. De esta manera se reduce el costo de elaboración y distribución de informes y aumenta la satisfacción del usuario, pues obtiene en tiempo y forma los datos que realmente necesita. Para su desarrollo se siguieron los pasos que propone el Proceso Unificado de Software. Su implementación es sobre la plataforma Microsoft .NET, específicamente en el lenguaje C#.NET.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO I FUNDAMENTACIÓN DEL TEMA.....	6
1.1 Introducción	6
1.2 La gestión académica en la UCI	6
1.2.1 Estrategia y misión de la UCI.....	6
1.2.2 La obtención de información académica. Análisis crítico.....	7
1.3 Módulo de Reportes del Sistema de Gestión Académica Akademos 1.0.....	7
1.4 Sistemas de Gestión Académica	8
1.4.1 Descripción General.....	8
1.4.2 Análisis comparativo de otras soluciones con la propuesta.....	9
1.4.2.1 GestAcad.....	9
1.4.2.2 Sistema del Corte, Sistema de Asistencia y Sistema de Evaluaciones	9
1.4.2.3 Sistema de Gestión Académica: Akademos	11
1.5 La Obtención y Distribución de Información	14
1.5.1 Visualización de Datos.....	14
1.5.2 Asistentes de Reportes y Gráficos.....	15
1.5.3 La distribución de información mediante el correo electrónico	16
1.6 Diseño de aplicaciones WEB.....	17
1.6.1 El uso del XML	17
1.6.2 La tecnología .NET	18
1.6.2.1 Arquitectura de aplicaciones .NET	21
1.7 Fundamentación de la metodología utilizada	27
1.7.1 Rational Unified Process™ (RUP)	27
1.7.2 Unified Modeling Language(UML)	29
1.8 Lenguaje, Gestor de Base de Datos y Software utilizado.....	29
1.8.1 C#.....	29
1.8.2 ASP.NET.....	32
1.8.3 Visual Studio.NET 2003	33
1.8.4 SQL Server 2000	33
1.9 Conclusiones	35
CAPÍTULO II DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA	36
2.1 Introducción	36
2.2 Modelo del Dominio	36
2.2.1 Definición de los conceptos principales	36
2.2.2 Diagrama de Clases del Modelo de Objetos.....	37
2.3 Requisitos funcionales	37
2.4 Requisitos no funcionales	41
2.5 Beneficios tangibles e intangibles.....	44
2.6 Análisis de costos y beneficios	45
2.7 Descripción del sistema propuesta	46
2.7.1 Concepción general del sistema	46
2.7.2 Modelo de Casos de Uso del Sistema	48
2.7.2.1 Diagrama de Casos de Uso del Sistema	48
2.7.2.2 Descripción general de los Casos de Uso no detallados.....	49
2.7.2.3 Descripción detallada de los Casos de Uso.....	50
2.8 Conclusiones	62
CAPÍTULO III CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA	63
3.1 Introducción	63
3.2 Diagramas de Clases y de Interacción	64

3.2.1	Paquete: Crear Reporte de Listado	64
3.2.1.1	Diagrama de Clase: Crear Reporte de Listado	64
3.2.1.2	Diagrama de Secuencia: Crear Reporte de Listado.....	65
3.2.2	Paquete: Crear Reporte Cruzado	67
3.2.2.1	Diagrama de Clases: Crear Reporte Cruzado	67
3.2.2.2	Diagrama de Secuencia: Crear Reporte Cruzado.....	68
3.2.3	Paquete: Crear Gráfico	70
3.2.3.1	Diagrama de Clases: Crear Gráfico	70
3.2.3.2	Diagrama de Secuencia: Crear Gráfico	71
3.2.4	Paquete: Conformar Filtro.....	74
3.2.4.1	Diagrama de Clases: Conformar Filtro.....	74
3.2.4.2	Diagrama de Secuencia: Conformar Filtro	75
3.2.4.3	Diagrama de Secuencia: Contenido Filtro.....	77
3.2.5	Paquete: Actualizar Carpetas	78
3.2.5.1	Diagrama de Clases: Actualizar Carpetas	78
3.2.5.2	Diagrama de Secuencia: Actualizar Carpetas_Crear.....	79
3.2.5.3	Diagrama de Secuencia: ActualizarCarpetas_Mover.....	80
3.2.5.4	Diagrama de Secuencia: ActualizarCarpetas_Eliminar.....	81
3.2.6	Paquete: Actualizar Reportes	82
3.2.6.1	Diagrama de Clases: Actualizar Reportes	82
3.2.6.2	Diagrama de Secuencia: Actualizar Reportes_Modificar	83
3.2.6.3	Diagrama de Secuencia: ActualizarReportes_Eliminar.....	84
3.2.7	Paquete: Planificar Distribución	85
3.2.7.1	Diagrama de Clases: Planificar Distribución	85
3.2.7.2	Diagrama de Secuencia: Planificar Distribución	86
3.2.8	Paquete: Consultar Reporte.....	87
3.2.8.1	Diagrama de Clases: Consultar Reporte.....	87
3.2.8.2	Diagrama de Secuencia: Consultar Reporte_Graficar	88
3.2.8.3	Diagrama de Secuencia: Sección Graficar	90
3.3	Diseño de la Base de Datos	91
3.3.1	Diagrama de Clases Persistentes.....	91
3.3.2	Modelo de Datos	92
3.4	Principios de Diseño	93
3.4.1	Estándares en la interfaz de la aplicación.....	93
3.4.2	Concepción general de la ayuda.....	94
3.4.3	Tratamiento de excepciones	94
3.5	Estándares de Codificación	94
3.6	Modelo de Despliegue	97
3.6.1	Diagrama de Despliegue.....	97
	Estructura de Nodos	98
3.6.2	Arquitectura de Capas	98
3.7	Modelo de Implementación.....	99
3.7.1	Diagrama de Paquetes de Componentes	99
3.7.1.1	Diagrama de Componentes del Paquete Presentación	100
3.7.1.2	Diagrama de Componentes del Paquete Negocio	102
3.7.1.3	Diagrama de Componentes del Paquete Acceso a Datos	103
3.7.2	Descripción de los componentes	104
3.8	Conclusiones	109
	CONCLUSIONES.....	110
	RECOMENDACIONES.....	111
	REFERENCIAS BIBLIOGRÁFICAS	112

BIBLIOGRAFÍA.....	113
GLOSARIO DE SIGLAS Y TÉRMINOS.....	114
ANEXOS.....	116

INTRODUCCIÓN

En toda empresa o institución 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, que no sólo exige que contenga la información que se requiere en un momento dado, sino también que aparezca con un formato coherente, que haga el informe fácil de leer y estructure la información que representa de manera ordenada y lógica.

La Universidad de Ciencias Informáticas no está exenta de esta situación, a diario se genera información en las aulas, departamentos, facultades, que es necesario procesar, resumir y obtener en un momento determinado.

Debido a lo antes mencionado y al carácter dinámico que tienen los procesos académicos en cualquier centro educativo, existe una gran demanda de información de este tipo por parte de los diferentes niveles de la universidad (rector, vicerrector de formación, secretaria general). Constantemente se están solicitando reportes que pueden ser referente a la matrícula, la asistencia, las evaluaciones diarias, parciales o finales e incluso a la planificación de los profesores, evidentemente se hacen extensivos a facultades, departamentos y grupos académicos, y lo que es más importante a tener en cuenta, no siempre son los mismos. Todos estos datos que se generan y de los cuales se requiere información, se procesan manualmente, primero por grupos, después por facultades hasta que llega a manos del más alto nivel que lo solicitó.

La unión de todos estos elementos conlleva a que no siempre se tenga predeterminado la forma de generar un informe, además de que se incurre en un gasto de tiempo enorme y un gran riesgo de introducción de errores en los mismos. En la mayoría de las ocasiones la información no llega al usuario en tiempo y forma, atentando contra la calidad de la universidad ya que no se pueden concentrar los esfuerzos en las tareas de análisis, control y propuesta de mejoras. Estas consecuencias no han sido más que el resultado del siguiente problema:

¿Cómo satisfacer las necesidades de información de los involucrados en el proceso de la gestión académica?

Se hace evidente la necesidad de un sistema que sea capaz de proveer a los diferentes usuarios de un servicio de reportes que cubra sus necesidades de información en el momento preciso y que le permita además distribuirlo con rapidez a todo aquel que se lo solicite.

La mayoría de los sistemas informáticos actuales proveen salidas de información más o menos avanzadas a través de informes, esto se debe 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 esta razón se ha decidido crear asistentes de reportes que le faciliten al usuario poder construir sus propias consultas para generar informes y exportar los datos a otros formatos como Microsoft Excel donde se le pueda dar formato a su gusto.

Existe una primera versión de este sistema, que es el Módulo de Reportes del Sistema de Gestión Académica: Akademos, sin embargo es insuficiente en cuanto a las opciones que necesita tener el usuario y en cuanto a la información que maneja. Permite generar reportes sólo con datos pertenecientes a la matrícula actual que existe en la universidad y la única opción que tiene es la de ver el reporte, sin darle la posibilidad de almacenarlo, imprimirlo, modificarlo o distribuirlo.

Al concluir el Sistema de Diseño y Distribución de Reportes se espera sea capaz de ofrecer informes con los datos pertenecientes a evaluaciones, asistencia y planificación de profesores, además de garantizar una buena seguridad a los mismos, permitiendo que el acceso a ellos sea sólo por parte de las personas que tienen permiso a consultarlos, por lo que se espera haya una opción que le permita al usuario que está diseñando un reporte, determinar qué estructuras o personas tienen este permiso. Esto conlleva a que el sistema permita almacenar, modificar y eliminar los informes que se generen, además de brindar la opción de imprimirlo o exportarlo a Microsoft Excel o Word. Por otra parte debe garantizar la distribución de la información contenida en los reportes a las personas o estructuras que lo soliciten y enriquecerla con gráficos creados por el usuario a partir de dichos reportes.

Para dar cumplimiento a estas exigencias se han trazado los siguientes objetivos:

Objetivo General:

Desarrollar un sistema automatizado que permita a los usuarios crear y distribuir de forma inmediata, reportes asociados a la información manejada por el sistema de gestión académica: Akademos.

Objetivos Específicos:

- Crear un asistente que guíe al usuario a elaborar reportes de listado de estudiantes.
- Crear un asistente que guíe al usuario a elaborar reportes de totales o cruzados.
- Diseñar un asistente que guíe al usuario a elaborar reportes de las evaluaciones.
- Diseñar una herramienta que le permita al usuario planificar la distribución de los reportes por los diferentes niveles de la universidad.
- Diseñar un asistente para que el usuario grafique de manera personalizada la información de los reportes.
- Permitir al usuario restringir el acceso a los reportes creados por él.

Con el fin de dar cumplimiento a los objetivos antes mencionados es necesario desarrollar un conjunto de tareas que se mencionan a continuación:

- Entrevistar a los diferentes tipos de usuario que harán uso del sistema.
- Estudiar las opciones generales que brindan los asistentes de reportes.
- Analizar las opciones que brinda SQL Server 2000.
- Investigar y estudiar los Sistemas de Gestión Académica existentes.
- Buscar componentes capaces de representar gráficos de pastel, de barra y de línea.
- Estudiar las opciones generales que brindan los asistentes de gráficos.
- Analizar el organigrama de la UCI.
- Analizar la política seguida en la UCI en cuanto a la seguridad de los usuarios.

- Hacer el análisis y diseño del Módulo de Diseño y Distribución de Reportes del Sistema de Gestión Académica: Akademos.
- Implementación del Módulo de Diseño y Distribución de Reportes del Sistema de Gestión Académica: Akademos.
- Redacción del informe que contiene la documentación del sistema.

El objeto de estudio de este trabajo lo constituye la Universidad de Ciencias Informáticas y los sistemas de gestión académica.

La Universidad de Ciencias Informáticas (UCI) es la más joven del país, creada para formar un personal altamente calificado en ingeniería informática, que contribuya en gran medida al desarrollo del mismo. En ella se ha comenzado la tarea de automatizar todos los procesos de la universidad, dentro de los cuales se encuentra la gestión académica.

Los sistemas de gestión académica involucran todos los procesos de la docencia, almacena y procesa la información relacionada con asistencia, evaluaciones, planes de estudio, matrícula, profesores. Estos procesos son de naturaleza dinámica, pues deben adaptarse a las condiciones en que se desarrolla la labor formativa, además involucran gran cantidad de personas, las cuales se relacionan directamente en la gestión automatizada en aras de aumentar la eficiencia.

Dentro de este entorno se encuentra el campo de acción de la investigación llevada a cabo para el desarrollo del sistema que se presenta y es precisamente la obtención y distribución de información a través de Sistemas de Diseño y Distribución de Reportes. Este tipo de aplicaciones se encargan de facilitar el acceso del usuario a la información difícil de obtener de manera inmediata, es decir la información que requiere de procesos de cálculo o de un proceso engorroso de búsqueda dentro de todos los datos de un sistema de gestión.

Toda la información que contiene este documento se encuentra distribuida en tres capítulos.

El primero, titulado Fundamentación del Tema, describe cómo se lleva a cabo la obtención y distribución de información académica en la UCI y los sistemas de gestión

que se han usado anteriormente con este fin. Por otra parte se explican las tecnologías actuales que se emplearán en el desarrollo del sistema y se fundamenta la metodología, lenguaje, gestor de base de datos y software utilizados en el mismo.

En el capítulo 2: Descripción de la Solución Propuesta, aparece reflejado el Modelo del Dominio, así como los requisitos funcionales y no funcionales. Se hace el análisis de costo beneficio del sistema, partiendo de los beneficios tangibles e intangibles que este aporta y que también se mencionan. Por último, se describe el sistema propuesta mediante el Modelo de Casos de Uso del Sistema.

El tercer y último capítulo: Construcción de la Solución Propuesta, contiene los diagramas de clases y de interacción de cada Caso de Uso detallado en el capítulo anterior, el diseño de la Base de Datos que almacenará la información del sistema, el Modelo de Despliegue y de Implementación. También se explican los principios de diseño y estándares de codificación presentes en la aplicación.

CAPÍTULO I FUNDAMENTACIÓN DEL TEMA

1.1 Introducción

En el presente capítulo se describen los conceptos fundamentales asociados a la gestión académica y los problemas que presenta en el entorno en que se desarrolla el Módulo de Diseño y Distribución de Reportes, planteando una propuesta de solución basada en algunas tendencias y tecnologías actuales que también se abordan en esta sección. Por otra parte se fundamenta el uso del lenguaje C# en Visual Studio .Net y el gestor de base de datos SQL Server 2000.

1.2 La gestión académica en la UCI

1.2.1 Estrategia y misión de la UCI

La Universidad de las Ciencias Informáticas (UCI) “tiene la misión de contribuir e impulsar la informatización del país y la industria cubana del software” [1]. Para lograrlo, centra sus esfuerzos en la formación de un personal altamente calificado en ingeniería informática con un indeclinable y firme compromiso con la Patria y la Revolución, capaz de apoyar en gran medida el desarrollo de ese campo y de la economía del país.

En estos momentos se está desarrollando en esta institución el Proyecto UCI-Ciudad Digital, que persigue la creación de una micro sociedad completamente informatizada, que pueda 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 gestión académica, que como bien su nombre lo indica está enfocada al desarrollo de la automatización del área académica de la institución.

1.2.2 La obtención de información académica. Análisis crítico

Actualmente todas las necesidades de información que surgen en los diferentes niveles de la universidad se solicitan del nivel superior al inferior.

La Secretaria General solicita a las secretarías docentes de cada facultad la información referente a matrícula y a promoción y estas las solicitan a cada profesor perteneciente a los diferentes grupos y la entrega al usuario final (Secretaría General).

EL Vicerrector de Formación Académica solicita al Decano de Formación Académica reportes de asistencia y evaluaciones, este a los decanos y estos a los vicedecanos, los que a su vez solicitan la información a los profesores de cada asignatura y a los profesores guías de cada grupo respectivamente, y fluye la información procesada en sentido contrario hasta llegar al usuario final (Vicerrector de Formación Académica). También el Vicerrector de Formación Académica solicita a los vicedecanos la información referente a la planificación de los profesores, en cuanto a los horarios de clase en cada grupo.

De todo el flujo de información descrito anteriormente sólo se encuentra automatizado el referente a los datos de la matrícula a través del Módulo de Reportes del Sistema de Gestión Académica Akademos. Luego, al tener que acumular y procesar toda esta información manualmente se incurre en un gran costo de elaboración de informes, lo que trae como consecuencia que no se puedan concentrar los esfuerzos en las tareas de análisis, control y propuestas de mejora. Además la frecuencia y disponibilidad de la información es muy baja en cantidad y calidad, por lo que en ocasiones no llega en el momento más oportuno. Por otra parte, no se logra la máxima autonomía técnica para diseñar informes complejos, tan necesaria en esta sociedad donde el tiempo es un recurso muypreciado.

1.3 Módulo de Reportes del Sistema de Gestión Académica Akademos 1.0

El Módulo de Reportes de Akademos es una primera versión del sistema que se propone. Permite al usuario diseñar reportes referentes a la información contenida en la hoja de

matrícula y en la de prematrícula y le brinda la opción de seleccionar qué tipo de reporte desea: de listado de estudiantes o de totales.

Como resultado de crear un reporte de listado se obtiene precisamente una lista de alumnos y sus datos respectivos. Luego el usuario tiene la posibilidad de indicar los alumnos que conforman su lista y los datos que desea consultar.

Por otra parte el resultado de crear reportes de totales es una tabla, cuyas filas son los distintos valores que puede tomar un dato determinado y las columnas los valores que puede tomar otro dato, las casillas del cuerpo de la tabla muestran números que representan el total de alumnos que cumplen con uno y otro valor respectivamente. Para esto, el usuario tiene la opción de escoger qué datos desea que se muestren en su tabla.

Evidentemente esta es una versión insuficiente respecto a la ya explicada necesidad de información que existe en cualquier centro educativo, ya que pasa por alto la asistencia, las evaluaciones, los datos de los profesores en cuanto a su planificación. Además presenta limitaciones a la hora de diseñar los reportes, pues no existe la manera de crear una expresión lógica que filtre la información, lo que trae como consecuencia que no se obtenga con exactitud el informe solicitado.

1.4 Sistemas de Gestión Académica

1.4.1 Descripción General

La gestión académica constituye un elemento de gran importancia dentro de un centro educacional, ya que con esta se contribuye a definir el rumbo del mismo, se realiza el diseño, implantación y operación de las decisiones referentes al proceso docente, se evalúan los programas académicos y se asegura su calidad.

Los sistemas que se encargan de la gestión académica almacenan y procesan la información del área docente de los centros educacionales. Por lo general abarcan todo lo relacionado con las pruebas de acceso al centro, la matrícula y prematrícula de los estudiantes, el control de las evaluaciones y la asistencia de cada uno de ellos, sus expedientes y los títulos otorgados. Por otra parte, se incluye la gestión de los planes de estudio, las becas, el horario de clases y exámenes y los datos de los profesores.

De acuerdo a lo antes descrito se llega a la conclusión de que estos sistemas constituyen herramientas que le permiten a los centros educacionales adquirir ventajas sostenibles, haciendo eficientes sus procesos, optimizando sus recursos e integrando la información de la organización.

1.4.2 Análisis comparativo de otras soluciones con la propuesta

1.4.2.1 GestAcad

GESTACAD es un sistema informativo automatizado, desarrollado por el Departamento de Informática de la Universidad de Matanzas "Camilo Cienfuegos". Permite actualizar y mantener la información sobre estudiantes y profesores de una Universidad y obtener determinados resultados propios del trabajo de las áreas implicadas, aunque el grueso de las informaciones se obtiene mediante el acceso al sitio WEB de la Universidad.

Los objetivos del trabajo a partir de lo anterior han sido:

- Obtener un diseño de una base de datos que permite obtener las informaciones académicas necesarias para el perfeccionamiento de la gestión universitaria, incluyendo los datos docentes de alumnos, asignaturas, profesores, etc.
- Desarrollar una herramienta de fácil manejo para el personal de secretaría docente de las facultades y otros que se determinen (en la medida que el trabajo abarque otros aspectos).
- Desarrollar un ambiente WEB que permita obtener resultados de consultas desde cualquier lugar de la red universitaria en cualquier momento de tiempo, con acceso para todas las personas que se determine. [2]

1.4.2.2 Sistema del Corte, Sistema de Asistencia y Sistema de Evaluaciones

Los Sistemas del Corte, de Asistencia y de Evaluaciones son sistemas creados por la Dirección de Informatización de la UCI en el año 2003-2004 con el objetivo de garantizar la gestión académica de la universidad y satisfacer las necesidades de información de sus

usuarios, ya que los reportes brindados por GESTACAD no eran suficientes para el control y la toma de decisiones en el centro.

Los Sistemas del Corte y de Asistencia permiten brindan reportes que pueden ser consultados a nivel de facultad, año, grupo, provincia, municipio, asignatura, sexo, raza, vía de ingreso y procedencia.

El Sistema del Corte muestra la cantidad de estudiantes que hay para cada uno de los tipos de evaluaciones (Bien, Regular y Mal), y a partir de este reporte se puede obtener el listado de estos estudiantes con sus evaluaciones por asignaturas y su evaluación final. El Sistema de Asistencia cuenta con estos mismos reportes, sólo que en lugar de las evaluaciones por asignaturas, muestra la cantidad de ausencias por asignaturas.

Por su parte el Sistema de Evaluaciones pone a disposición del usuario los siguientes reportes:

- Cantidad de estudiantes sin mundial por facultad.
- Cantidad de estudiantes sin mundial por provincia.
- Listado de estudiantes por grupo y facultad que no tienen evaluaciones insertadas en las asignaturas obligatorias.
- Listado de estudiantes con 5 en todas las asignaturas.
- Listado de estudiantes con 4 y/o 5 en todas las asignaturas.
- Listado de estudiantes con 1 asignatura suspensa.
- Listado de estudiantes con 2 asignaturas suspensas.
- Listado de estudiantes con 3 o más asignaturas suspensas.

En el caso de los tres últimos reportes también se muestran cuáles son las asignaturas que tienen suspensas.

Como se puede apreciar estos sistemas brindan una información muy limitada del área docente a los usuarios, haciéndose evidente el desarrollo de un sistema de gestión académica más flexible a los cambios y a la inserción de nuevos reportes.

1.4.2.3 Sistema de Gestión Académica: Akademos

Sin dudas, los procesos que abarca la gestión académica son de carácter dinámico, ya que tienen que ajustarse a las condiciones en que se desarrolla la labor formativa. Además involucran gran cantidad de actores que es necesario interactúen directamente con el sistema encargado de la gestión académica, para así lograr una mayor eficiencia.

Los sistemas de gestión académica que existen en la actualidad, incluyendo los antes descritos, que además fueron usados en la UCI, no dan solución a esta problemática totalmente. Sólo permiten la interacción con el sistema al personal de secretaría, recargando así su labor e impidiendo que el personal que produce la información sea el mismo que la introduzca. Además, no se pueden adaptar a los cambios que pueden surgir en los procesos docentes, ya que no permiten la configuración del sistema. Por último y siendo el problema fundamental que ocupa este trabajo, los sistemas hasta este momento existentes, sólo brindan reportes que han sido previamente definidos, es decir, está a disposición del usuario un conjunto de información que siempre es la misma. Evidentemente, esto no respalda la dinámica que se mencionaba anteriormente y atenta contra el buen funcionamiento de cualquier centro de estudios, ya que para lograr el buen desenvolvimiento de los procesos docentes es necesario que los directivos se tracen estrategias para planificar, organizar y controlar los recursos que intervienen en los mismos, para esto es de vital importancia la posibilidad de hacer disímiles e impredecibles consultas, a través de las cuales obtengan la información necesaria en ese momento para llevar a cabo una correcta toma de decisiones.

Debido a que estos sistemas no cubrieron las expectativas de los usuarios de la UCI, actualmente no existe ninguno que gestione la parte académica de la universidad. Por esta razón la Dirección de Informatización de la Universidad de las Ciencias Informáticas ha desarrollado Akademos, un sistema para la gestión académica cuyo objetivo es permitir la gestión automatizada de los elementos que intervienen en la labor académica de un centro de estudios, que pueda enfrentar los cambios de forma natural, adaptándose a las nuevas condiciones y formas de hacer con el menor costo.

Akademos cuenta con seis módulos que en su conjunto garantizan toda la gestión académica de la universidad:

Módulo de Matrícula: El objetivo fundamental del módulo de matrícula es el control de los datos de los estudiantes, así como la gestión de los movimientos a que estos son sometidos en su paso por la universidad. Este módulo permite la definición de los estados de los alumnos, las transiciones entre estos estados y los documentos que deben llenarse por el personal de secretaría para asentar los datos del movimiento, los cuales van a parar al expediente digital del estudiante. Este módulo es una gran fuente de información para el de Diseño y Distribución de Reportes, ya que contiene todos los datos de los estudiantes, excluyendo sólo los académicos.

Módulo de Registro del Profesor: es el encargado de permitir el control del desarrollo de un período académico y depende en gran manera del Módulo de Plan de Estudio, pues la información de las evaluaciones y las formas de calificación de las asignaturas pertenecientes a este último, son consultadas por los profesores para saber la manera correcta de introducir las notas en el registro. Además, de la información contenida en el registro se pueden diseñar disímiles reportes que permiten conocer el estado académico de un estudiante, un grupo o una escuela completa.

Módulo de Plan de Estudio: el plan de estudios es la entidad fundamental del sistema. Este módulo permite la definición de los diferentes perfiles y disciplinas en las que se agrupan las asignaturas y la modificación de los planes de estudio con la introducción de otras nuevas que pueden ser de diferentes tipos y para las cuales es posible definir su método de evaluación, la forma en que se califica cada una, documentos adjuntos que especifiquen los temas que abarca, etc.

Módulo de Planificación: este módulo se encarga de gestionar la planificación de la carga docente de los profesores y los horarios de la escuela. Permite a los diferentes directivos de la universidad mantener un control de la plantilla de los profesores y un equilibrio en la asignación de la tarea docente a cada uno. En cuanto a la gestión de los horarios, es válido aclarar que no constituye una herramienta de generación de horarios, sin embargo ayuda a la confección de los mismos, permitiendo su validación en cuanto al solapamiento de profesores y locales.

Módulo de Expediente: el módulo de expediente es un repositorio digital de los documentos de los estudiantes. Los expedientes pueden almacenar documentos basados en plantillas, así como otros de libre formato o generados por el propio sistema. Cada vez que se crea un documento se almacena junto con otros datos, como su fecha de creación y el autor, para facilitar su posterior localización. Este módulo permite la definición de plantillas para cada plan de estudio que pueden ser utilizadas para crear documentos para los expedientes de los estudiantes, estas definen la forma en que se verá el documento, así como los controles necesarios para su llenado. El módulo de expediente se encarga de garantizar el acceso a la información histórica de los estudiantes, poniéndola a disposición de todo el que lo necesite, dando la posibilidad de consultarla y modificarla en función de su responsabilidad.

Módulo de Seguridad: garantiza que Akademos sea un sistema seguro, atendiendo a tres niveles de seguridad. Seguridad de los datos, con la utilización de Microsoft SQL Server 2000, el cual es capaz de garantizar la seguridad tanto lógica como de integridad de los datos almacenados en la base de datos. Seguridad de los recursos lógicos: para esto se deja al sistema operativo, Windows en este caso, la responsabilidad de restringir el acceso a cada uno de los recursos lógicos de la aplicación como páginas Web, Servicios Web, archivos XML, etc. Seguridad de los elementos específicos del negocio que automatiza la aplicación: Akademos implementa varios niveles de acceso para restringir las acciones que puede realizar un usuario determinado y lleva un control sobre las mismas. Este sistema tiene gran importancia para el Módulo de Diseño y Distribución de Reportes ya que es el encargado de asegurar los reportes creados por los diseñadores.

Módulo de Diseño y Distribución de Reportes: la generación de reportes es una de las tareas más importantes de todo sistema de gestión de datos, tal es así, que la mayoría de estos quedan obsoletos por no comprender la trascendencia que puede tener no darle el enfoque correcto a este asunto. Es muy usual encontrarse que los sistemas se limitan a brindar un conjunto de reportes fijos, sin percatarse que con la evolución de un determinado proceso de negocio aparecen nuevas necesidades de información.

El módulo de Diseño y Distribución de Reportes, que además se presenta en este trabajo, constituye la solución que da Akademos a este problema. Permitirá al usuario diseñar,

distribuir y graficar reportes que involucran diferentes aristas de los estudiantes, pueden ser las notas y la proveniencia social, convirtiéndose en una poderosa herramienta para el estudio de la comunidad estudiantil.

Contará con asistentes de reportes que orientan al usuario paso a paso para lograr que este diseñe un reporte conforme a sus necesidades de información en un momento determinado, es decir, el usuario puede incluir en su informe datos provenientes de la mayor parte de los módulos de Akademos, si así lo desea. Por último este reporte queda registrado con el título, el subtítulo y el nombre que decida el diseñador, reflejándose también la persona que lo creó, de tal manera que pueda modificarlo cuando desee. Existe un asistente para crear reportes de listados de estudiantes y otro para crear reportes de totales o cruzados, cuyo resultado es una tabla que tiene por las filas un dato y por las columnas otro y las casillas del cuerpo de la tabla son los datos numéricos correspondientes al cruce de las filas y las columnas.

Además brindará la posibilidad de crear un gráfico a partir de un reporte, también con la ayuda de un asistente que permite seleccionar el tipo de gráfico, la información del reporte que se desea visualizar y cómo se desea que se muestre, quedando registrado de igual forma con el nombre, título y subtítulo que el autor decida.

Para llevar a cabo la distribución de los reportes, el sistema tendrá una herramienta que da la posibilidad de programar con antelación las personas a las que debe llegar, así como el momento: fecha, hora y frecuencia, en el caso de que se decida que la distribución sea periódica.

Este módulo garantizará a los diseñadores la seguridad de sus reportes, ya que estos pueden determinar las acciones que puede hacer cada persona sobre los mismos. Es necesario aclarar que para esta tarea se apoya en gran manera del Módulo de Seguridad.

1.5 La Obtención y Distribución de Información

1.5.1 Visualización de Datos

Toda empresa, organización o institución debe ser capaz de adaptarse a los cambios que puedan presentarse en el negocio en que se encuentra sumergida, de tal manera que

pueda mantenerse estable a pesar de la sucesión de dichos cambios. Para lograr este requerimiento las entidades deben implementar soluciones dirigidas al análisis de la información y a una correcta toma de decisiones, para lo cual es muy importante reducir el tiempo de demora entre la captura o almacenamiento de los datos y su procesamiento y disponibilidad. Los directivos necesitan constante acceso a los datos en tiempo real, interactivo, dinámico, para poder hacer decisiones críticas del negocio.

Existen tres métodos básicos para mostrar información a los encargados de tomar las decisiones en las empresas: los informes tradicionales, las alarmas y la visualización de los datos.

La visualización es particularmente eficaz para ambientes dónde hay grandes volúmenes de datos y mucha información que analizar, ya que una revisión rápida a través de un intuitivo visual resulta más factible para identificar los errores, formular hipótesis y tomar decisiones. Usar la visualización de datos como una tecnología de compromiso proporciona una funcionalidad reforzada para que los usuarios puedan mejorar la toma de decisiones, ya que esta incluye reportes en tiempo real y una representación gráfica fácil de entender para poder actuar.

El sistema objeto de este trabajo, emplea este método, ya que permite crear reportes de acuerdo a las necesidades de información del usuario y obtener dicha información actualizada y en el momento que se desee, ya sea como un informe o como un gráfico. [3]

1.5.2 Asistentes de Reportes y Gráficos

Los asistentes de reportes son herramientas que permiten a los usuarios crear, generar y almacenar reportes personalizados acerca de un determinado conjunto de datos. Están compuestos por pasos continuos que se muestran al usuario con el objetivo de que este indique qué información desea incluir en su informe, así como la manera en que debe mostrarse, por esta razón deben ser flexibles y con una interfaz rápida e intuitiva, fácil de comprender por el usuario. Generalmente dependen de otros subsistemas que almacenan la información que se puede reportar y brindan la opción de graficar los reportes que se crean. Con este fin son empleados los asistentes de gráficos que de igual manera ayudan al usuario paso a paso a construir su propio gráfico, permitiéndole seleccionar el tipo de

gráfico que desea (pastel, barra, línea...), los datos a representar y otras características acerca de la leyenda, las etiquetas de las series y categorías, etc.

1.5.3 La distribución de información mediante el correo electrónico

En la distribución de información mediante el correo electrónico, dicha información va de un origen a un destino. El origen es una persona o una institución que prepara una información, que va hacia un destino: otra persona que la recupera.

La interacción a través de correo electrónico es flexible en cuanto a que se puede establecer con una o varias personas, incluso con un grupo de ellas anteriormente definido. Este es el caso del empleo de las listas de distribución que permiten poner en contacto a todas las personas que comparten los mismos intereses.

Una lista de distribución es un conjunto de direcciones electrónicas que se utilizan para enviar ciertos mensajes o anuncios con un contenido de interés general para todos los miembros de la lista. Ésta es gestionada por uno o varios coordinadores (administradores), cuya misión principal es garantizar que se respeten las normas mínimas. Las listas de distribución son grupos de personas que se intercambian mensajes sobre una temática particular, compartiendo así los conocimientos y debatiendo temas de interés común, y que forman una comunidad virtual. Son herramientas poderosas para la coordinación y distribución de información.

Algunas de las ventajas del correo electrónico sobre otros medios de interacción humana residen en que:

- El correo electrónico es asincrónico. Cuando nos comunicamos por e-mail no necesitamos quedar con anterioridad con la persona para que esté en el lugar de la recepción, como sería en el caso del teléfono.
- El tiempo transcurrido entre la emisión y la recepción del mensaje es prácticamente instantáneo.
- Los participantes o interlocutores se encuentran en un ciberespacio educativo con pocos límites para la participación por el estatus o problemas personales.

- El correo electrónico no requiere un espacio y tiempo concreto para realizar comunicación por lo que las comunicaciones frecuentemente se hacen entre otras actividades.
- La comunicación puede ser entre individuos o grupos.

1.6 Diseño de aplicaciones WEB

1.6.1 El uso del XML

Según la W3C, XML es “el formato universal para documentos estructurados y datos en la Web” [4].

Sus siglas significan Extensible Markup Language, es decir XML es un lenguaje extensible porque puede agregar nuevas palabras para que se adecuen a propósitos específicos, y marcado porque incluye símbolos especiales en un documento para cumplir alguna función específica.

XML fue pensado en un principio para usarse en aplicaciones Web, sin embargo su utilidad se ha extendido y ahora se emplea también en aplicaciones estándares de la plataforma .NET y para el intercambio de datos entre sistemas desiguales, ya que permite que esto ocurra sin tener en cuenta la plataforma ni el lenguaje de desarrollo. Además da la posibilidad de que los datos estructurados sean descritos usando un formato texto-base, y entregados e intercambiados sobre HTTP que es el protocolo usado en la Web, una vez recibidos podrán ser usados para mostrarlos o para su intercambio entre servidores.

Al usar XML en el desarrollo de aplicaciones Web, se derivan un gran número de beneficios: un mejor contexto de búsqueda, sistemas más flexibles y la integración de datos provenientes de fuentes no compatibles entre sí, sin que estos tengan un formato predefinido, debido a que XML es auto descriptible. La separación de los datos de la presentación de los mismos, permite aplicaciones Web más poderosas, así como verlos múltiples veces sin tener que ir al servidor en cada ocasión para obtenerlos. Otra de las ventajas que proporciona es que la actualización de datos se puede lograr sin necesidad de retransmitirlos todos, sino sólo aquellos que han de cambiarse.

1.6.2 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 que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados. Ésta es la llamada plataforma .NET, y a los servicios antes comentados se les denomina servicios Web.

Para crear aplicaciones para la plataforma .NET, tanto servicios Web como aplicaciones tradicionales, Microsoft ha publicado el denominado kit de desarrollo de software conocido como *.NET Framework SDK*, que incluye las herramientas necesarias tanto para su desarrollo como para su distribución y ejecución. Este fue diseñado con tres objetivos en mente, lograr aplicaciones Windows mucho más estables y seguras, simplificar el desarrollo de aplicaciones y servicios Web que no sólo funcionaran en plataformas tradicionales, sino también en dispositivos móviles y proporcionar un solo grupo de bibliotecas que pudieran trabajar con varios lenguajes.[5]

El *.NET Framework* está compuesto por el Entorno Común de Ejecución o Common Language Runtime(CLR), la biblioteca de clases del *.NET Framework*, un grupo de lenguajes de programación y el entorno ASP.NET. A continuación se presenta el CLR y la biblioteca de clases y en próximos epígrafes se abordará el tema de los lenguajes de programación y el entorno ASP.Net

Common Language Runtime (CLR)

El Common Language Runtime (CLR) es un entorno de ejecución en el que aplicaciones escritas en diferentes lenguajes pueden trabajar juntas, esto se consigue a través de lo que se denomina código gestionado, que no es más que el código de las aplicaciones desarrolladas para la plataforma .NET, es decir, el CLR se encarga de gestionar la ejecución de estas aplicaciones, a las que además ofrece numerosos servicios que simplifican su desarrollo y favorecen su fiabilidad y seguridad. A continuación se presentan las principales características del CLR y algunos de estos servicios.

- **Modelo de programación consistente:** A todos los servicios y facilidades ofrecidos por el CLR se accede de la misma forma: a través de un modelo de programación orientado a objetos, lo que constituye una diferencia importante respecto al modo de acceso a los servicios ofrecidos por algunos de los sistemas operativos actuales, por ejemplo, los de la familia Windows, en los que existen servicios que se acceden a través de llamadas a funciones globales definidas en DLLs y otros que se acceden a través de objetos (objetos COM en el caso de la familia Windows).
- **Ejecución multiplataforma:** El CLR actúa como una máquina virtual que se encarga de ejecutar las aplicaciones diseñadas para la plataforma .NET, es decir, cualquier plataforma para la que exista una versión del CLR podrá ejecutar cualquier aplicación .NET. Microsoft ha desarrollado versiones del CLR para la mayoría de las versiones de Windows: Windows 95, Windows 98, Windows ME, Windows NT 4.0, Windows 2000, Windows XP y Windows CE (que puede ser usado en CPUs que no sean de la familia x86) y además ha firmado un acuerdo con Corel para portar el CLR a Linux.
- **Integración de lenguajes:** Desde cualquier lenguaje para el que exista un compilador que genere código para la plataforma .NET es posible utilizar código generado para la misma usando cualquier otro lenguaje. Esta integración de lenguajes hace posible escribir una clase en C# que herede de otra escrita en Visual Basic.NET que, a su vez, herede de otra escrita en C++ con extensiones gestionadas.
- **Gestión de memoria:** El CLR incluye un recolector de basura que evita que el programador tenga que tener en cuenta cuándo ha de destruir los objetos que dejen de serle útiles. Este recolector es una aplicación que se activa cuando se quiere crear algún objeto nuevo y se detecta que no queda memoria libre para hacerlo, caso en que recorre la memoria dinámica asociada a la aplicación, detecta qué objetos hay en ella que no puedan ser accedidos por el código de la aplicación, y los elimina para limpiar la memoria de "objetos basura" y permitir la creación de otros nuevos. Gracias a este recolector se evitan errores de

programación muy comunes como intentos de borrado de objetos que ya no existen, agotamiento de memoria por olvido de eliminación de objetos inútiles o solicitud de acceso a miembros de objetos ya destruidos.

- **Seguridad avanzada:** El CLR proporciona mecanismos para restringir la ejecución de ciertos códigos o los permisos asignados a los mismos, según su procedencia o el usuario que los ejecute. Por ejemplo, puede no darse el mismo nivel de confianza a código procedente de Internet que a código instalado localmente o procedente de una red local; puede no darse los mismos permisos a código procedente de un determinado fabricante que a código de otro; y puede no darse los mismos permisos a un mismo código según el usuario que lo esté ejecutando o según el rol que éste desempeñe.
- **Interoperabilidad con código antiguo:** El CLR incorpora los mecanismos necesarios para poder acceder desde código escrito para la plataforma .NET a código escrito previamente a la aparición de la misma y por tanto no preparado para ser ejecutando dentro de ella. Estos mecanismos garantizan el acceso a objetos COM y a funciones sueltas de DLLs preexistentes como la API Win32. [6]

La compilación de un programa gestionado por el CLR no se hace directamente a código nativo, sino a un lenguaje, más o menos como el ensamblador, llamado MSIL (Microsoft Intermediate Language).

Microsoft Intermediate Language (MSIL)

Todos los compiladores que generan código para la plataforma .NET no generan código máquina para CPUs x86 ni para ningún otro tipo de CPU concreta, sino que generan código escrito en el lenguaje intermedio conocido como Microsoft Intermediate Language (MSIL). El CLR da a las aplicaciones la sensación de que se están ejecutando sobre una máquina virtual, y precisamente MSIL es el código máquina de la misma. Es decir, MSIL es el único código que es capaz de interpretar el CLR, y por tanto cuando se dice que un compilador genera código para la plataforma .NET lo que se está diciendo es que genera MSIL.

MSIL ha sido creado por Microsoft tras consultar a numerosos especialistas en la escritura de compiladores y lenguajes tanto del mundo académico como empresarial. Es un lenguaje de un nivel de abstracción mucho más alto que el de la mayoría de los códigos máquina de las CPUs existentes, e incluye instrucciones que permiten trabajar directamente con objetos (crearlos, destruirlos, inicializarlos, llamar a métodos virtuales, etc.), tablas y excepciones (lanzarlas, capturarlas y tratarlas).

La principal ventaja del MSIL es que facilita la ejecución multiplataforma y la integración entre lenguajes al ser independiente de la CPU y proporcionar un formato común para el código máquina generado por todos los compiladores que generen código para .NET. Sin embargo, dado que las CPUs no pueden ejecutar directamente MSIL, antes de ejecutarlo habrá que convertirlo al código nativo de la CPU sobre la que se vaya a ejecutar. De esto se encarga un componente del CLR conocido como compilador JIT (Just-In-Time) o jitter que va convirtiendo dinámicamente el código MSIL a ejecutar, en código nativo según sea necesario. [7]

Bibliotecas de clases de .Net Framework

Las bibliotecas de clases de .NET contienen código para programar subprocesos, entrada y salida de archivos, compatibilidad para bases de datos, análisis XML y estructuras de datos, como pilas y colas y además está disponible para cualquier lenguaje de programación compatible con *.NET Framework*. Gracias al CLR los lenguajes .NET pueden usar cualquier clase de la biblioteca .NET, por lo que una funcionalidad disponible para uno de estos lenguajes lo estará también para el resto. [8]

1.6.2.1 Arquitectura de aplicaciones .NET

Actualmente existe un principio muy ventajoso para diseñar aplicaciones distribuidas. Consiste en dividir la aplicación en componentes que a su vez se agrupan en capas de acuerdo a los tipos de funciones que realizan. En la mayoría de las ocasiones estos componentes están organizados en forma de apilamiento para que los que se encuentran por "encima" de una capa determinada utilicen los servicios proporcionados por ésta, y un componente específico utilice la funcionalidad proporcionada por otros componentes de su propia capa, y otras capas "inferiores", para realizar su trabajo. Las capas son

agrupaciones lógicas de los componentes de software que conforman la aplicación o servicio, cada una contiene un número de tipos de componentes discretos agrupados en subcapas, las cuales realizan un tipo de tarea específica. Por lo general se dividen en capa de presentación, capa empresarial, capa de acceso a datos y capa de datos.

Al analizar la mayoría de las soluciones empresariales basadas en modelos de componentes por capas, se llega a la conclusión de que existen varios tipos de componentes que es habitual encontrarlos y que se muestran en la figura # 1.

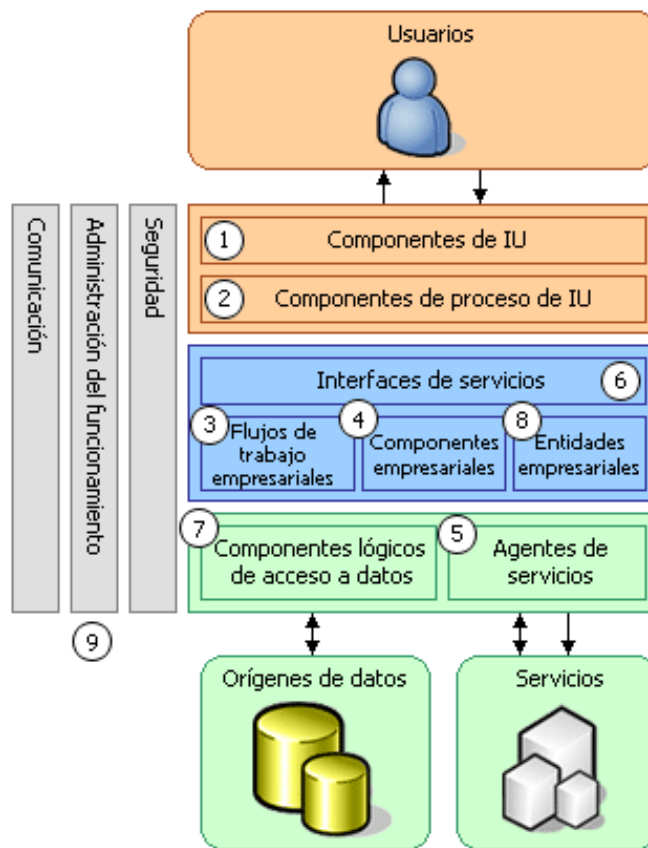


Figura # 1 Tipos de componentes habituales en aplicaciones distribuidas

En el Módulo de Diseño y Distribución de Reportes de Akademos sólo están presentes cinco de esos tipos de componentes que se explican a continuación:

Componentes de interfaz de usuario (IU): son el medio que tiene el usuario de interactuar con la aplicación. Las interfaces de usuario se implementan utilizando formularios de Windows Forms, páginas Microsoft ASP.NET, controles u otro tipo de tecnología que

permita procesar y dar formato a los datos de los usuarios, así como adquirirlos y validarlos.

Componentes de proceso de usuario: se emplean cuando el usuario interactúa con el sistema de acuerdo a un proceso predecible.

Componentes empresariales: estos componentes implementan toda la lógica empresarial o del negocio de la aplicación. En la mayoría de las ocasiones los datos no se almacenan tal y como se reciben por parte del usuario sino que necesitan ser transformados de acuerdo a alguna regla o procedimiento, de la misma manera que se requiere mostrar al usuario sólo la información de interés para él, por lo que de ambas tareas se encargan estos componentes.

Componentes lógicos de acceso a datos: usualmente las aplicaciones o servicios necesitan acceder a un almacén de datos para realizar determinadas funcionalidades, tanto para insertar uno nuevo como para extraer alguna información, por lo que resulta ventajoso abstraer la lógica necesaria para obtener acceso a los datos en un capa independiente, de modo que se centralice esta funcionalidad y se facilite la configuración y el mantenimiento de la misma.

Componentes de entidad empresarial: la mayoría de las aplicaciones requieren el paso de datos entre distintos componentes. Los datos se utilizan para representar entidades del mundo real. Las entidades que se utilizan de forma interna en la aplicación suelen ser estructuras de datos, como conjuntos de datos, `DataReader`, `DataSet`, o secuencia de lenguaje de marcado extensible. Es necesario aclarar que en el “Sistema de Diseño y Distribución de Reportes” los datos que se transmiten de una capa a otra son `DataSets`.

Diseño de capas de presentación

La capa de presentación contiene los componentes de interfaz de usuario y los de proceso de usuario, estos últimos, se diseñan cuando es necesario organizar los elementos de la interfaz y controlar la interacción de la aplicación con el que la usa, por lo que estos componentes son muy útiles cuando dicha interacción sigue una serie de pasos predecibles como es el caso de un asistente para diseñar reportes o crear gráficos. Por lo

general son clases .NET con métodos que encapsulan la lógica necesaria para realizar una acción específica en el proceso de usuario y para recorrer todos los pasos de este, la interfaz de usuario construye un objeto de esta clase.

Separar la interacción de la aplicación con el usuario en estos dos tipos de componentes permite mantener el estado de la ejecución de procesos largos más fácilmente, lo que hace posible el abandono y la reanudación de la interacción, incluso utilizando una interfaz de usuario diferente. Además brinda la posibilidad de que varias interfaces de usuario empleen el mismo componente de proceso de usuario.

Por otra parte, los componentes de proceso de usuario en sí ayudan a resolver ciertos problemas que pueden presentarse a la hora de diseñar las interfaces de usuario. Por ejemplo, cuando las aplicaciones permiten a los usuarios realizar varias tareas a la vez poniendo a su disposición más de un elemento de interfaz, estos componentes simplifican la administración del estado de varios procesos salientes encapsulando todo el estado necesario para el proceso en un solo componente. Si se utilizan varias ventanas o paneles en una actividad de usuario determinada, son capaces de sincronizarlas a través de la centralización del estado de las mismas en una única ubicación. Otra de los problemas que resuelve es que ciertos procesos de usuario se pueden poner en pausa y posteriormente reanudar, los componentes de proceso de usuario permiten almacenar su estado en una ubicación diferente a la de los datos empresariales, de manera que se puedan efectuar operaciones con estos sin necesidad de implementar reglas de filtrado para procesos incompletos.

Algunas funcionalidades de los componentes de interfaz de usuario

Cuando funcionan como componentes de entrada:

- Capturan los eventos del usuario y llaman a las funciones de control para indicar a los elementos de la interfaz de usuario que cambien el modo de visualización de los datos, bien inicializando una acción en el proceso de usuario actual, o bien, modificando los datos del mismo.

- Llevan a cabo la asignación y transformación simple de la información proporcionada por los controles del usuario en los valores necesarios para que los componentes subyacentes realicen su trabajo.
- Realizan la validación de entrada de datos. [9]

Cuando funcionan como componentes de salida:

- Adquieren y procesan los datos de los componentes empresariales o de los componentes lógicos de acceso a datos de la aplicación.
- Normalmente, procesan los datos de una entidad empresarial. Estas entidades se suelen obtener del componente de proceso de usuario, aunque también se pueden obtener de los componentes de datos. Los componentes de IU pueden procesar datos a través del enlace a datos de su visualización con los atributos y colecciones adecuados de los componentes de la entidad, sí ésta se encuentra disponible. Si se encuentra administrando los datos de una entidad como conjuntos de datos, esta tarea resulta bastante sencilla. Si ha implementado objetos de entidad personalizados, tal vez sea preciso implementar código adicional para facilitar el enlace a datos.
- Proporcionan la información de estado al usuario, por ejemplo, indicando cuando una aplicación se encuentra en modo "desconectado" o "conectado".
- Pueden utilizar un componente de utilidad para proporcionar funcionalidad de deshacer.

Algunas funcionalidades de los componentes de proceso de usuario

- Separan el flujo de la interacción del usuario conceptual de la implementación o dispositivo en el que ocurre.
- Encapsulan el modo en que las excepciones pueden afectar al flujo de proceso de usuario.
- Realizan el seguimiento del estado actual de la interacción del usuario.
- Mantienen los datos internos relacionados con la lógica empresarial de la aplicación y su estado interno, manteniendo los datos del modo adecuado.

- Mantienen el estado empresarial interno, normalmente aferrándose a una o varias entidades empresariales afectadas por la interacción del usuario. [11]

Componentes empresariales en el diseño de capas empresariales

La capa empresarial contiene las interfaces de servicio y los componentes, entidades y flujos de trabajo empresariales. Para diseñar la lógica empresarial, es necesario decidir si es preciso organizar o no el proceso empresarial, o si será suficiente con disponer de un conjunto de componentes empresariales. Estos últimos son suficientes cuando se necesita encapsular funcionalidad y lógica que se pueda volver a utilizar por parte de numerosos procesos empresariales y además cuando se necesita disponer del control total de los datos y del flujo de lógica.

Algunas funcionalidades de los componentes empresariales:

- Deben validar la entrada y la salida.
- Pueden llamar a componentes lógicos de acceso a datos para recuperar y actualizar los datos de la aplicación.
- Pueden llamar a otros componentes empresariales e inicializar flujos de trabajo empresariales.
- Pueden enviar una excepción al llamador si se produce algún error al utilizar transacciones atómicas. [12]

Componentes lógicos de acceso a datos en el diseño de capas de datos

La capa de datos incluye los componentes lógicos de acceso a datos, los orígenes de datos, los agentes de servicios y los servicios.

El uso de los componentes lógicos de acceso a datos se recomienda para todas las aplicaciones que necesiten realizar alguna acción sobre uno o varios orígenes de datos. Estos componentes suelen implementar un patrón de diseño sin estado que separa el procesamiento empresarial de la lógica de acceso a datos y presentan métodos que crean, leen, modifican y eliminan alguna entidad empresarial determinada de la aplicación. Es válido aclarar que la existencia de varios almacenes de datos debe

corresponderse con el diseño de un componente de este tipo para cada uno de ellos, cediendo esta funcionalidad a los componentes empresariales.

Algunas funcionalidades de los componentes lógicos de acceso a datos

Cuando se llaman, los componentes lógicos de acceso a datos realizan lo siguiente:

- Llevan a cabo asignaciones y transformaciones simples de argumentos de entrada y salida. De este modo, se abstrae la lógica empresarial de los esquemas de la base de datos y las formas de procedimientos almacenados.
- Obtienen acceso de un único origen. De este modo, aumenta la facilidad del mantenimiento desplazando toda la funcionalidad de agregación de datos a los componentes empresariales, donde los datos se pueden agregar en función de la operación empresarial específica que se está realizando.
- Actúan en una tabla principal y realizan operaciones en tablas relacionadas. (Los componentes lógicos de acceso a datos no tienen por qué encapsular necesariamente operaciones sólo en una tabla de un origen de datos subyacente.) De este modo, se aumenta la facilidad de mantenimiento de la aplicación. [13]

1.7 Fundamentación de la metodología utilizada

1.7.1 Rational Unified Process™ (RUP)

El Proceso Unificado, primeramente, es un proceso de desarrollo de software, y un proceso de desarrollo de software, es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado no es un simple proceso, sino que constituye un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

Utiliza el Lenguaje Unificado de Modelado(UML) para preparar todos los esquemas de un sistema y tiene tres aspectos definitorios de los cuales se argumenta a continuación:[16]

- **Dirigido por Casos de Uso**

El proceso de desarrollo sigue un hilo conductor y avanza a través de una serie de flujos de trabajo que parten de los casos de uso. Los casos de uso se especifican, se diseñan y los casos de uso finales son la fuente a partir de la cual se construyen los casos de prueba. Estos casos de uso van encaminados a satisfacer los requisitos funcionales, o sea, las necesidades del cliente.

- **Centrado en la arquitectura**

El papel de la arquitectura software es parecido al papel que juega la arquitectura en la construcción de edificios. Así como un edificio se contempla desde varios puntos de vista la arquitectura en un sistema software se describe mediante diferentes vistas del sistema.

- **Iterativo e incremental**

Consiste en dividir el proyecto en mini-proyectos. Cada mini-proyecto es una iteración que resulta en un incremento. Una iteración es una secuencia de actividades con un plan establecido y criterios de evaluación, cuyo resultado es una versión del software. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto. Para una efectividad máxima, las iteraciones deben estar controladas, o sea, deben seleccionarse y ejecutarse de una forma planificada.

Las iteraciones al ser mini-proyectos cumplen con las fases de un sistema en desarrollo: requerimientos, análisis, diseño, implementación y prueba.

Los beneficios de un proceso iterativo controlado son los siguientes:

- ❖ Reduce el costo del riesgo a los costos de un solo incremento.
- ❖ Reduce el riesgo de no sacar al mercado el producto en la fecha prevista.
- ❖ Acelera el ritmo del esfuerzo de desarrollo en su totalidad debido a que se trabaja de manera más eficiente para obtener resultados claros a corto plazo.

- ❖ Permite definir completamente las necesidades del usuario y sus correspondientes requisitos, lo cual es difícil hacerlo desde un principio. De esta forma se hace más fácil la adaptación a los requisitos cambiantes.

1.7.2 Unified Modeling Language(UML)

El Lenguaje Unificado de Modelado es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos, además de la visualización, especificación, construcción y documentación de los artefactos de sistemas en los que el software juega un papel importante.

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema y pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

1.8 Lenguaje, Gestor de Base de Datos y Software utilizado

1.8.1 C#

C#, leído en inglés “C Sharp” y en español “C Almohadilla”, es el nuevo lenguaje de propósito general diseñado por Microsoft para su plataforma .NET. Toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo.

Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que C# carece de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que C# es el lenguaje nativo de esta plataforma.

A continuación se recoge de manera resumida las principales características de C#:

- **Modernidad:** C# incorpora en el propio lenguaje elementos muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular. Es el caso de una instrucción *foreach* que permite recorrer colecciones

con facilidad y es ampliable a tipos definidos por el usuario, la inclusión de un tipo básico *string* para representar cadenas o la distinción de un tipo *bool* específico para representar valores lógicos.

- **Orientación a objetos:** como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro, ya que no admite ni funciones ni variables globales. Soporta todas las características propias del paradigma de programación orientada a objetos: encapsulamiento, herencia y polimorfismo.
- **Orientación a componentes:** es el primer lenguaje de la familia C/C++ que está orientado al trabajo con componentes. La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente propiedades, eventos o atributos.

Para crear un componente en C# sólo es necesario crear una nueva clase, es decir, cualquier clase de objeto C# es un componente sin necesidad de crear GUID para interfaces y clases de componentes, efectuar operaciones de registro o implementar interfaces. Luego podemos usar C# para fabricar los componentes que encapsulan la lógica del negocio y acceden a bases de datos, las interfaces de usuario y el código del servidor que actúa detrás de las interfaces Web, por lo que el desarrollo de aplicaciones distribuidas es más simple.

- **Compatible:** para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir directamente en código escrito en C#, fragmentos de código escrito en estos lenguajes, sino que el CLR también ofrece, a través de los llamados Platform Invocation Services (PInvoke), la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos, tales como las DLLs de la API Win32. Nótese que la capacidad de usar punteros en código inseguro permite que se pueda acceder con facilidad a este tipo de funciones, ya que éstas muchas veces esperan recibir o devuelven punteros.

También es posible acceder desde código escrito en C# a objetos COM. Para facilitar esto, el *.NET Framework SDK* incluye las herramientas llamadas *tlbimp* y *regasm* mediante las que es posible generar automáticamente clases *proxy* que permitan, respectivamente, usar objetos COM desde .NET como si de objetos .NET se tratase y registrar objetos .NET para su uso desde COM.

Finalmente, también se da la posibilidad de usar controles ActiveX desde código .NET y viceversa. Para lo primero se utiliza la utilidad *aximp*, mientras que para lo segundo se usa la ya mencionada *regasm*.

Integración en Visual Studio.NET

Como se ha podido apreciar C# es un lenguaje potente, flexible, moderno y seguro. Sin embargo el mayor sentido práctico de este lenguaje se encuentra en Visual Studio .NET gracias a la estrecha integración entre dicho entorno y C#, mucho mayor que la que tiene Visual C++ .NET y equivalente e incluso superior a la de Visual Basic .NET. Gracias a esta integración, los diseñadores de Visual Studio .NET son capaces de generar código C# al crear interfaces de usuario basadas en formularios Windows y formularios Web, producir adaptadores de datos, guiones embebidos en documentos ASP.NET, etc .

Este lenguaje está pensado con el objetivo de ofrecer capacidades RAD a los desarrolladores de C++ sin sacrificar la potencia y flexibilidad. Esas cualidades RAD son las que encontramos en el entorno de Visual Studio .NET, mientras que el propio lenguaje pone el resto. [14]

Estandarización de C#

“El lenguaje C# no ha sido creado tan sólo por y para Microsoft y su plataforma .Net. Conjuntamente con Intel y HP, a finales del año 2000 Microsoft remitió al ECMA (European Computer Manufacturers Association), una institución independiente dedicada al proceso de creación de estándares, tanto el borrador del lenguaje C# como el de CLI(Common Language Infraestructura), parte fundamental de .Net. “ . [15]

El hecho de que C# sea un estándar significa que potencialmente se puede crear un compilador de este lenguaje para cualquier sistema operativo, recurriendo para ello al estándar definido en el ECMA. No se dependerá de Microsoft para esto, ni esta empresa controlará de manera unilateral la evolución del lenguaje.

1.8.2 ASP.NET

ASP.NET mejora al original ASP proporcionando “código detrás”. En ASP, el HTML y las secuencias de comandos se mezclaban en un documento, con ASP.NET y su código detrás se pueden separar. Ahora, cuando la lógica de una página Web necesite cambiar no hace falta buscar por cientos o miles de líneas de HTML para cambiar la secuencia de comandos que necesita modificarse. Por otra parte ASP.NET introduce dos importantes ventajas que son los Servicios y los Formularios Web, estos últimos permiten arrastrar y colocar controles en sus formularios y codificarlos como haría en cualquier típica aplicación Windows.

ASP.NET está integrado al Visual Studio .NET, el cual provee un diseñador GUI y una barra de herramientas enriquecida, como es lógico también usa el compilador JIT que compila el código cuando es instalado en el servidor o la primera vez que es necesario, lo que aumenta enormemente la velocidad, pues las páginas ASP tradicionales se ejecutaban muy lentamente porque siempre era interpretado el código.

Otras de las ventajas que presenta ASP.NET son:

- **Flexibilidad:** Toda la librería de clases de .NET puede ser accedida por aplicaciones ASP.NET, se puede usar el lenguaje que mejor se adecue a la funcionalidad que se quiere implementar porque ASP.NET es un lenguaje independiente.
- **Seguridad:** Las aplicaciones .NET usan un conjunto de esquemas de autenticación y autorización por defecto, estos se pueden modificar de acuerdo a la necesidad de seguridad de una aplicación.

1.8.3 Visual Studio.NET 2003

Visual Studio .NET es la herramienta que Microsoft ha publicado con el fin de crear aplicaciones para la plataforma .NET desde una interfaz visual basada en ventanas. Este software tiene un rasgo particular al constituir un solo ambiente de desarrollo de aplicaciones para todos los lenguajes.

Visual Studio .NET simplifica el desarrollo de aplicaciones en un ambiente que es una mezcla de lenguajes y lo hace a través de ciertas características como por ejemplo diseñadores visuales para XML, HTML, datos y código del lado del servidor. Visual Studio .NET es capaz de proveer este nivel de integración porque cuenta con las facilidades del *.NET Framework*.

Estas son algunas de las principales características del Visual Studio .NET y sus ventajas:

- Lista de Tareas: organiza las tareas y administra los errores y las advertencias en un solo lugar. Las tareas se leen de los comentarios especializados del código fuente y se presentan en un formato tabular, donde se le hace un *doble-click* para saltar a la sección del código fuente donde está el error o advertencia.
- Explorador de la Solución: brinda una vista jerárquica de una solución dividida en proyectos. Permite la administración de los proyectos involucrados en una solución.
- Explorador de Servidores: administra su computador y otros que están en red, incluyendo recursos como el Servidor de SQL, colas de mensajes, servicios.
- Multi-Monitor: permite un mejor uso del espacio disponible en la pantalla.
- Ayuda Dinámica: hace referencias a la documentación de la ayuda disponible de acuerdo a la materia sobre la cual se esté trabajando.

1.8.4 SQL Server 2000

En el mes de Septiembre del año 2000 Microsoft anunció oficialmente su línea .NET Enterprise Servers como el modelo de la arquitectura de las aplicaciones .NET, su meta

fundamental es ofrecer una dirección simplificada, escalabilidad, disponibilidad y un gran apoyo para las aplicaciones de este tipo.

SQL Server es el primer gestor de base de datos perteneciente a la familia de .NET Enterprise Servers y como tal, ofrece almacenamiento de datos, administración de componentes de los servicios .NET, apoyo nativo a XML y a los protocolos estándares de Internet como HTTP y SSL. Además numerosas mejoras de productividad para los programadores de SQL Server, las cuales se abordan a continuación:

- Soporta XML: SQL Server 2000 ofrece ayuda para leer, escribir, entregar y usar documentos XML, apoyado por nuevas funciones que le fueron adicionadas como por ejemplo: *Select...For XML*, *OPENXML*, *sp_xml_preparedocument*, *sp_xml_removedocument*.
- Herramientas y tecnologías de desarrollo: SQL Server 2000 trae muchas herramientas y tecnologías nuevas, algunas de estas se presentan a continuación:
 - ❖ *Query Analyzer*: incluye un *Object Browser* que proporciona un *tree view* común de todos los objetos que existen en el Servidor de SQL, por lo que no será necesario cambiar del *Query Analyzer* al *Enterprise Manager* o al modelo de datos, sólo basta con buscar los objetos en este navegador. Además tiene un depurador que permite ir paso a paso en el procedimiento almacenado de una manera más fácil que en herramientas anteriores, ya que no se necesita de mucho conocimiento para trabajar con él correctamente.
 - ❖ Nuevos tipos de datos: SQL Server 2000 agrega tres tipos de datos nuevos a la base de datos relacional. *Bigint*, un tipo de dato entero de 8 byte con un rango de valores desde -2 elevado a 63 ($-9,223,372,036,854,775,808$) hasta 2 elevado a $63-1$ ($9,223,372,036,854,775,807$). *SQL_variant*, capaz de almacenar varios tipos datos SQL Server, exceptuando *text*, *ntext*, *image*, *timestamp*, y *sql_variant* y el más interesante de todos es el tipo de datos *table*. Una variable de este tipo puede ser usada en lugar de las tablas temporales tradicionales para construir conjuntos de resultados o

para devolver conjuntos de resultados provenientes de una nueva UDF (User-Defined Functions) de SQL Server 2000.

- ❖ UDF (User-Defined Function): SQL Server 2000 no se restringe a un conjunto de funciones predefinidas, ahora el programador puede crear sus propias funciones, usando la declaración *Create Function*, la cual permite definir parámetros de entrada y salida a esta función personalizada. UDFs aumentará la productividad del diseñador en las aplicaciones de SQL.

1.9 Conclusiones

La investigación llevada a cabo para la realización de este capítulo ha servido en gran manera para conocer el entorno en que se desarrollará el sistema y por tanto las deficiencias que este presenta a la hora de obtener y distribuir información, por esta razón, ha facilitado la precisión de los objetivos concretos del sistema que se presenta.

Además el estudio de las tendencias y tecnologías actuales permitió definir cuáles se usarían en el sistema, adoptando este las más ventajosas y novedosas.

Por otra parte se profundizó en las características del lenguaje C#, el Gestor de Base de Datos SQL Server 2000 y el software Visual Studio. NET 2003, lo cual ha garantizado, de seguro, una mayor explotación de sus ventajas a la hora implementar la aplicación.

CAPÍTULO II DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

2.1 Introducción

En el presente capítulo se describe, a través de varios epígrafes, la solución propuesta al problema planteado inicialmente.

Debido a que la obtención y distribución de información no constituye un negocio constante y predecible se hace uso del Modelo del Dominio para lograr una comprensión del contexto del sistema. Este modelo captura los tipos de objetos más importantes y los representa en el Diagrama de Clases del Modelo de Objetos.

Por otra parte aparecen claros y precisos los requisitos funcionales y no funcionales que debe cumplir el sistema y también se desarrolla el Modelo de Casos de Uso del mismo.

2.2 Modelo del Dominio

2.2.1 Definición de los conceptos principales

Reporte: es cualquier tipo de informe que se solicite o se cree que contiene alguna información. Existen reportes de Matrícula, Evaluaciones.

Estudiante: persona que recibe la formación académica proporcionada por cualquier centro educativo.

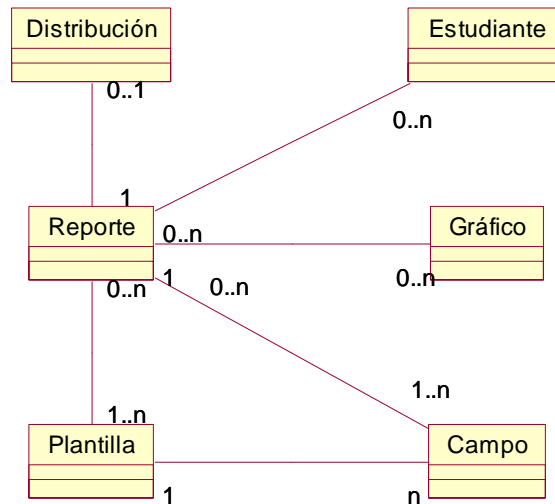
Plantilla: documento que presenta un formato y contenido específicos.

Campo de una plantilla: es un dato contenido en una plantilla determinada.

Gráfico: es una imagen que representa algún dato estadístico en forma visual.

Distribución: es el evento de hacer llegar información a todo aquel personal que la solicite y realmente la necesite para cumplir su función laboral.

2.2.2 Diagrama de Clases del Modelo de Objetos



2.3 Requisitos funcionales

- 1 Iniciar el sistema
- 2 Diseñar reportes de listados de estudiantes a partir de un asistente.
 - 2.1 El usuario especifica:
 - Las plantillas de las que desea extraer información para el reporte.
 - Los campos de las plantillas seleccionadas que desea incluir en el informe.
 - 2.2 Conformar una expresión lógica que filtre la información que se ha seleccionado.
 - 2.2.1 El usuario especifica:
 - El campo por el que se desea filtrar la información.
 - El operador de comparación que puede aplicar a ese campo (mayor que, menor que, igual a, diferente de, un valor determinado).
 - El valor que tomará el campo.
- 2.3 Ordenar los datos del informe.

2.3.1 El usuario especifica:

- Los campos por los que se quiere ordenar el informe.
- El sentido de la ordenación.

2.4 Almacenar automáticamente el diseño del reporte.

2.4.1 El usuario especifica:

- Nombre
- Título
- Subtítulo

3 Diseñar reportes de totales o cruzados a partir de un asistente.

3.1 El usuario especifica:

- Las plantillas de las que desea extraer información para el reporte
- El campo que desea se muestre en las filas de su tabla.
- El campo que desea se muestre en las columnas de su tabla.

3.2 Conformar una expresión lógica que filtre la información que se ha seleccionado.

3.2.1 El usuario especifica:

- El campo por el que se desea filtrar la información.
- El operador de comparación que puede aplicar a ese campo (mayor que, menor que, igual a, diferente de, un valor determinado).
- El valor que tomará el campo.

3.3 Almacenar automáticamente el diseño del reporte.

3.3.1 El usuario especifica:

- Nombre
- Título
- Subtítulo

4 Diseñar reportes de evaluaciones a partir de un asistente.

4.1 El usuario especifica:

- Evaluación
- La plantilla de la que desea extraer información para el reporte
- El campo de esa plantilla que interceptará con los datos de la evaluación indicada.

4.2 Conformar una expresión lógica que filtre la información que se ha seleccionado.

4.2.1 El usuario especifica:

- El campo por el que se desea filtrar la información.
- El operador de comparación que puede aplicar a ese campo (mayor que, menor que, igual a, diferente de, un valor determinado).
- El valor que tomará el campo.

4.3 Almacenar automáticamente el diseño del reporte.

4.3.1 El usuario especifica:

- Nombre
- Título
- Subtítulo

5 Permitir consultar un reporte.

5.1 El usuario especifica el reporte que desea consultar.

5.2 Actualizar y mostrar la información del reporte en el momento de la consulta.

5.3 Imprimir el informe.

5.4 Exportar el informe.

5.4.1 El usuario especifica la herramienta a la que desea exportar.

5.5 Graficar el reporte consultado.

5.5.1 Crear un gráfico que muestre la información del reporte consultado.

6 Actualizar la lista de reportes del sistema.

6.1 Modificar el diseño de un reporte.

6.1.1 El usuario especifica el reporte que desea modificar.

6.1.2 Insertar nuevos campos al reporte.

6.1.3 Eliminar campos del reporte.

6.1.4 Aplicar un filtro a la información del reporte.

6.1.5 Cambiar el filtro.

6.1.5.1 Cambiar el campo y/o el valor por el que se filtrará la información.

6.1.5.2 Cambiar el operador de comparación entre el campo y el valor correspondiente.

6.1.5.3 Insertar un operador lógico.

6.1.5.4 Cambiar el operador lógico.

6.1.6 Aplicar un ordenamiento

6.1.6.1 Agregar campos al ordenamiento.

6.1.6.2 Cambiar el sentido del ordenamiento.

6.2 Eliminar un reporte.

6.2.1 El usuario especifica el reporte a eliminar.

7 Actualizar las carpetas que contienen los reportes del sistema.

7.1 Modificar el contenido de una carpeta.

7.1.1 El usuario especifica la carpeta que desea modificar.

7.1.2 Mover una carpeta o reporte contenida en la carpeta seleccionada.

7.1.2.1 El usuario especifica:

- El elemento que desea mover.
- La carpeta a la que desea mover el elemento.

7.1.2.2 Eliminar el elemento seleccionado de la carpeta origen e insertarlo en la carpeta destino.

7.2 Eliminar una carpeta.

7.2.1 El usuario especifica la carpeta que desea eliminar.

7.3 Crear una nueva carpeta.

7.3.1 El usuario especifica el nombre.

8 Crear un gráfico a partir de un asistente.

8.1 El usuario especifica:

- El reporte para diseñar el gráfico que visualizará su información.
- El tipo de datos que desea crear (barra, línea, pastel).
- Los campos del reporte que se convertirán en series de su gráfico.
- Título de los ejes.

8.2 Almacenar automáticamente el diseño del reporte.

8.2.1 El usuario especifica:

- Nombre
- Título
- Subtítulo

9 Planificar la distribución de un reporte.

9.1 El usuario especifica:

- El reporte para planificar su distribución a los diferentes trabajadores que lo soliciten.

- La persona o lista de distribución destino del informe.
 - La fecha de la distribución.
 - La periodicidad si así lo desea (diaria, semanal, mensual o anual).
- 9.2 Almacenar la planificación de la distribución del reporte señalado.
- 10 Permitir gestionar la seguridad de un reporte determinado, usando el subsistema de seguridad de Akademos.
- 11 Distribuir un reporte según la planificación.
- 12 Permitir consultar un gráfico.
- 12.1 El usuario especifica el gráfico que desea consultar.
- 12.2 Actualizar y mostrar la información del gráfico en el momento de la consulta.
- 12.3 Imprimir el gráfico.

2.4 Requisitos no funcionales

- Apariencia o interfaz externa
 - ❖ Debe desarrollarse en un ambiente Web, de manera que pueda accederse a la información desde cualquier lugar de la universidad y en cualquier momento que se requiera, sin tener que estar limitado a un puesto de trabajo.
 - ❖ No deben predominar colores fuertes e incandescentes que agoten al usuario, sólo de ver la interfaz. Se recomienda el color verde.
 - ❖ Debe ser sencilla y mostrar todas las opciones claramente con el objetivo de agilizar todo el proceso.

- Usabilidad
 - ❖ El sistema debe ser fácil de entender y manejar, logrando que el usuario se sienta cómodo al notar que puede crear sus propios reportes, aplicando sólo un poco de lógica.

- ❖ Los asistentes cumplirán su objetivo en pocos pasos, para que no se convierta en algo tedioso y lento obtener información necesaria en un momento determinado.
 - ❖ No todas las personas usarán el sistema de la misma manera y con el mismo objetivo, ya que existirán algunas, que sólo podrán consultar la información de los reportes si cuentan con este permiso, esto garantiza la seguridad de los mismos.
- Rendimiento
 - ❖ El sistema debe diseñarse de tal forma que demore en caducar, los asistentes serán generales y personalizables para cubrir el mayor porcentaje de necesidades de información que puedan surgir.
 - ❖ Debe tener una alta eficiencia en cuanto al tratamiento de la información y garantizar su consistencia y disponibilidad en todo momento, por lo que se requiere de una alta velocidad de procesamiento y un bajo tiempo de recuperación.
 - Portabilidad

Se prevé futura migración del sistema a un entorno libre por lo que es necesario no hacer uso de funcionalidades de la plataforma de desarrollo (Microsoft.NET) específicas del sistema operativo.

- Seguridad

Como todo sistema informático el Sistema de Diseño y Distribución de Reportes también debe garantizar la seguridad en todas sus aristas. Para ello, acoge la política que sigue Akademos en este aspecto, el cual cuenta con un módulo llamado Seguridad que se encarga de gestionar todo lo relacionado con este requisito para todo el sistema en general. Este subsistema responde por la seguridad de los datos a través de Microsoft SQL Server 2000, la seguridad de los recursos lógicos, con la

utilización del sistema operativo Windows y por la seguridad de los elementos específicos del negocio que automatiza la aplicación. El sistema propuesto debe hacer uso de un control, implementado por el Módulo de Seguridad, que se encarga, del control y definición del acceso a los elementos del sistema.

- Ayuda y documentación en línea

Para informar a los usuarios todo lo referente a cómo trabaja el sistema, se incorporará una ayuda que les permitirá orientarse dentro de la aplicación y les explicará cómo trabajar con ella. Además cada vez que se cometa un error se mostrará un mensaje sugiriendo la posible causa que lo ocasionó y una propuesta para solucionarlo.

- Software

- ❖ Cliente: Sistema operativo con interfaz grafica y soporte para red, Navegador Web.

- ❖ Servidor: Windows 2000 Professional Service Pack 2.0 o superior, .NET Framework. 1.1, SQL Server 2000.

- Hardware

- ❖ Para el desarrollo: Pentium 600 MHz o superior, 128 MB de memoria RAM, 20 GB de disco duro.

- ❖ Para la explotación:

Clientes: Pentium de 133 MHz o superior, 128 MB de memoria RAM mínima, 256 MB de memoria RAM recomendada.

Servidor Web y de Correo: Dual PIII de 933MHz, 512 MB de memoria RAM, 60 GB de disco duro.

Servidor de Base de Datos: Dual PIII de 933MHz, 512 MB de memoria RAM, 100 GB de disco duro.

- Restricciones en el diseño y la implementación

Utilizar la plataforma Microsoft.NET

2.5 Beneficios tangibles e intangibles

Beneficios intangibles

El uso del sistema familiarizará más a los usuarios a la navegación, y por tanto al empleo de la tecnología moderna, lo cual hará que aumenten sus conocimientos prácticos. Además estos usuarios, en su mayoría directivos de la universidad, con una extremada carga de trabajo, afectarán menos su tiempo cuando necesiten obtener alguna información del estado académico del área que les interese, debido a que el sistema la facilita en menos tiempo y de manera más cómoda, al ser accesible desde cualquier computadora conectada a la red del Instituto y lo que es más importante, cada uno podrá elaborar sus propios informes sin necesidad de solicitarlo a niveles inferiores y esperar que estos recopilen y procesen la información para llegar al resultado. No obstante, si algún directivo desea asignarle la responsabilidad de algún reporte a otra persona, esta podrá hacérselo llegar, periódicamente y de forma actualizada, a través de la aplicación, con sólo planificar la distribución de dicho reporte. Esto hace que se mejore la calidad de los servicios prestados en cuanto a tiempo de respuesta, procesamiento de datos y centralización y organización de la información y se le permita a los usuarios dedicarle más tiempo al análisis y a la toma de decisiones.

Por otra parte, como la aplicación ofrece una interfaz para definir los permisos sobre los reportes, cada persona que diseñe uno de estos, podrá sentirse segura y confiada en cuanto a la integridad de la información que contiene.

El Módulo de Diseño y Distribución de Reportes permite hacer investigaciones sociales. La masa estudiantil de la Universidad de las Ciencias Informáticas es en sí misma una población representativa de la masa estudiantil cubana, dada su variedad en términos de proveniencia geográfica, social, académica. Es por eso que mediante la utilización del

sistema pueden realizarse análisis importantes dado que este posibilita generar reportes basados en la combinación de múltiples variables.

Beneficios tangibles

Aún cuando existan restricciones legales en cuanto a la documentación dura que se maneja en la gestión académica, la posibilidad de que tanto profesores como estudiantes, directivos y personal de secretaría, puedan elaborar en línea y desde cualquier punto de la intranet universitaria, reportes con gran parte de la información que se gestiona en Akademos, hace que se incurra en un ahorro considerable de materiales de oficina (papel, toner de impresoras, tinta de lapiceros, etc).

Para el desarrollo de este sistema se estudiaron un conjunto de otros que tienen objetivos similares. Uno de ellos es el Crystal Report 9, el cual tiene un precio de 1995 dólares en su edición avanzada[18], por lo que se llega a la conclusión de que el desarrollo de la aplicación ahorra a la universidad esa cifra considerable de divisas, garantizándole mejores resultados pues se adapta a las características de la gestión académica.

2.6 Análisis de costos y beneficios

Dada la naturaleza dinámica de los procesos académicos, la utilización de herramientas que permitan la creación personalizada de reportes es indispensable, no obstante, el uso de determinado sistema y la viabilidad de su puesta en explotación depende de los beneficios que este ofrezca y de los costos asociados al mismo.

Anteriormente se han analizado todos los beneficios de la utilización del sistema propuesto y teniendo en cuenta que la UCI ya cuenta con el capital humano y la infraestructura tecnológica necesaria para la construcción y explotación del sistema se considera factible el desarrollo del mismo.

2.7 Descripción del sistema propuesta

2.7.1 Concepción general del sistema

El Sistema de Diseño y Distribución de Reportes automatiza la actividad de obtener y distribuir información académica. Como se ha reflejado anteriormente, esta actividad no siempre se desarrolla de la misma manera, por lo que se ha decidido hacer uso de asistentes generales que puedan emplearse en la mayoría de las ocasiones. Para mostrar los datos contenidos en los reportes diseñados, se emplean técnicas de visualización como son los propios reportes y las imágenes o gráficos, de esta manera será mucho más fácil para el usuario entender la información mostrada. Como dentro de los objetivos del sistema no está sólo el permitir la obtención de información, sino también hacerla fluir por los distintos niveles de la universidad, es necesario adoptar una vía para cumplirlo, por esta razón se ha seleccionado el correo electrónico como la más apropiada para este caso. Una de las tendencias actuales en el mundo de la informática es el desarrollo de las aplicaciones en el ambiente Web, como es lógico, también esta tendencia se adecua al sistema, ya que será usado por los ordenadores de la mayoría de las personas del centro, quienes podrán interactuar con el mismo con sólo acceder a la Intranet, ahorrándose el trabajo de instalarlo en cada uno de estos ordenadores.

La aplicación ofrece varios tipos de opciones, por lo que es necesario crear roles para definir quienes podrán asumir uno u otro. A continuación se describen cada uno de los actores y las tareas que realizan.

Consultor: este actor representa a todos los usuarios que tienen acceso al sistema, podrá iniciarlo y consultar la información que brindan los reportes a los que tiene acceso, esto incluye que pueda exportarlos a Microsoft Word, Microsoft Excel, imprimirlos y obtener un gráfico de esta información si así lo desea.

Diseñador: representa a las personas que tienen derecho, porque su trabajo así lo requiere, de diseñar reportes de listado, de totales y de evaluaciones, a los cuales puede también crearle un gráfico personalizado que represente la información que contiene y definirle los permisos que tendrá sobre ellos cada persona. También podrá actualizar la lista de reportes y carpetas del sistema.

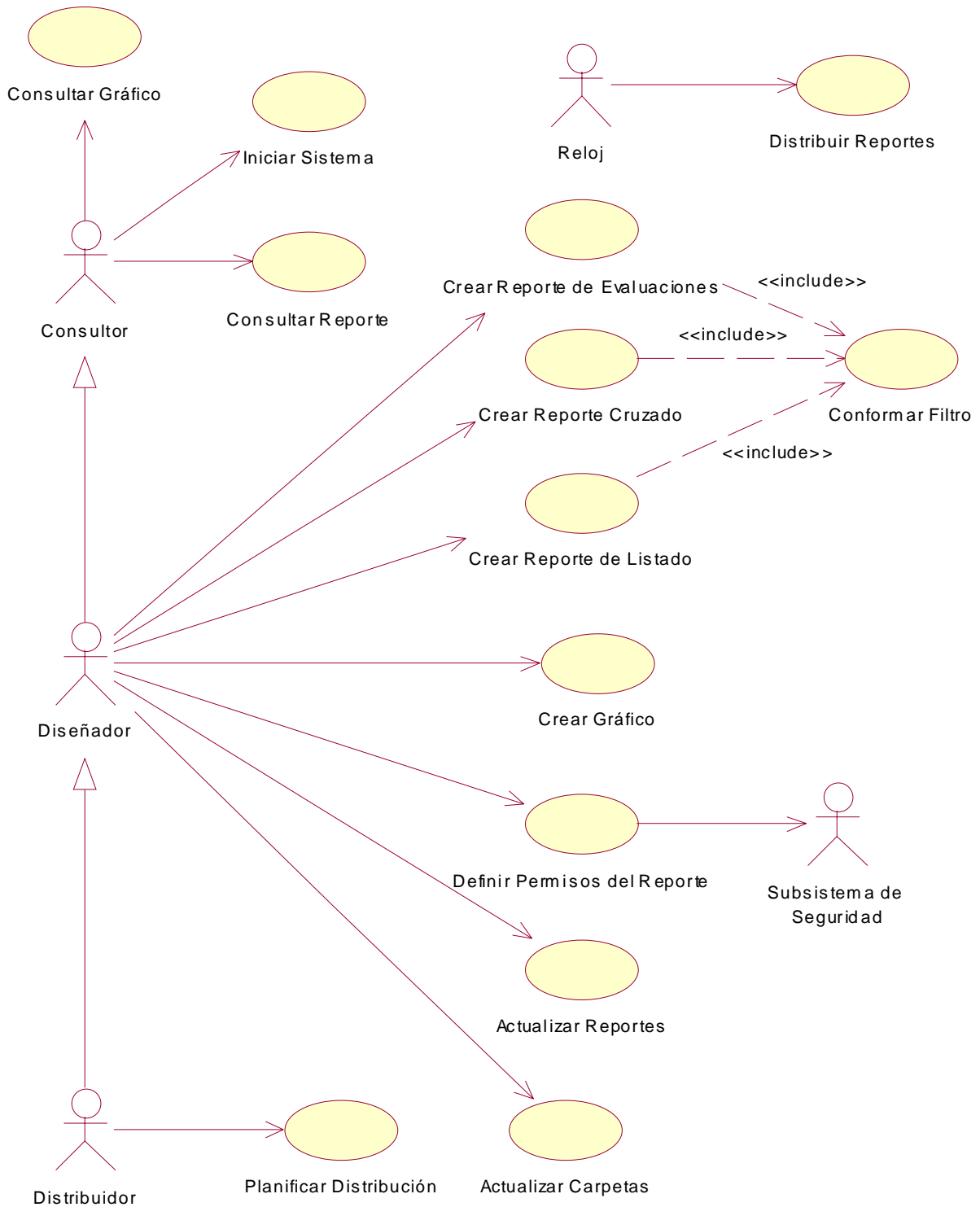
Distribuidor: representa a las personas que tienen derecho, porque su trabajo así lo requiere, de distribuir la información de los reportes a través del correo electrónico, para esto podrá planificar esta distribución con antelación.

Subsistema de Seguridad: este actor es un sistema que se encarga de la seguridad en todo el sistema Akademos, por tanto tiene la responsabilidad de gestionar los permisos que se le asignará a cada persona sobre los reportes creados.

Reloj: este actor es el reloj del Servidor Web y activa la distribución de los reportes según la planificación hecha por el distribuidor.

2.7.2 Modelo de Casos de Uso del Sistema

2.7.2.1 Diagrama de Casos de Uso del Sistema



2.7.2.2 Descripción general de los Casos de Uso no detallados

Caso de Uso:	Distribuir Reportes
Actores:	Reloj
Propósito:	Efectuar la distribución de los reportes según la planificación.
Resumen:	El Caso de Uso comienza cuando el reloj del servidor indica el momento especificado por un diseñador en la planificación de la distribución de un reporte. El sistema hará llegar la información por correo electrónico a las personas que aparecen en la planificación.
Referencias:	R11
Precondiciones:	Tiene que existir alguna planificación.

Caso de Uso:	Crear Reporte de Evaluaciones
Actores:	Diseñador
Propósito:	Gestionar el diseño de un informe personalizado que contenga información de las evaluaciones.
Resumen:	El Caso de Uso comienza cuando el Diseñador solicita crear un reporte de evaluaciones. Este selecciona la información que desea incluir en su informe transitando por varios pasos de un asistente. Por último se registra el informe con el nombre, título y subtítulo que el usuario indique.
Referencias:	R4
Poscondiciones:	Queda almacenado el diseño de un nuevo reporte.

Caso de Uso:	Definir Permisos del Reporte
Actores:	Diseñador, Subsistema de Seguridad
Propósito:	Asegurar la información contenida en un determinado reporte.
Resumen:	El Caso de Uso comienza cuando el Diseñador solicita definir los permisos de acceso a un reporte diseñado por él. Este proceso es gestionado por el Subsistema de Seguridad de Akademos.

Referencias:	R10
Poscondiciones:	Quedan definidos los permisos del reporte.

Caso de Uso:	Consultar Gráfico
Actores:	Consultor
Propósito:	Permitir al usuario consultar los datos actualizados de un reporte representado en un gráfico que ya ha sido diseñado.
Resumen:	El Caso de Uso comienza cuando el Consultor solicita ver un gráfico, el sistema le muestra el gráfico con los datos actualizados del reporte al que pertenece y la opción de imprimirlo.
Referencias:	R12

2.7.2.3 Descripción detallada de los Casos de Uso

Caso de Uso:	Crear Reporte de Listado.	
Actores:	Diseñador.	
Propósito:	Gestionar el diseño de un informe personalizado que muestre una lista de estudiantes.	
Resumen:	El caso de uso comienza cuando el Diseñador solicita crear un reporte de listado. Este selecciona la información que desea incluir en su informe transitando por varios pasos de un asistente. Además indica el orden en que aparecerá dicha información. Por último se registra el informe con el nombre, título y subtítulo que el usuario indique.	
Referencias:	R2, Conformar Filtro(include).	
Acción del actor:	Respuesta del sistema:	
1. El diseñador solicita crear un nuevo informe o reporte de listado.	2. Muestra una lista de las plantillas donde está la información a la que tiene acceso y las opciones de "Adicionar" "Eliminar", "Ubicar	

CAPÍTULO II DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	<p>Primero”, “Ubicar Después”, “Intercambiar Sentido“, para realizar esas operaciones en la lista de los campos de esas plantillas que han sido seleccionados por el actor.</p> <p>Ver Anexo # 1.</p>
3. Selecciona las plantillas de las que desea obtener información.	4. Muestra los campos de dichas plantillas.
5. Selecciona los campos que desea mostrar en su reporte y presiona el botón Adicionar.	6. Añade a una lista los campos seleccionados por el Diseñador.
7. Presiona el botón Siguiente.	8. Ir al Caso de Uso: Conformar Filtro.
	<p>9. Muestra una pantalla para especificar el orden del informe con los campos seleccionados por el actor y las opciones de “Ubicar Primero”, “Ubicar Después”, “Intercambiar Sentido” para realizar esas operaciones en la lista de los campos por los que se va a ordenar.</p> <p>Ver Anexo # 3.</p>
10. Selecciona los campos con la misma prioridad con la que desea se ordene la información y presiona el botón Adicionar.	11. Agrega los campos seleccionados a una lista.
12. Presiona el botón Siguiente.	<p>13. Muestra un formulario solicitando el nombre, título y subtítulo con los que quedará registrado el reporte.</p> <p>Ver Anexo # 4.</p>
14. Teclea los datos solicitados y presiona el botón Terminar.	15. Almacena el diseño del nuevo reporte y muestra la página de inicio del sistema.
Precondiciones: El Diseñador tiene que haber iniciado el sistema.	
Poscondiciones: Queda añadido un nuevo reporte a la lista de reportes del sistema.	

Caminos Alternativos
<ul style="list-style-type: none"> • Si el Diseñador selecciona el botón Cancelar en cualquiera de los pasos el sistema regresa al paso 1 del flujo normal. • Si el Diseñador selecciona el botón Atrás en cualquiera de los pasos el sistema regresa al paso anterior del asistente. • Si en el paso 7 el Diseñador no ha especificado ningún campo, el sistema muestra un mensaje de error, indicándole que debe señalar al menos una.

Caso de Uso:	Crear Reporte Cruzado.	
Actores:	Diseñador.	
Propósito:	Gestionar el diseño de un informe personalizado que muestre una tabla con valores que representan la cantidad total de la intersección de dos datos.	
Resumen: El caso de uso comienza cuando el Diseñador solicita crear un reporte cruzado. Este selecciona los datos que desea tener en las filas y las columnas de su tabla respectivamente transitando por varios pasos de un asistente. Por último se registra el informe con el nombre, título y subtítulo que el usuario indique.		
Referencias: R3, Conformar Filtro(include).		
Acción del actor:	Respuesta del sistema:	
1. El actor solicita crear un nuevo informe o reporte cruzado.	2. Muestra la lista de las plantillas donde está la información a la que tiene acceso.	
3. Selecciona la plantilla que contiene el campo que se mostrará en las filas.	4. Muestra los campos numéricos de dicha plantilla.	
5. Selecciona el campo que se mostrará en las filas.		
6. Selecciona la plantilla que contiene el campo que se mostrará en las columnas.	7. Muestra los campos numéricos de dicha plantilla.	
	Ver Anexo # 5.	

8. Selecciona el campo que se mostrará en las columnas y presiona el botón Siguiente.	9. Ir al Caso de Uso: Conformar Filtro.
	10. Muestra un formulario solicitando el nombre, título y subtítulo con los que quedará registrado el reporte. Ver Anexo # 6.
11. Teclea los datos solicitados y presiona el botón Terminar.	12. Almacena el diseño del nuevo reporte y muestra la página de inicio del sistema.
Precondiciones: El actor tiene que haber iniciado el sistema.	
Poscondiciones: Queda añadido un nuevo reporte a la lista de reportes del sistema.	
<p>Caminos Alternativos</p> <ul style="list-style-type: none"> • Si el Diseñador selecciona el botón Cancelar en cualquiera de los pasos el sistema regresa al paso 1 del flujo normal. • Si el Diseñador selecciona el botón Atrás en cualquiera de los pasos el sistema regresa al paso anterior del asistente. • Si en el paso 8 el Diseñador no especificó los campos correspondientes a las filas y las columnas, el sistema muestra un mensaje de error indicando que se requieren estos datos. 	

Caso de Uso:	Crear Gráfico.
Actores:	Diseñador.
Propósito:	Gestionar el diseño de un gráfico personalizado que representa los datos de un reporte seleccionado por el actor.
Resumen: El caso de uso comienza cuando el Diseñador solicita crear el gráfico de un reporte cruzado. El sistema le muestra un asistente a través del cual el actor selecciona el tipo de gráfico que desea crear, qué datos del reporte desea representar y las características visuales del gráfico.	
Referencias: R8	

Acción del actor:	Respuesta del sistema:
1. El diseñador solicita crear el gráfico de un reporte cruzado señalado por él.	2. Muestra una pantalla con los tipos de gráficos que puede crear. Ver Anexo # 13.
3. Selecciona el tipo de gráfico que desea.	4. Muestra una imagen de demostración con un gráfico del tipo seleccionado.
5. Presiona el botón Siguiente.	6. Muestra la opción de seleccionar las series de su gráfico: las filas o las columnas del reporte.
7. Selecciona una de ellas.	8. Muestra los valores que encabezan las filas o las columnas del reporte de acuerdo a lo seleccionado.
9. Selecciona las que desea representar en su gráfico y presiona el botón Adicionar.	10. Añade a una lista estos valores. Ver Anexo # 14.
11. Presiona el botón Siguiente.	12. Muestra un formulario solicitando que se indique si aparecerá o no la leyenda, el título del eje x y el título del eje y. Ver Anexo # 15.
13. Indica los datos solicitados y presiona el botón Siguiente.	14. Muestra un formulario solicitando el nombre, título y subtítulo con los que quedará registrado el gráfico. Ver Anexo # 16.
15. Teclea los datos solicitados y presiona el botón Terminar.	16. Almacena el diseño del nuevo gráfico y muestra la página de inicio del sistema.
Precondiciones: El Diseñador tiene que haber iniciado el sistema y tiene que existir un reporte señalado.	
Poscondiciones: Queda añadido un nuevo gráfico a la lista de gráficos del sistema.	
Caminos Alternativos	

- Si el Diseñador selecciona el botón Cancelar en cualquiera de los pasos el sistema regresa al paso 1 del flujo normal.
- Si el Diseñador selecciona el botón Atrás en cualquiera de los pasos el sistema regresa al paso anterior del asistente.
- En el paso 9 si el Diseñador no especifica ninguna serie el sistema le muestra un mensaje de error indicándole que este dato se requiere.

Caso de Uso:	Actualizar Reportes.
Actores:	Diseñador.
Propósito:	Permitir al usuario eliminar y modificar reportes.
Resumen: El caso de uso comienza cuando el Diseñador solicita eliminar o modificar algún reporte. En el caso de las modificaciones el sistema le muestra los datos actuales del reporte para que el actor realice los cambios.	
Referencias: R6	
Acción del actor:	Respuesta del sistema:
1. El diseñador solicita actualizar la lista de los reportes creados por él, eligiendo la opción de eliminar o modificar un reporte.	2. Si eligió: <ul style="list-style-type: none"> a. Eliminar: ir a la sección Eliminar Reporte. b. Modificar: ir a la sección Modificar Reporte.
Sección: Eliminar Reporte	
1. Selecciona la opción Eliminar.	2. Muestra un mensaje de confirmación para eliminar el reporte. Ver Anexo # 12.
3. Presiona el botón "Sí".	4. Elimina el reporte señalado.
Sección: Modificar Reporte	
1. Selecciona la opción Modificar.	2. Muestra los datos actuales del reporte indicado.
3. Introduce los nuevos datos y presiona el botón Aceptar.	4. Almacena los nuevos datos.

Precondiciones: El actor tiene que haber iniciado el sistema.
Poscondiciones: Queda actualizada la lista de reportes del sistema.
Caminos Alternativos <ul style="list-style-type: none"> Si el Diseñador selecciona el botón Cancelar en el paso 3 de la sección Modificar Reporte el sistema regresa al inicio del Caso de Uso. Si el Diseñador selecciona el botón “No” en el paso 3 de la sección Eliminar Reporte el sistema cancela la acción y regresa al inicio del Caso de Uso.

Caso de Uso:	Actualizar Carpetas.
Actores:	Diseñador.
Propósito:	Actualizar las carpetas del sistema.
Resumen: El caso de uso comienza cuando el Diseñador solicita crear o eliminar alguna carpeta o cuando solicita mover un reporte hacia una de ellas. Si desea crear una carpeta o mover algún reporte el sistema le da la opción de seleccionar la ubicación.	
Referencias: R7	
Acción del actor:	Respuesta del sistema:
1. El diseñador solicita actualizar las carpetas, eligiendo una de las siguientes opciones: Crear Carpeta, Eliminar Carpeta, Mover.	2. Si elige: <ul style="list-style-type: none"> a. Crear Carpeta: ir a la sección Crear Carpeta. b. Eliminar Carpeta: ir a la sección Eliminar Carpeta. c. Mover: ir a la sección Mover.
Sección: Crear Carpeta	
1. Selecciona la ubicación donde desea crear la carpeta y la opción Crear Carpeta.	2. Muestra una pantalla solicitando el nombre con el cual la carpeta será creada y el camino de la ubicación señalada por el actor.

	Ver Anexo # 9.
3. Introduce el dato solicitado y presiona el botón Aceptar.	4. Comprueba si no existe una carpeta con ese nombre en la ubicación indicada, crea la carpeta y regresa al inicio del Caso de Uso.
Sección: Eliminar Carpeta	
1. Selecciona la opción Eliminar Carpeta.	2. Muestra un mensaje de confirmación para eliminar la carpeta. Ver Anexo # 10.
3. Presiona el botón "Sí".	4. Elimina la carpeta señalada con todo su contenido, tanto reportes como otras carpetas y regresa al inicio del Caso de Uso.
Sección: Mover	
1. Selecciona un reporte o carpeta, la ubicación a donde desea trasladarlo y la opción Mover.	2. Muestra una pantalla con todas las ubicaciones donde puede estar el elemento seleccionado. Ver Anexo # 11.
3. Selecciona una de ellas y presiona el botón Aceptar.	4. Comprueba si existe un elemento con ese nombre en la ubicación destino y elimina el reporte o carpeta de la ubicación actual y lo inserta en la que ha sido indicada por el actor.
<p>Precondiciones: El actor tiene que haber iniciado el sistema.</p> <p>Para que se ejecute la sección Mover tiene que haber un reporte o una carpeta señalados.</p> <p>Para que se ejecute la sección Eliminar Carpeta tiene que haber una carpeta señalada.</p>	
<p>Poscondiciones: Quedan actualizadas las carpetas del sistema.</p>	
<p>Caminos Alternativos</p> <ul style="list-style-type: none"> • Si el Diseñador selecciona el botón Cancelar cualquiera de las secciones el sistema regresa al inicio del Caso de Uso. • En el paso 4 de la sección Crear Carpeta si el sistema comprueba que ya existe una carpeta con ese nombre en esa ubicación emite un mensaje de error al usuario. 	

- En el paso 3 de la sección Eliminar Carpeta si el actor selecciona “No” se regresa al inicio del Caso de Uso.
- En el paso 4 de sección Mover si el sistema comprueba que ya existe un elemento con ese nombre en esa ubicación emite un mensaje de error al usuario.

Caso de Uso:	Iniciar Sistema.	
Actores:	Consultor.	
Propósito:	Permitir al Consultor iniciar el sistema.	
Resumen: El Consultor solicita entrar al sistema y se le muestra la página de inicio.		
Referencias: R1		
Acción del actor:	Respuesta del sistema:	
1. El actor solicita entrar al sistema.	2. Muestra una pantalla con un explorador de los reportes que han sido diseñados distribuidos en carpetas.	

Caso de Uso:	Consultar Reporte.	
Actores:	Consultor.	
Propósito:	Permitir al usuario consultar los datos actualizados de un reporte que ya ha sido diseñado.	
Resumen: El actor solicita consultar un reporte señalado por él. El sistema le muestra los datos actualizados y las opciones de imprimir, graficar y exportar.		
Referencias: R5		
Acción del actor:	Respuesta del sistema:	
1. El actor solicita consultar los datos de un reporte que ha sido diseñado. Ver Anexo # 7	1. Muestra los datos actualizados de ese reporte y las opciones de imprimir, exportar y ver gráfico. Ver Anexo # 8.	

CAPÍTULO II DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	3. Si selecciona: a. imprimir: ir a la sección: Imprimir. b. exportar: ir a la sección: Exportar. c. graficar: ir a la sección: Ver Gráfico.
Sección: Imprimir	
1. Solicita imprimir los datos del reporte que ha sido mostrado.	2. Muestra un formulario solicitando los datos de impresión.
3. Introduce los datos solicitados y presiona el botón Imprimir.	4. Manda a imprimir el reporte.
Sección: Exportar	
1. Solicita exportar el reporte que ha sido mostrado.	2. Muestra las herramientas a las que puede exportar el reporte: Word y Excel.
3. Selecciona una de ellas y presiona el botón Exportar.	4. Muestra el reporte en la herramienta seleccionada.
Sección: Graficar	
1. Solicita graficar el reporte.	2. Muestra todos los datos del reporte en un gráfico definido por el sistema.
Precondiciones: El actor tiene que haber iniciado el sistema.	
Caminos Alternativos	
<ul style="list-style-type: none"> • Si el Diseñador selecciona el botón Cancelar en el paso 3 de las secciones Imprimir y Exportar, el sistema regresa al paso 2 del flujo normal del Caso de Uso. • Si en el paso 3 el actor no selecciona ninguna de las opciones termina el Caso de Uso. 	

Caso de Uso:	Planificar Distribución.
Actores:	Distribuidor.
Propósito:	Gestionar la distribución de los reportes de acuerdo a la necesidad de información de los usuarios.
Resumen:	El Distribuidor solicita planificar la distribución de un reporte determinado a

través del correo electrónico y el sistema le da la posibilidad de hacerlo de una manera personalizada.

Referencias: R9

Acción del actor:	Respuesta del sistema:
1. El actor solicita planificar la distribución de un determinado reporte.	2. Muestra una pantalla solicitando al usuario el destinatario, el asunto del mensaje portador del reporte, la fecha y la hora de la distribución, y la opción de indicar si esta será periódica o no. Ver Anexo # 17.
3. Introduce los datos solicitados e indica que la distribución será periódica.	4. Muestra los tipos de frecuencia que puede tener la distribución(diario, semanal, mensual o anual).
5. Selecciona una de ellas y presiona el botón Aceptar.	6. Comprueba si todos los datos fueron introducidos correctamente y almacena los datos de la distribución.

Precondiciones: El actor tiene que haber iniciado el sistema y tiene que existir un reporte señalado.

Poscondiciones: Queda almacenada una nueva planificación de la distribución de un reporte.

Camino Alternativos

- Si el usuario selecciona Cancelar en el paso 5 se regresa al inicio del Caso de Uso.
- En el paso 6 si el sistema detecta que todos los datos no fueron introducidos correctamente emite un mensaje de error informándolo al usuario.

Caso de Uso:	Conformar Filtro
Actores:	Diseñador.
Propósito:	Gestionar la construcción del filtro de cualquier reporte que se diseñe.

Resumen: El Caso de Uso comienza cuando en los asistentes para crear los diferentes tipos de reportes aparece la pantalla para que el Diseñador especifique el filtro de la información que se va a mostrar. El actor conforma la expresión lógica con la ayuda de paréntesis y operadores lógicos y de comparación.

Referencias: R2.2, 3.2, 4.2

Acción del actor:	Respuesta del sistema:
	<p>1. Muestra una pantalla para especificar el filtro a aplicar al reporte con las plantillas y los datos de cada una de ellas, así como las operaciones de filtro que puede aplicar: “igual a”, “mayor que” o “menor que” un valor determinado y las opciones de “Abrir Paréntesis”, “Cerrar Paréntesis”, “Y” y “O” para conformar la expresión lógica. Además muestra los botones: Siguiente, Anterior y Cancelar.</p> <p>Ver Anexo # 2.</p>
<p>2. Selecciona uno a uno los campos por los que desea filtrar la información.</p>	<p>3. Muestra los valores que puede tomar el campo seleccionado por el actor, si son constantes.</p>
<p>4. Indica la operación de filtro y el valor correspondiente. Presiona el botón Agregar.</p>	<p>5. Muestra el filtro conformado por todos los datos proporcionados por el actor:</p> <p>Plantilla.Campo operador valor</p>
<p>6. Presiona el botón Siguiente.</p>	<p>7. Comprueba si el filtro está correctamente formado, lo almacena y regresa a uno de los siguientes pasos:</p> <p>Paso 9 del Caso de Uso: Crear Reporte de Listado.</p> <p>Paso 10 del Caso de Uso: Crear Reporte Cruzado.</p>

Precondiciones: El actor tiene que haber iniciado cualquiera de los siguientes Casos de Uso: Crear Reporte de Listado, Crear Reporte Cruzado, Crear Reporte de Evaluaciones.

Poscondiciones: Queda almacenado el filtro de un nuevo reporte.

Caminos Alternativos

- En el paso 2 si el presiona el botón Siguiente termina el Caso de Uso y el sistema regresa a cualquiera de los siguientes pasos:

Paso 9 del Caso de Uso: Crear Reporte de Listado.

Paso 10 del Caso de Uso: Crear Reporte Cruzado.

- En el paso 7 si el sistema comprueba que el filtro no está lógicamente conformado emite un mensaje de error al usuario para que este lo rectifique.

2.8 Conclusiones

Al término de este capítulo, se llega a la conclusión de que no es necesario la existencia de un negocio que sirva como guía en el desarrollo de un sistema que cumpla verdaderamente con los objetivos planteados. Sin embargo, sí es preciso que se investigue el entorno en que se utilizará, con el objetivo de obtener puntos de partida que apunten a la necesidad de llevarlo a cabo. Además, se entiende que existen sistemas cuya dimensión no requiere subdividirlos en paquetes, pues ellos mismos constituyen subsistemas.

CAPÍTULO III CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

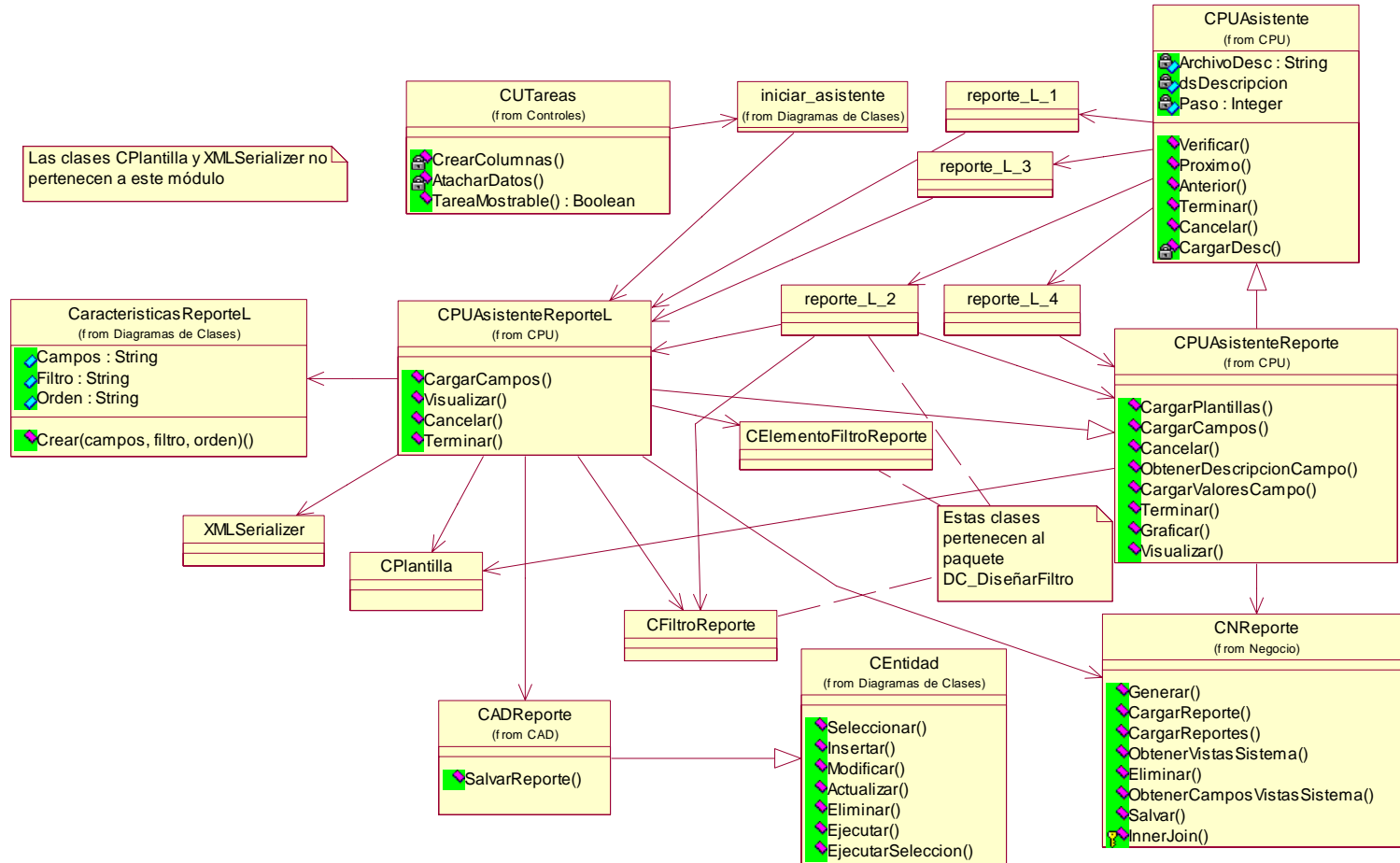
3.1 Introducción

Los Diagramas de Interacción correspondiente a cada realización de los casos de uso, proporcionan un gran nivel de detalle a la descripción de los mismos y reflejan las responsabilidades que tienen los objetos de las clases representadas en los diagramas que llevan ese nombre. De estos últimos, se pueden extraer las clases persistentes, que se relacionarán también en un diagrama, el cual dará lugar al Modelo de Datos. Todos estos aspectos aparecen en este capítulo y constituyen la construcción de la solución propuesta, junto a la explicación de los principios de diseño y los estándares de codificación empleados en el sistema.

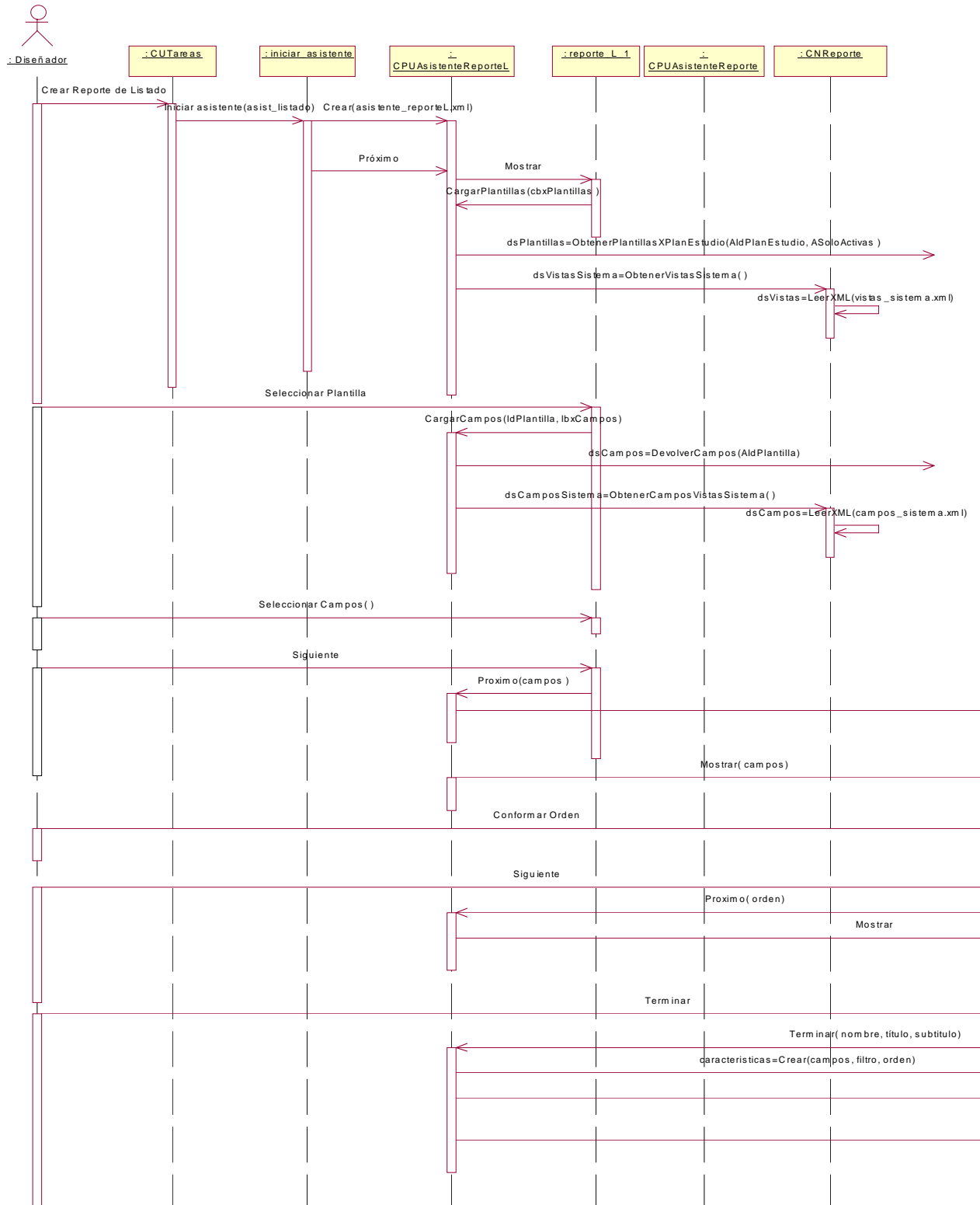
3.2 Diagramas de Clases y de Interacción

3.2.1 Paquete: Crear Reporte de Listado

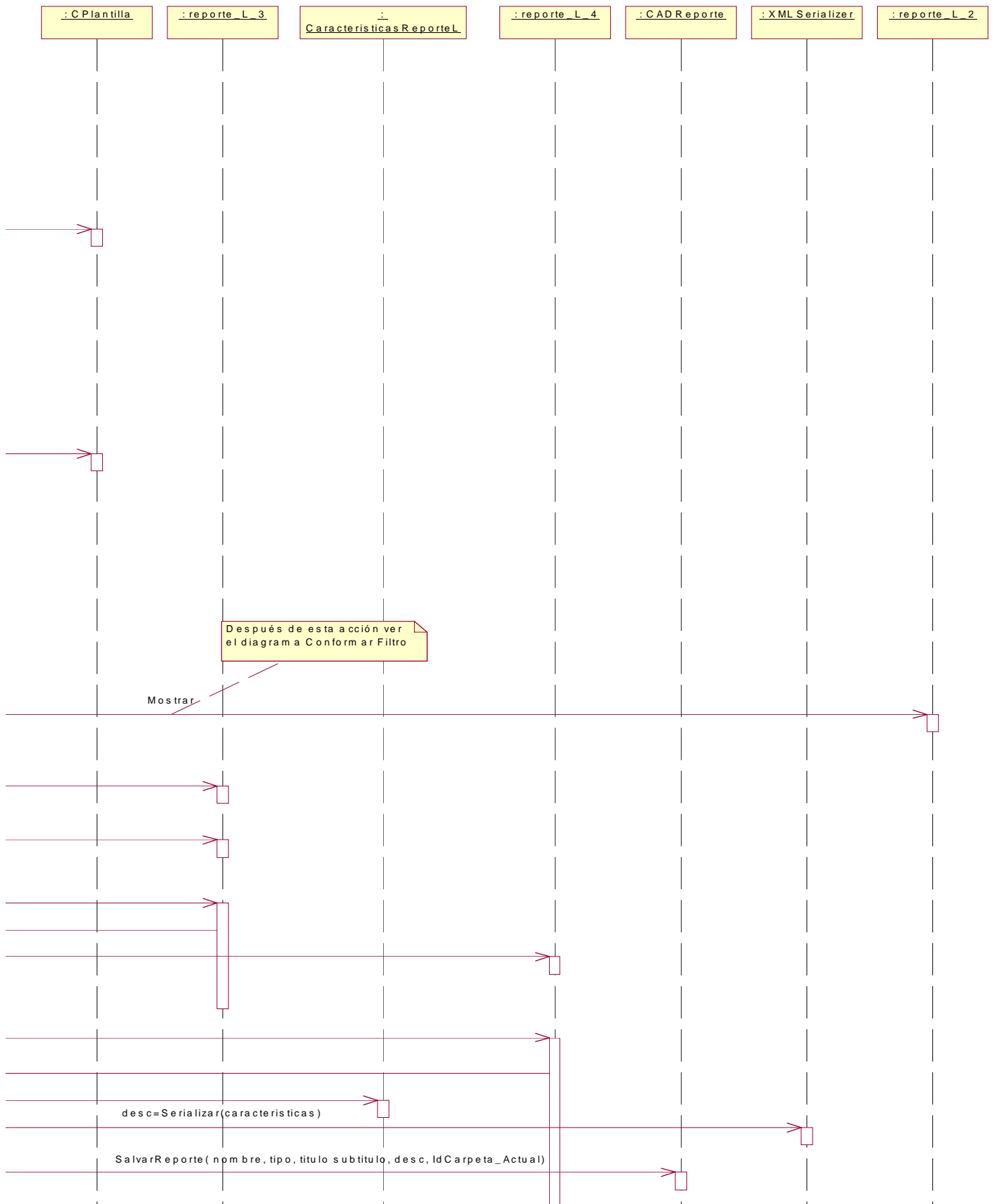
3.2.1.1 Diagrama de Clase: Crear Reporte de Listado



3.2.1.2 Diagrama de Secuencia: Crear Reporte de Listado

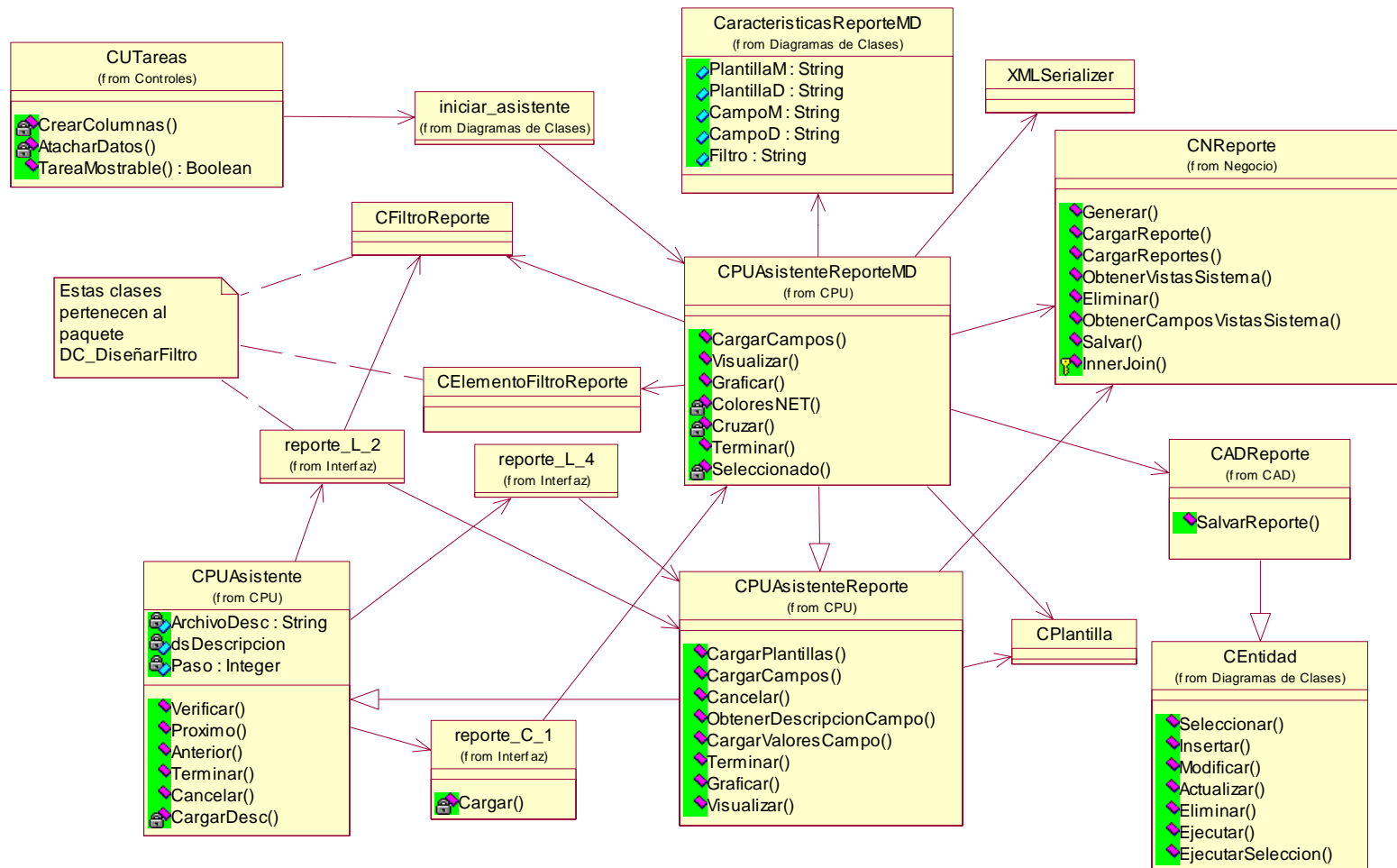


CAPÍTULO III CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

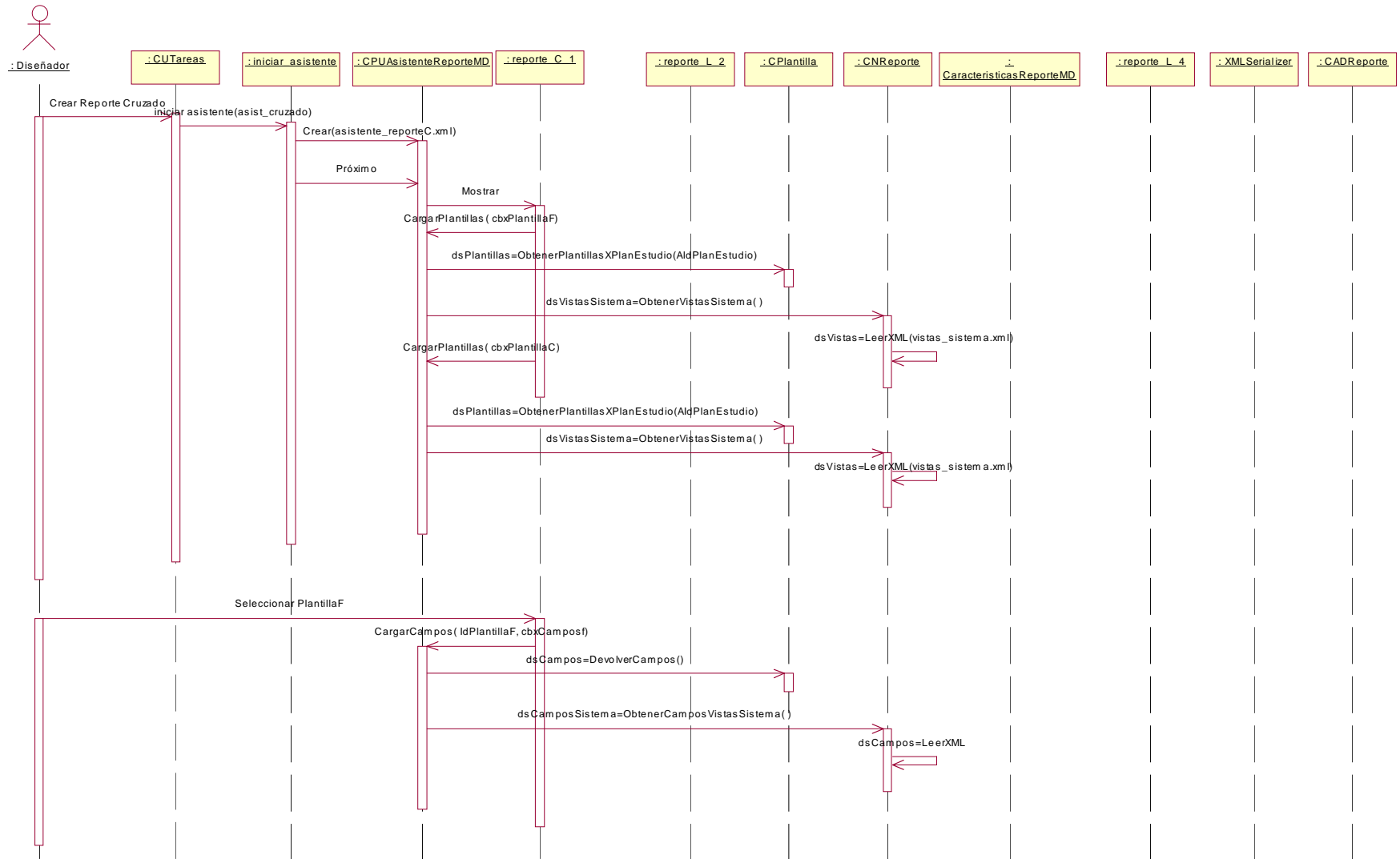


3.2.2 Paquete: Crear Reporte Cruzado

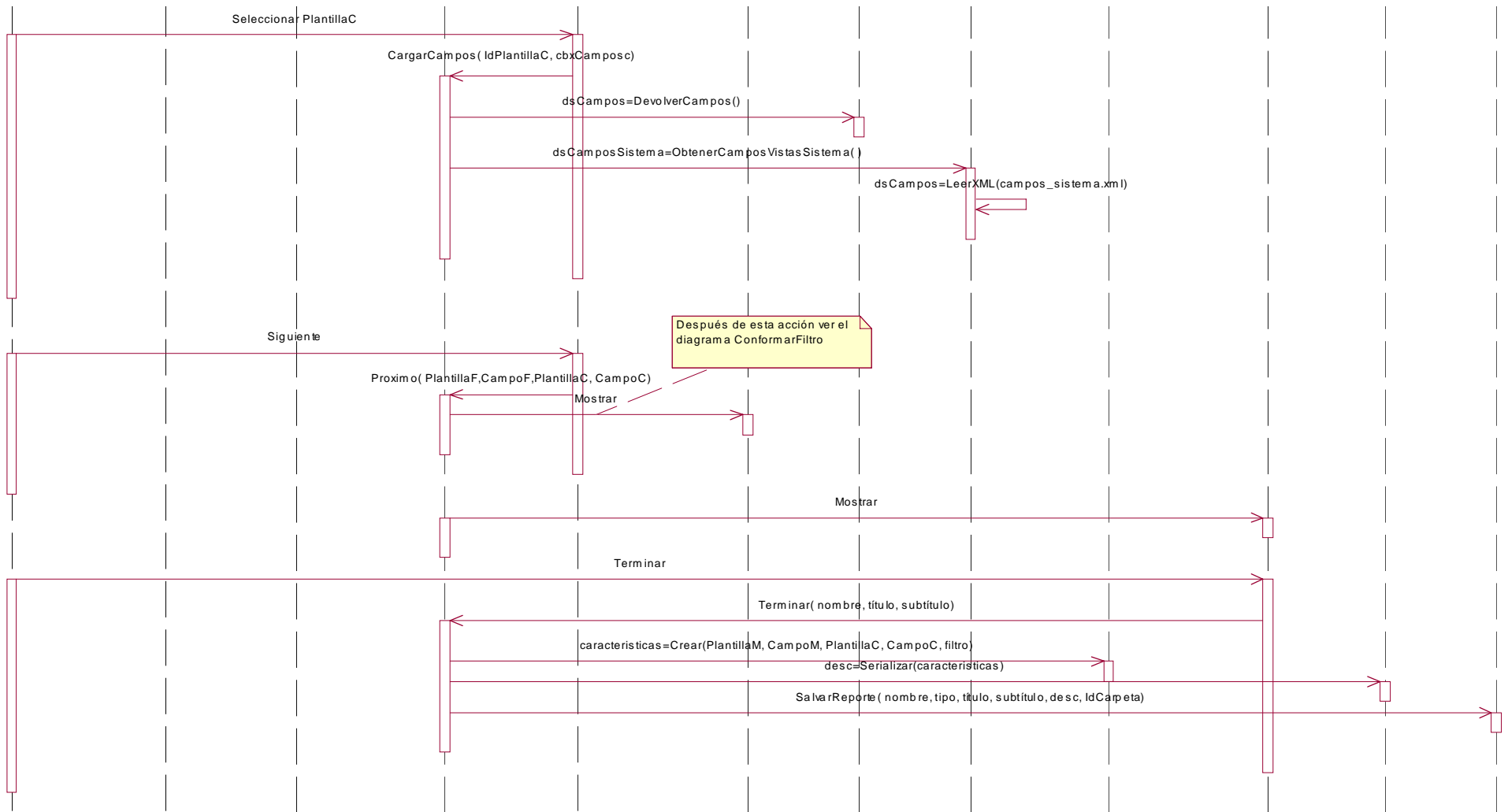
3.2.2.1 Diagrama de Clases: Crear Reporte Cruzado



3.2.2.2 Diagrama de Secuencia: Crear Reporte Cruzado

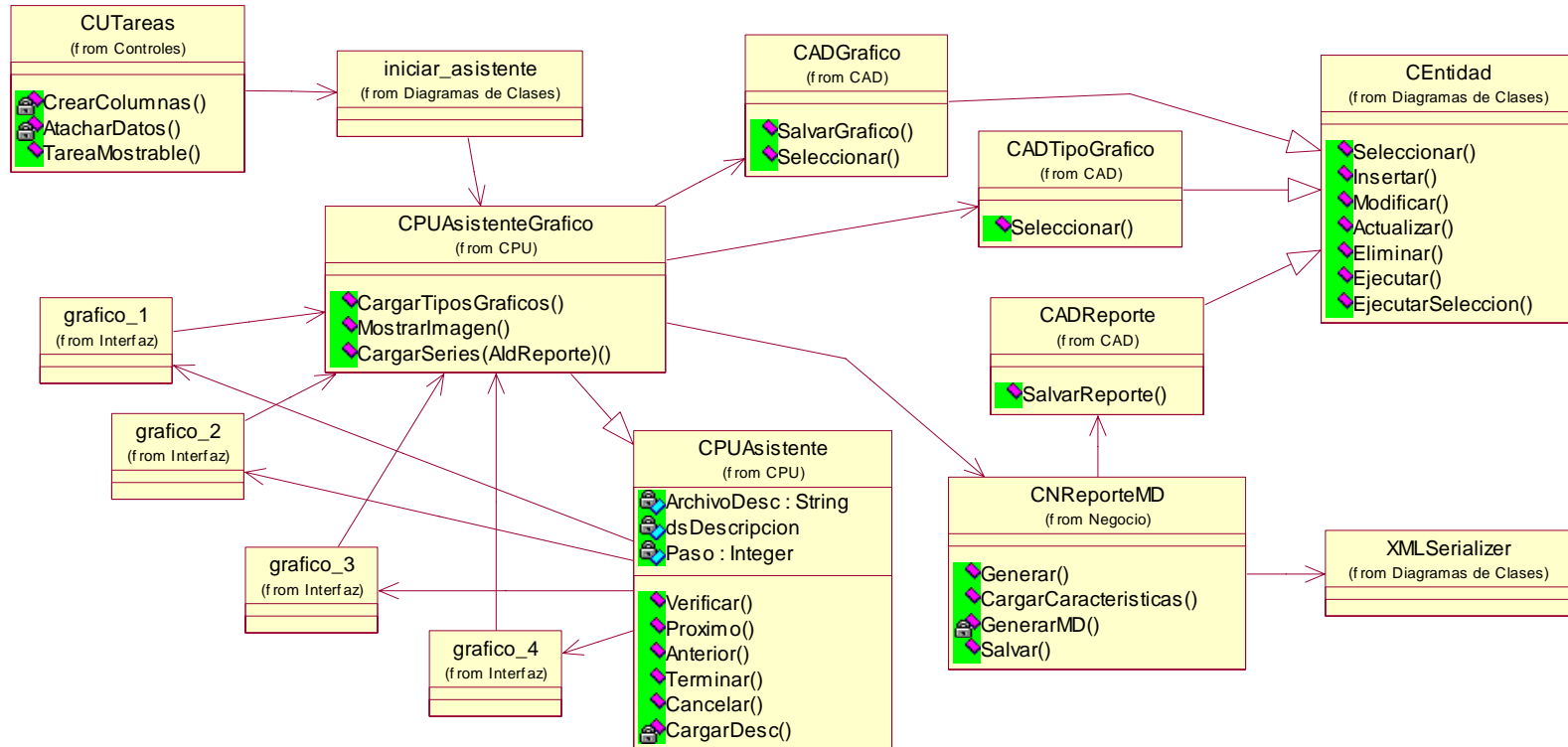


CAPÍTULO III CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

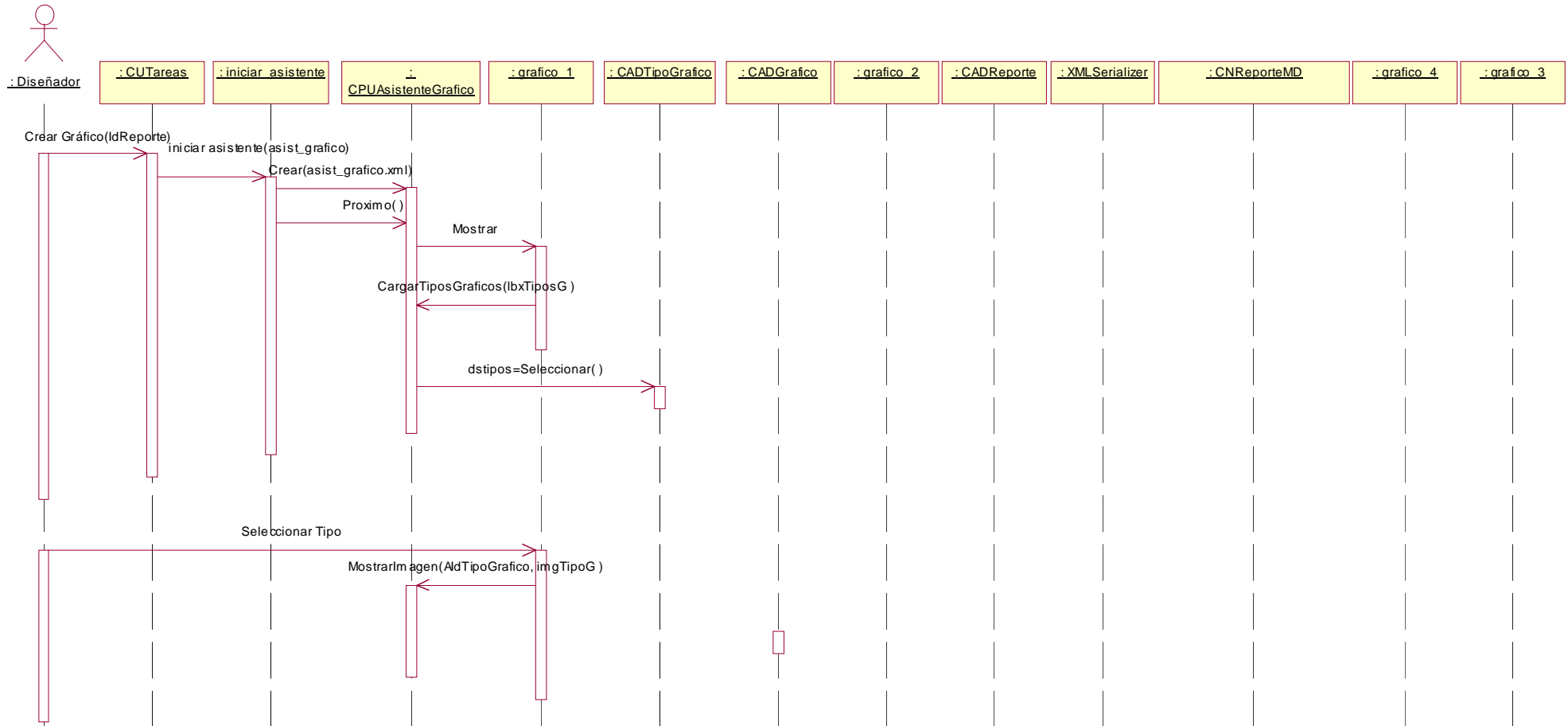


3.2.3 Paquete: Crear Gráfico

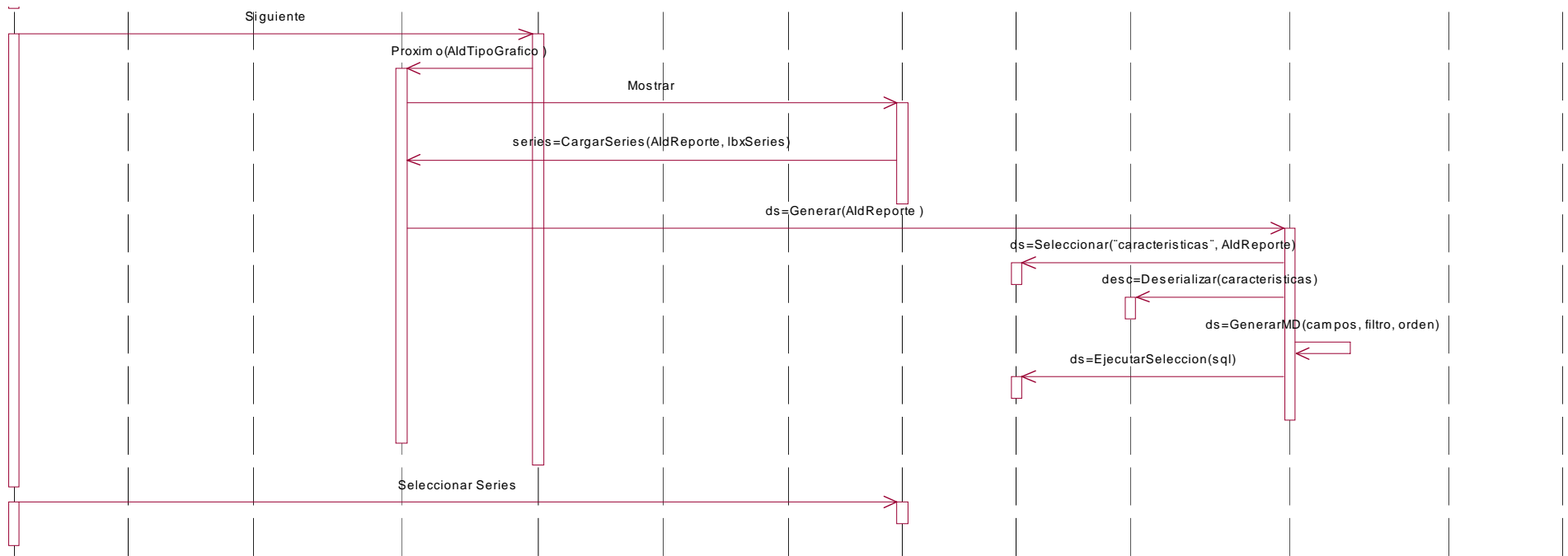
3.2.3.1 Diagrama de Clases: Crear Gráfico



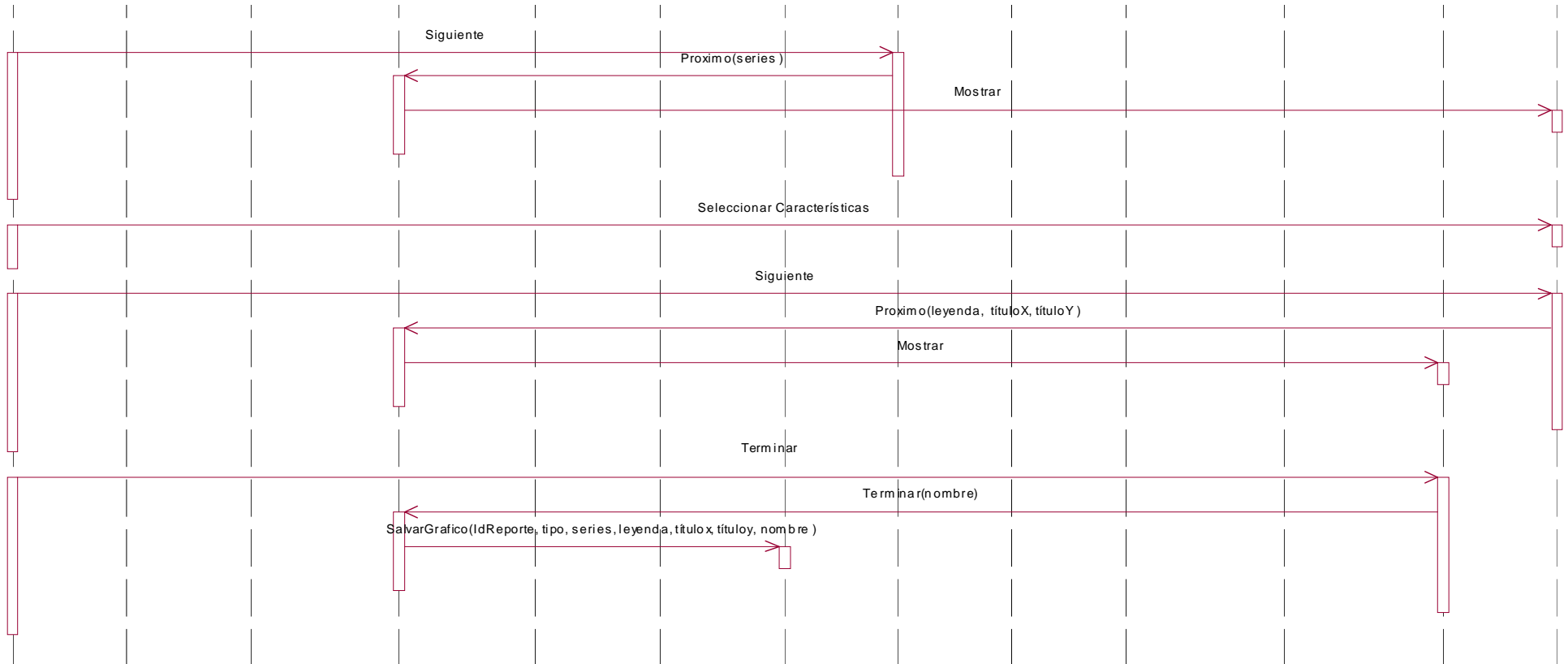
3.2.3.2 Diagrama de Secuencia: Crear Gráfico



CAPÍTULO III CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

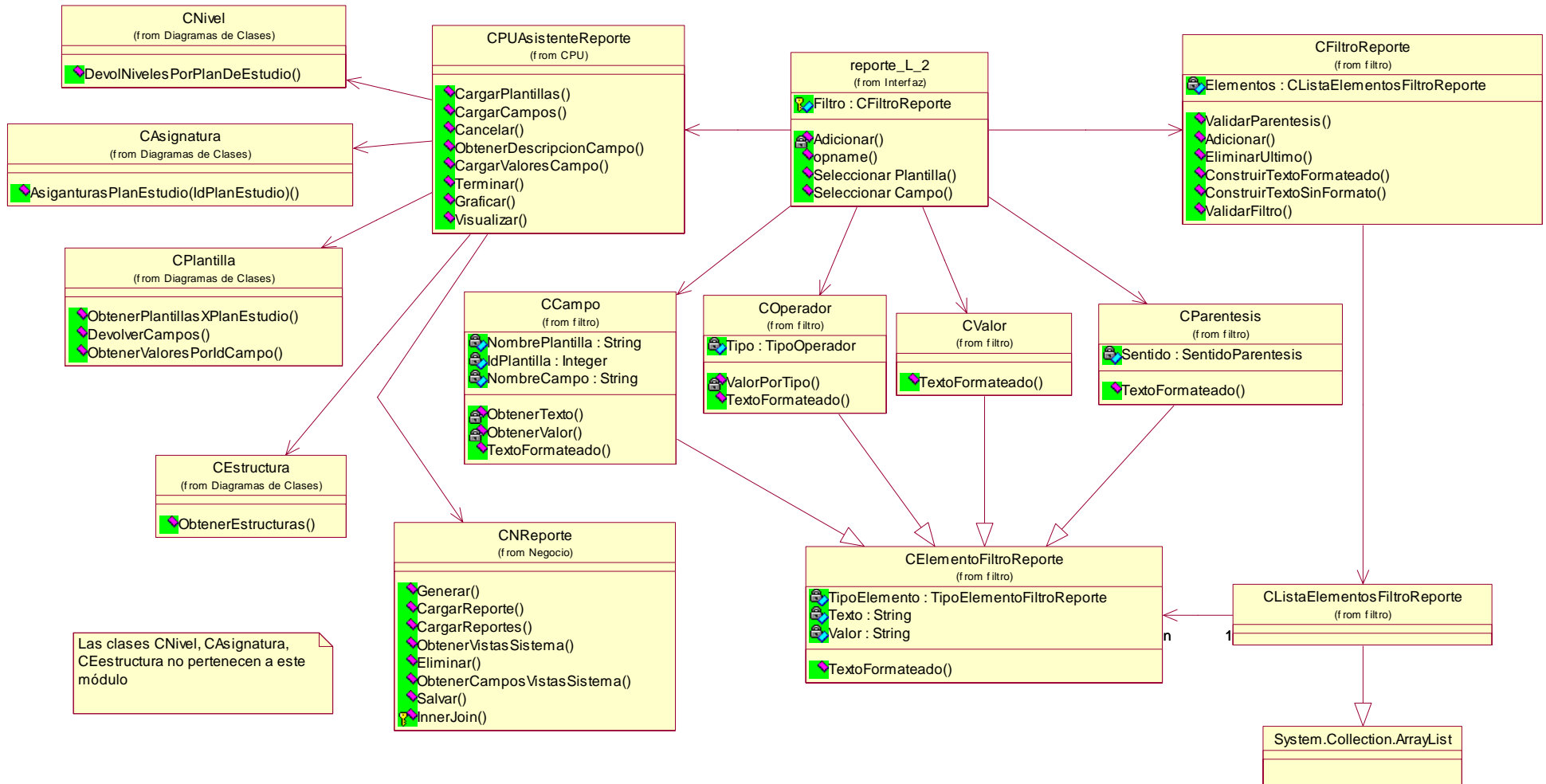


CAPÍTULO III CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

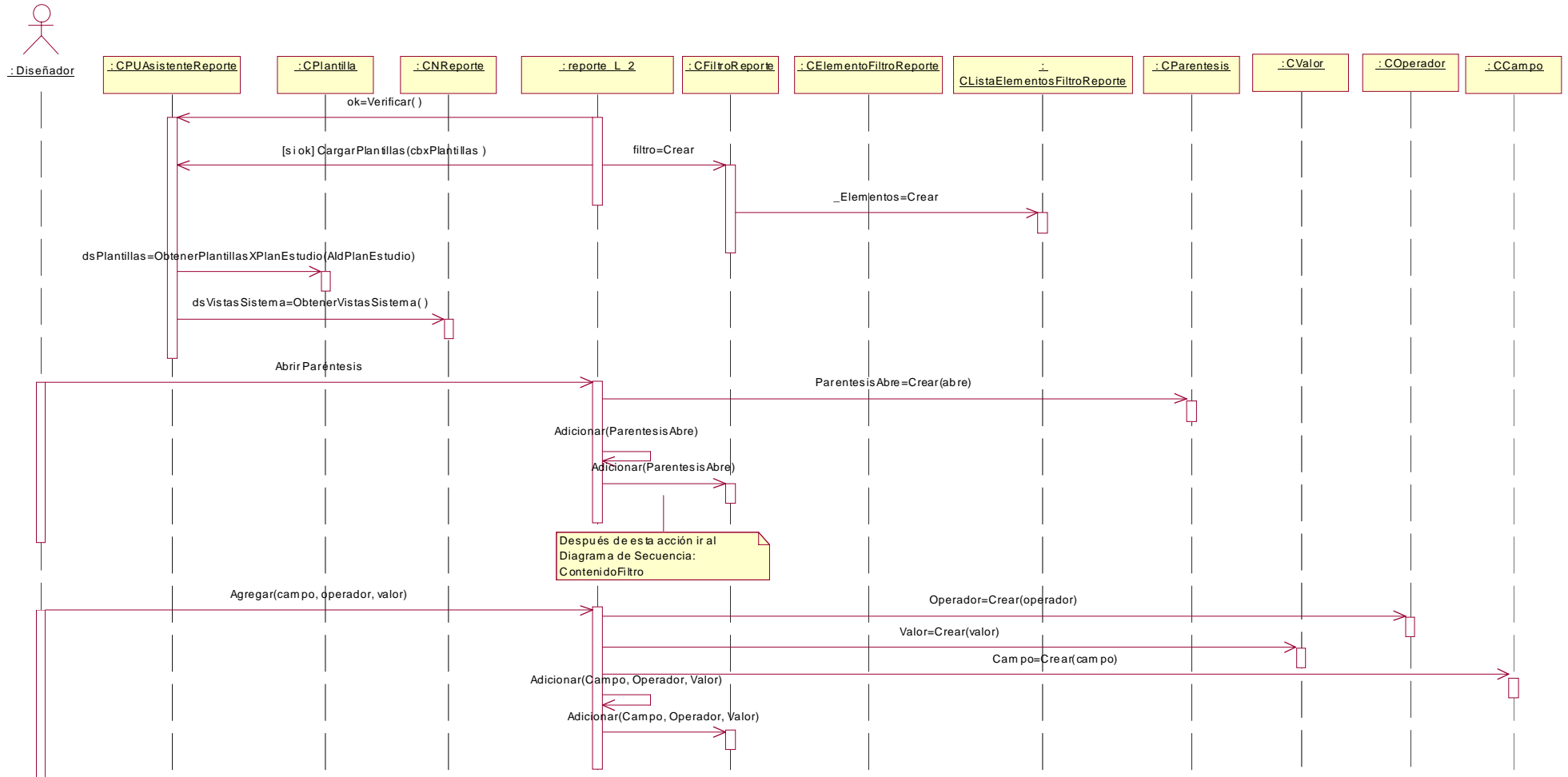


3.2.4 Paquete: Conformar Filtro

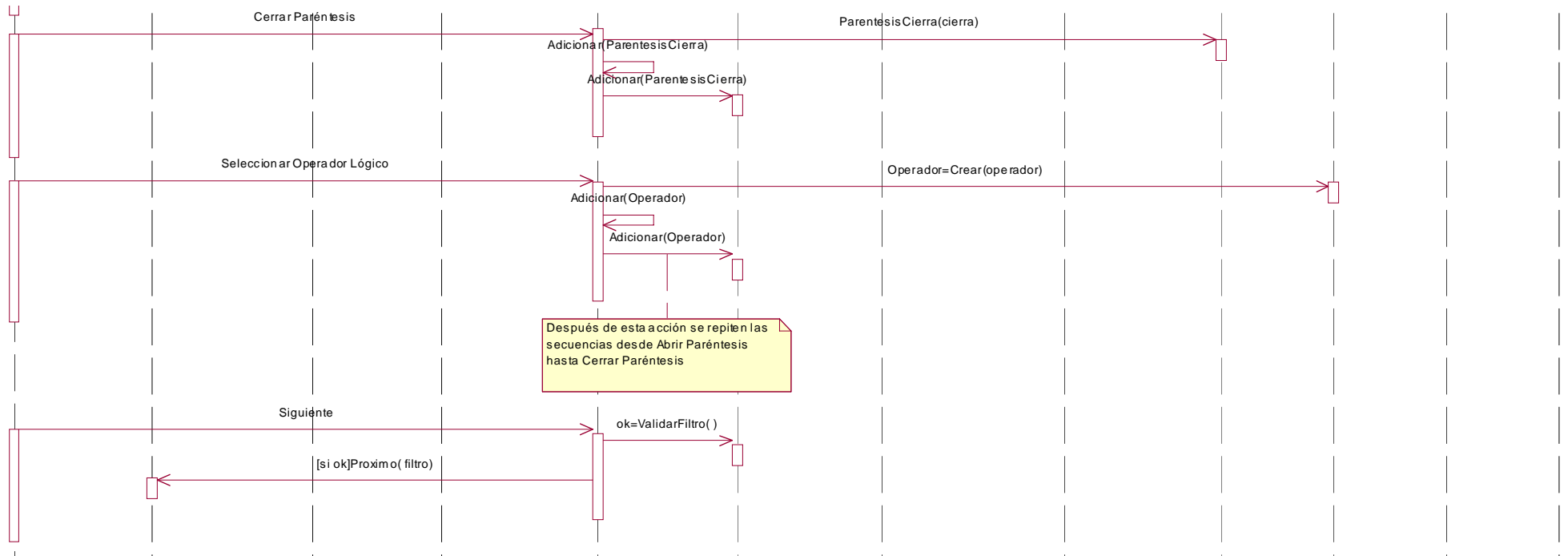
3.2.4.1 Diagrama de Clases: Conformar Filtro



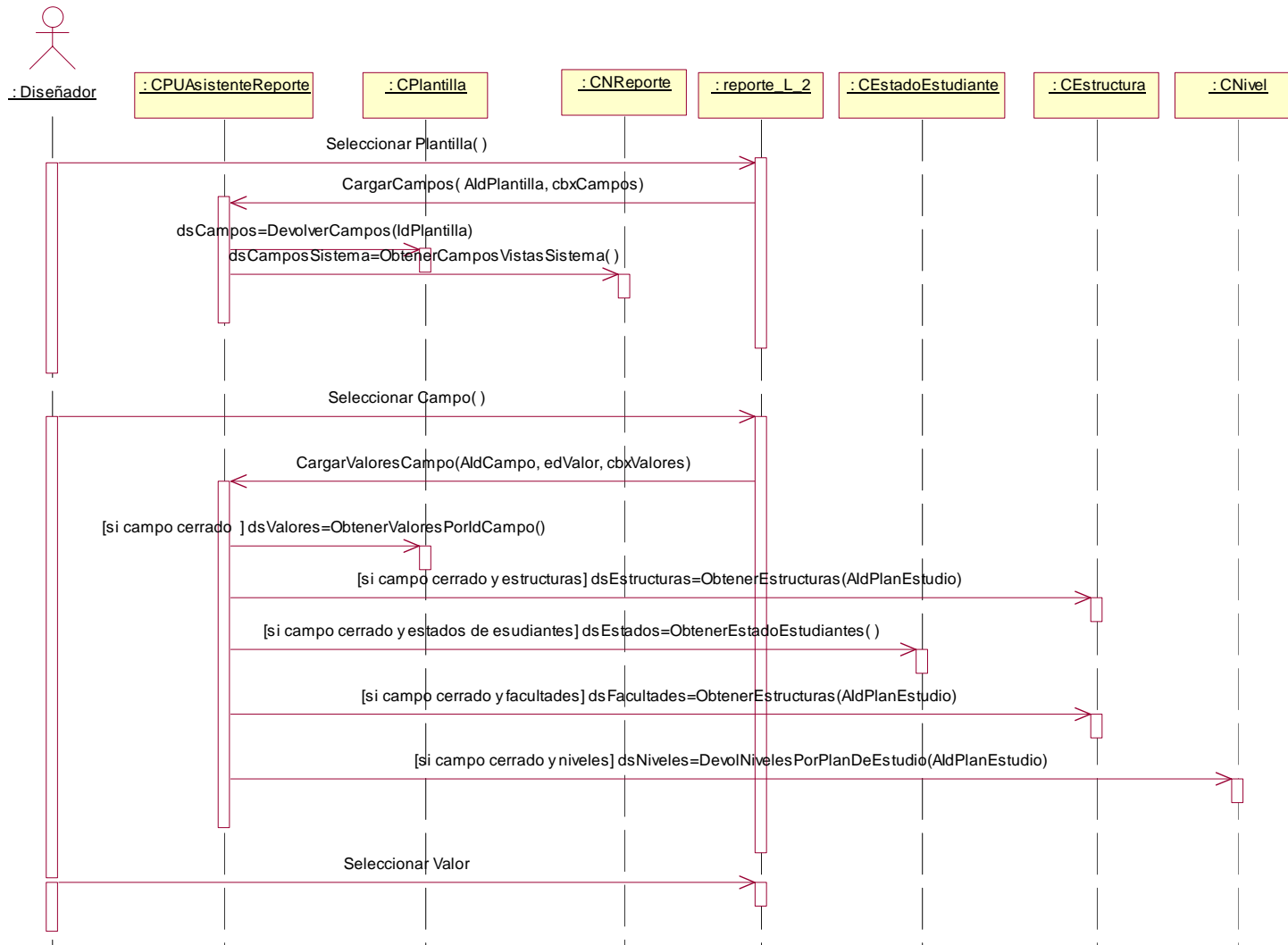
3.2.4.2 Diagrama de Secuencia: Conformar Filtro



CAPÍTULO III CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

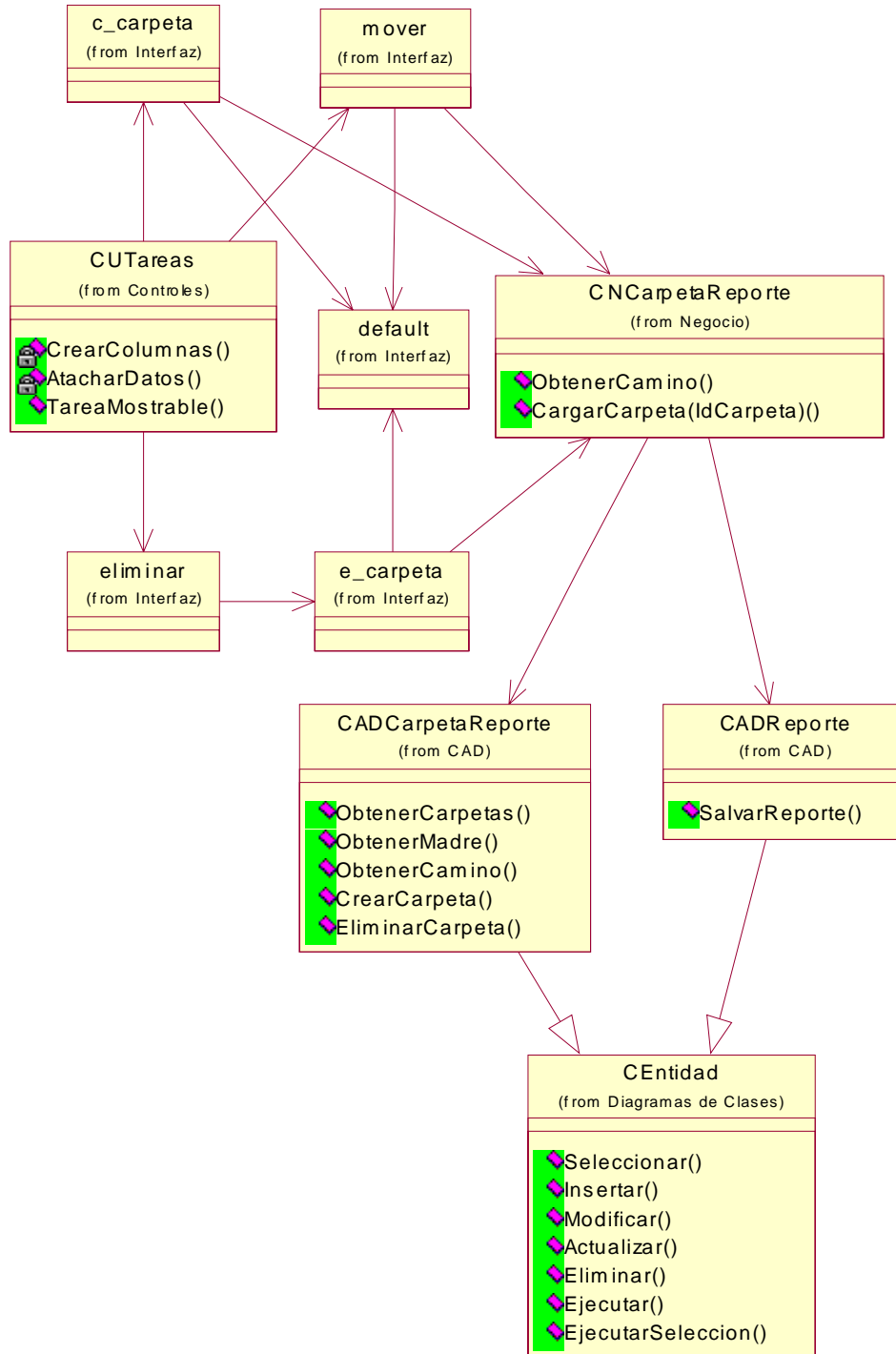


3.2.4.3 Diagrama de Secuencia: Contenido Filtro

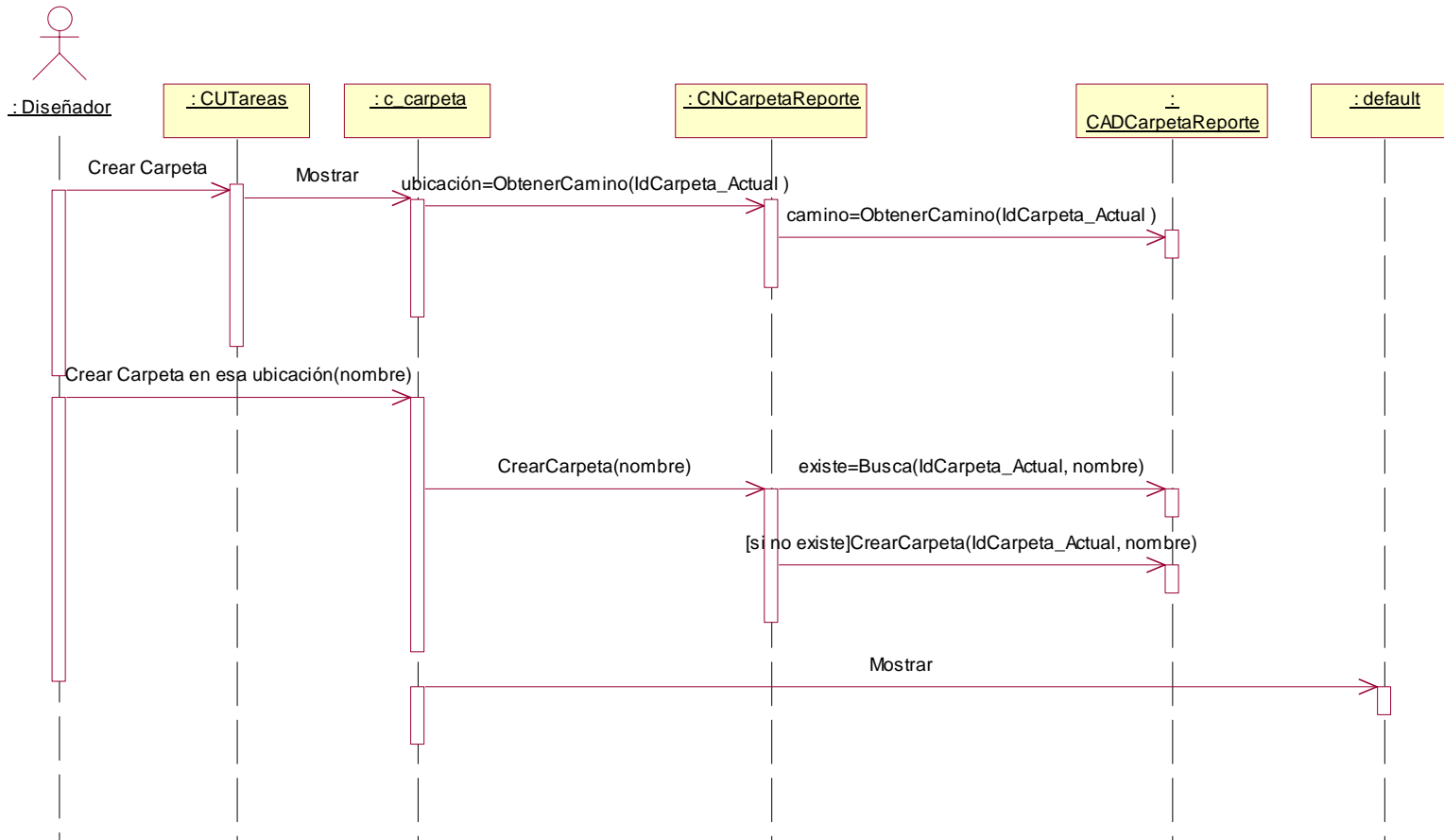


3.2.5 Paquete: Actualizar Carpetas

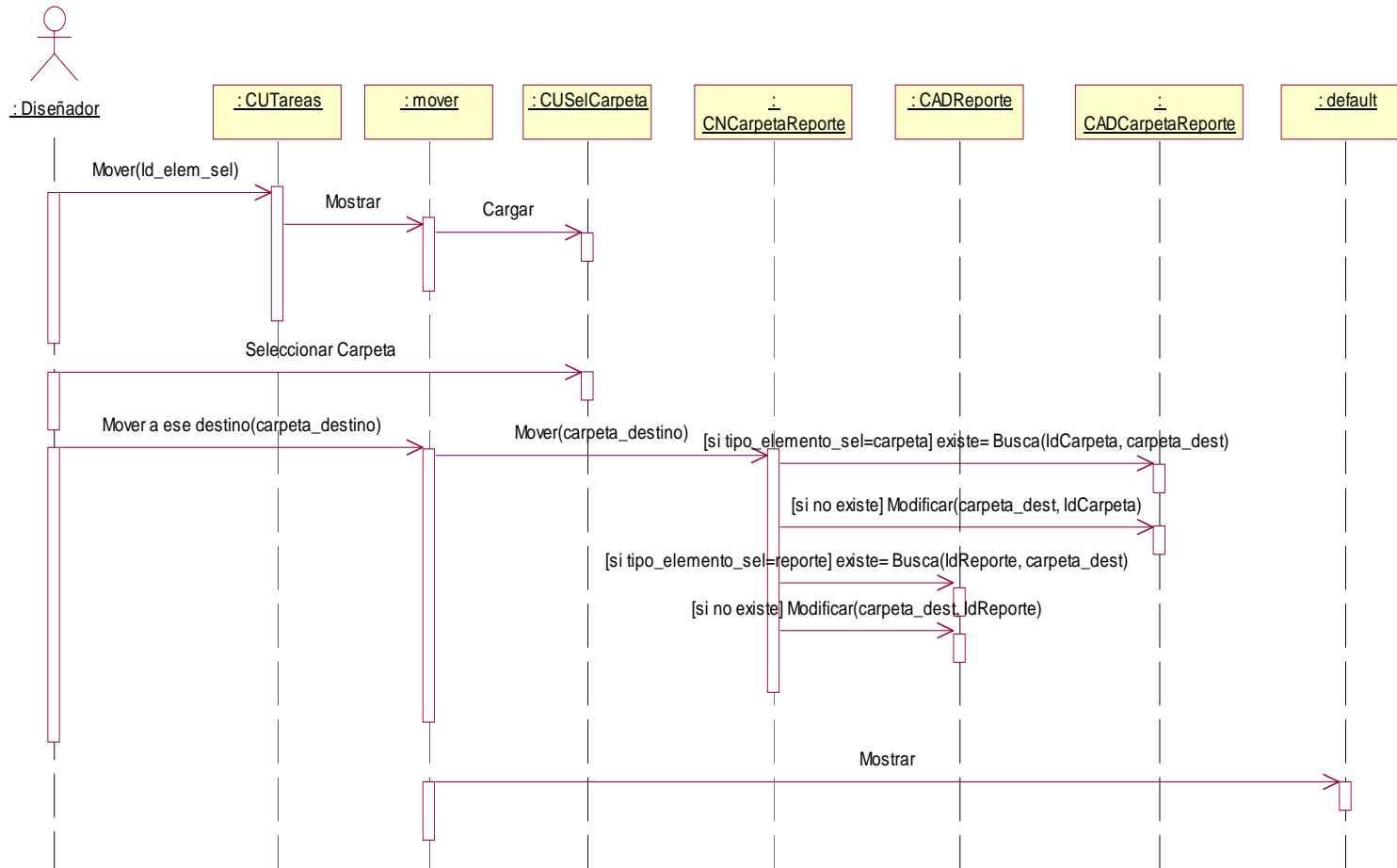
3.2.5.1 Diagrama de Clases: Actualizar Carpetas



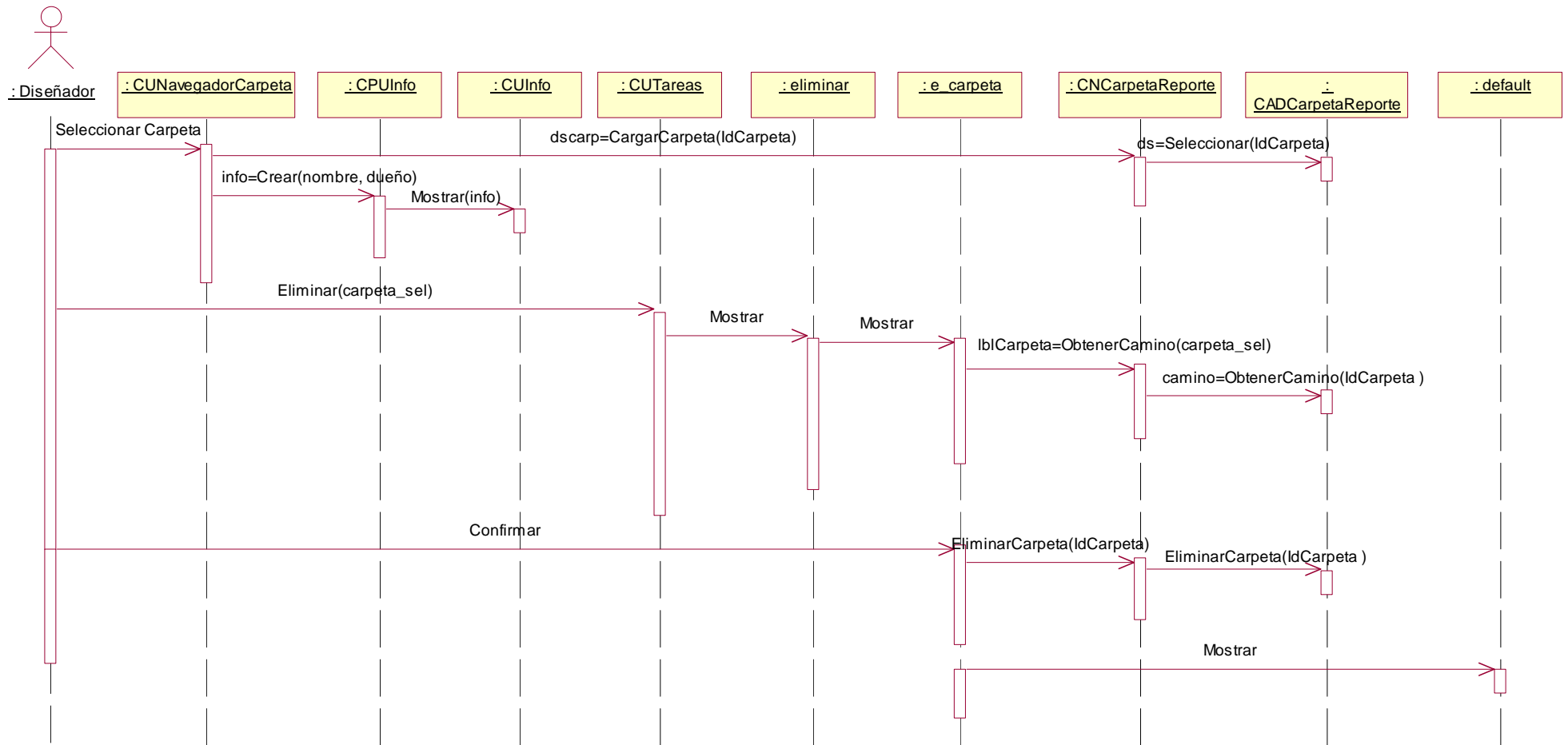
3.2.5.2 Diagrama de Secuencia: Actualizar Carpetas_Crear



3.2.5.3 Diagrama de Secuencia: ActualizarCarpetas_Mover

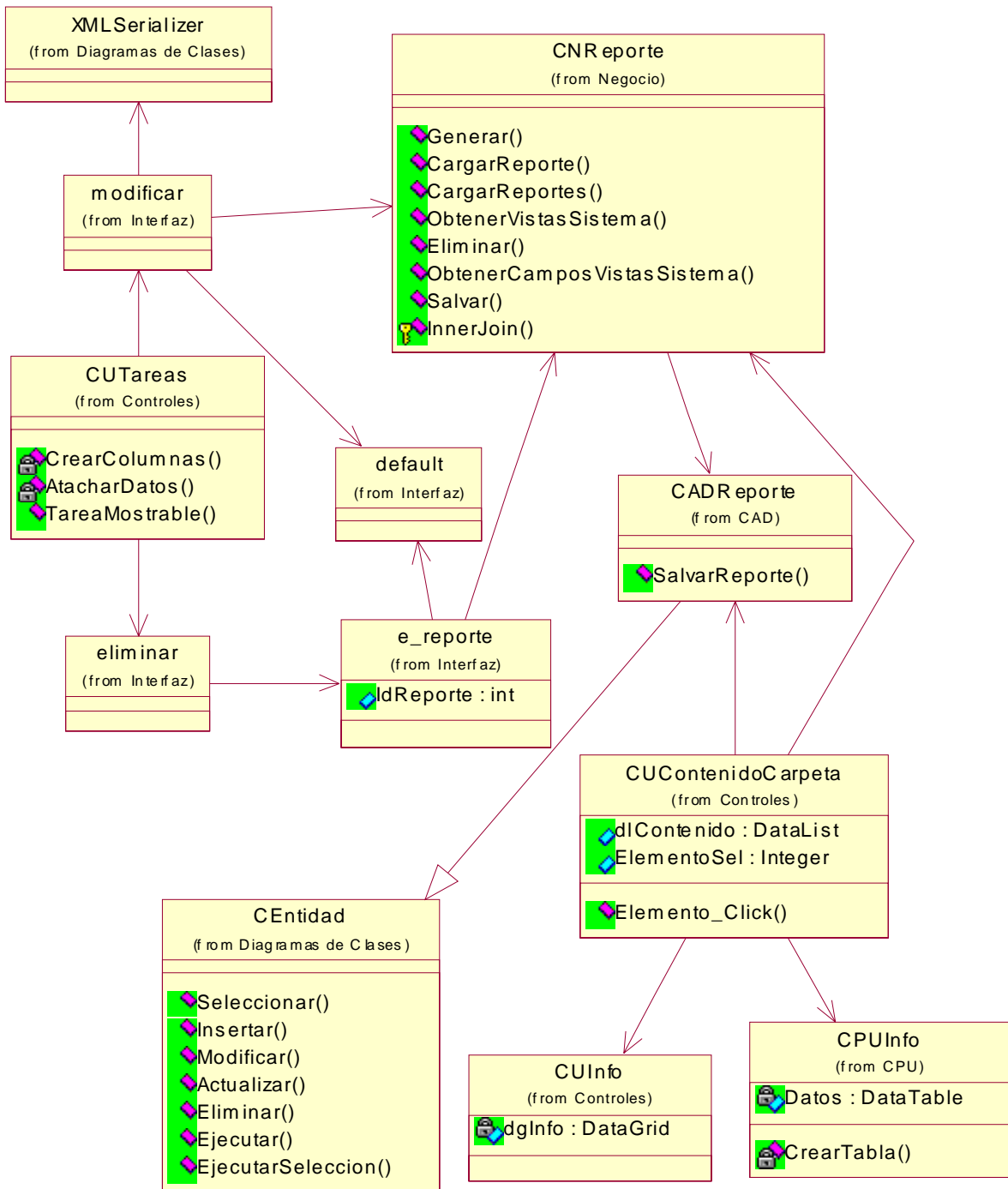


3.2.5.4 Diagrama de Secuencia: ActualizarCarpetas_Eliminar

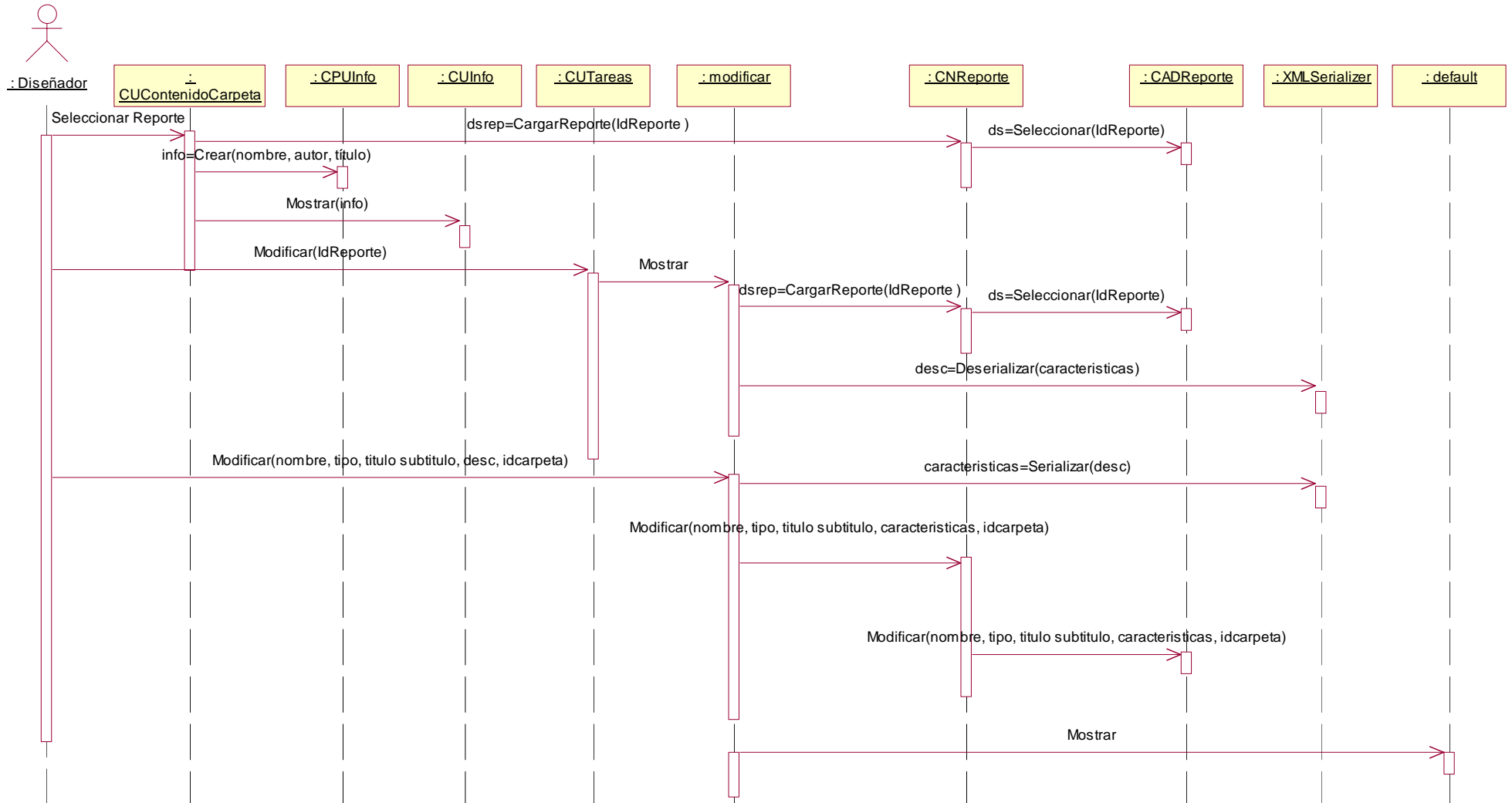


3.2.6 Paquete: Actualizar Reportes

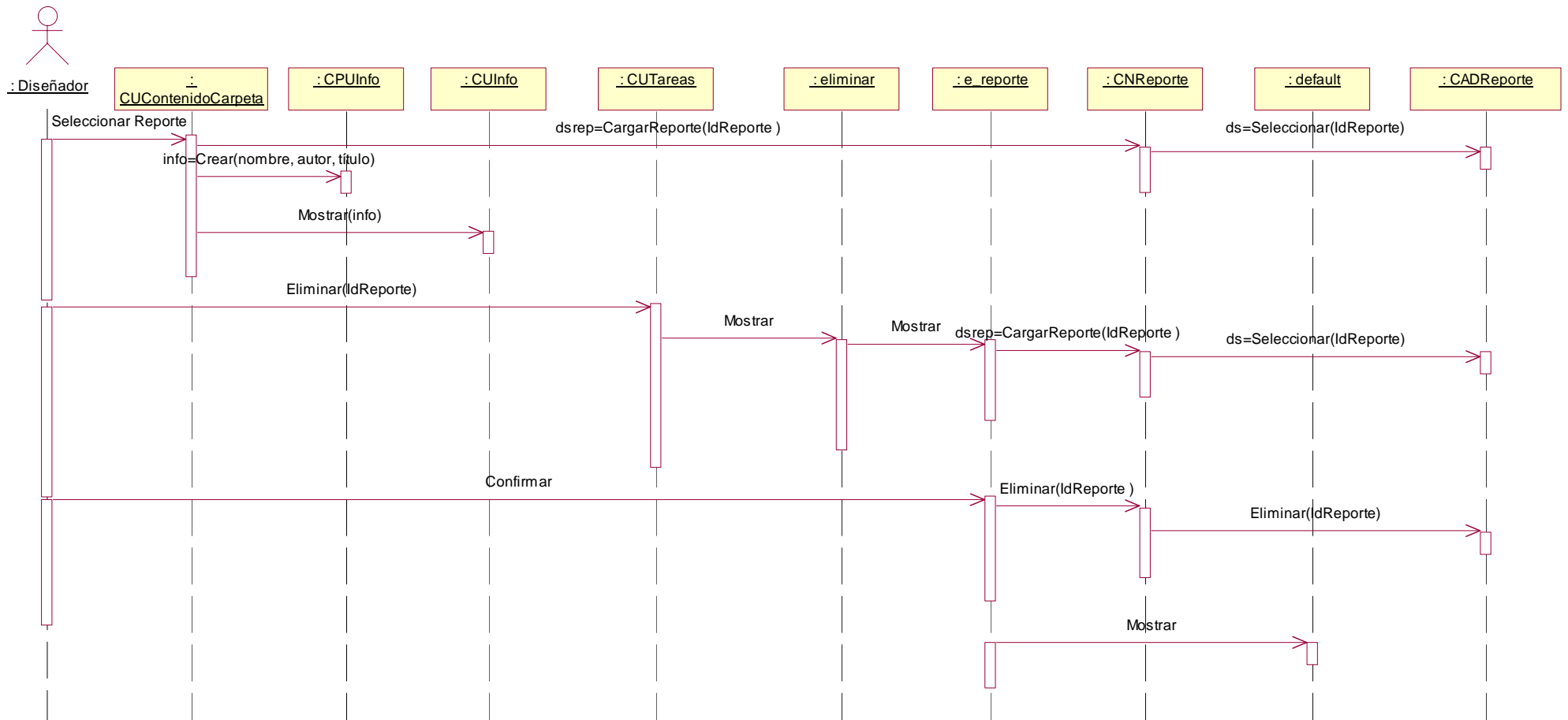
3.2.6.1 Diagrama de Clases: Actualizar Reportes



3.2.6.2 Diagrama de Secuencia: Actualizar Reportes_Modificar

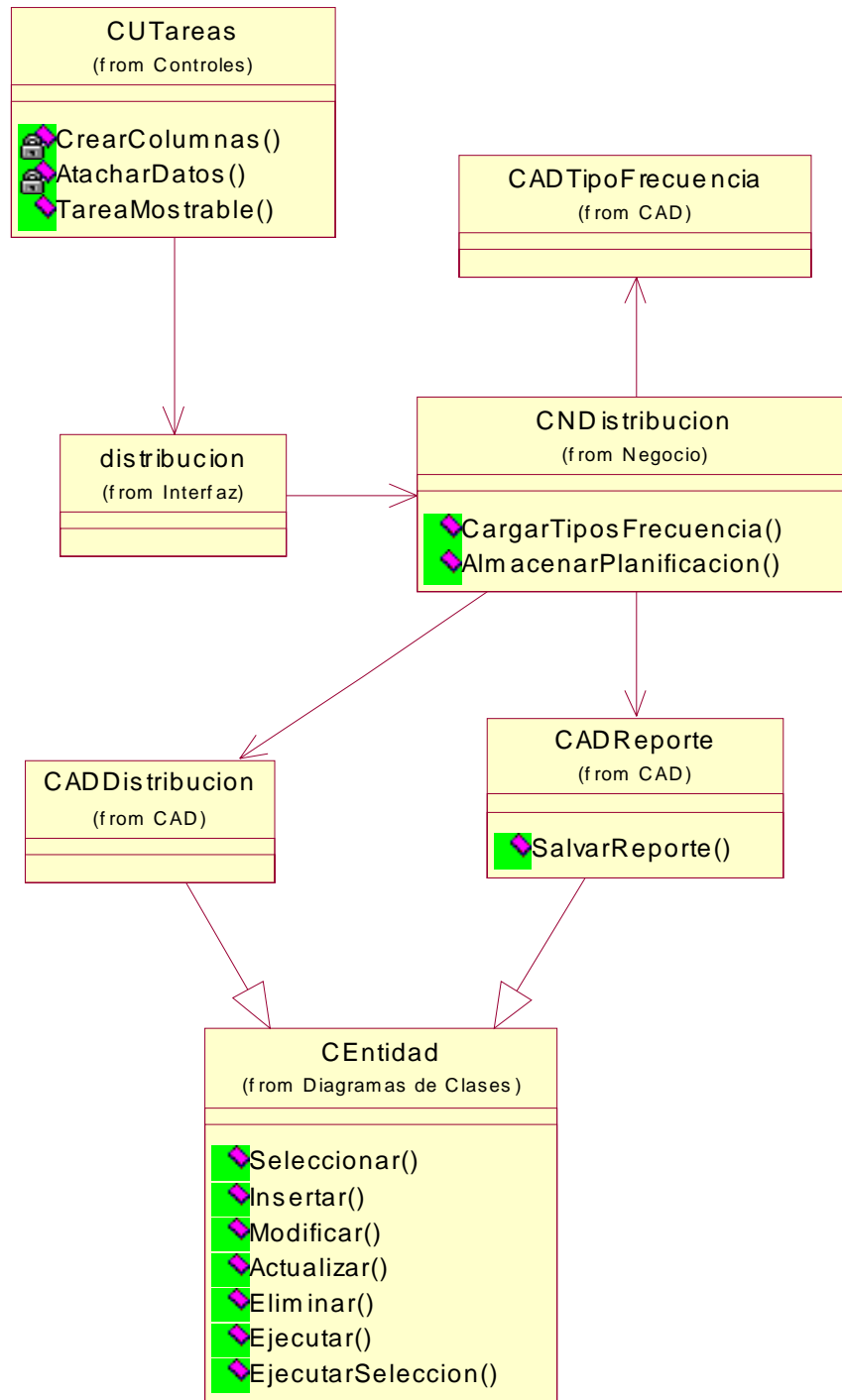


3.2.6.3 Diagrama de Secuencia: ActualizarReportes_Eliminar

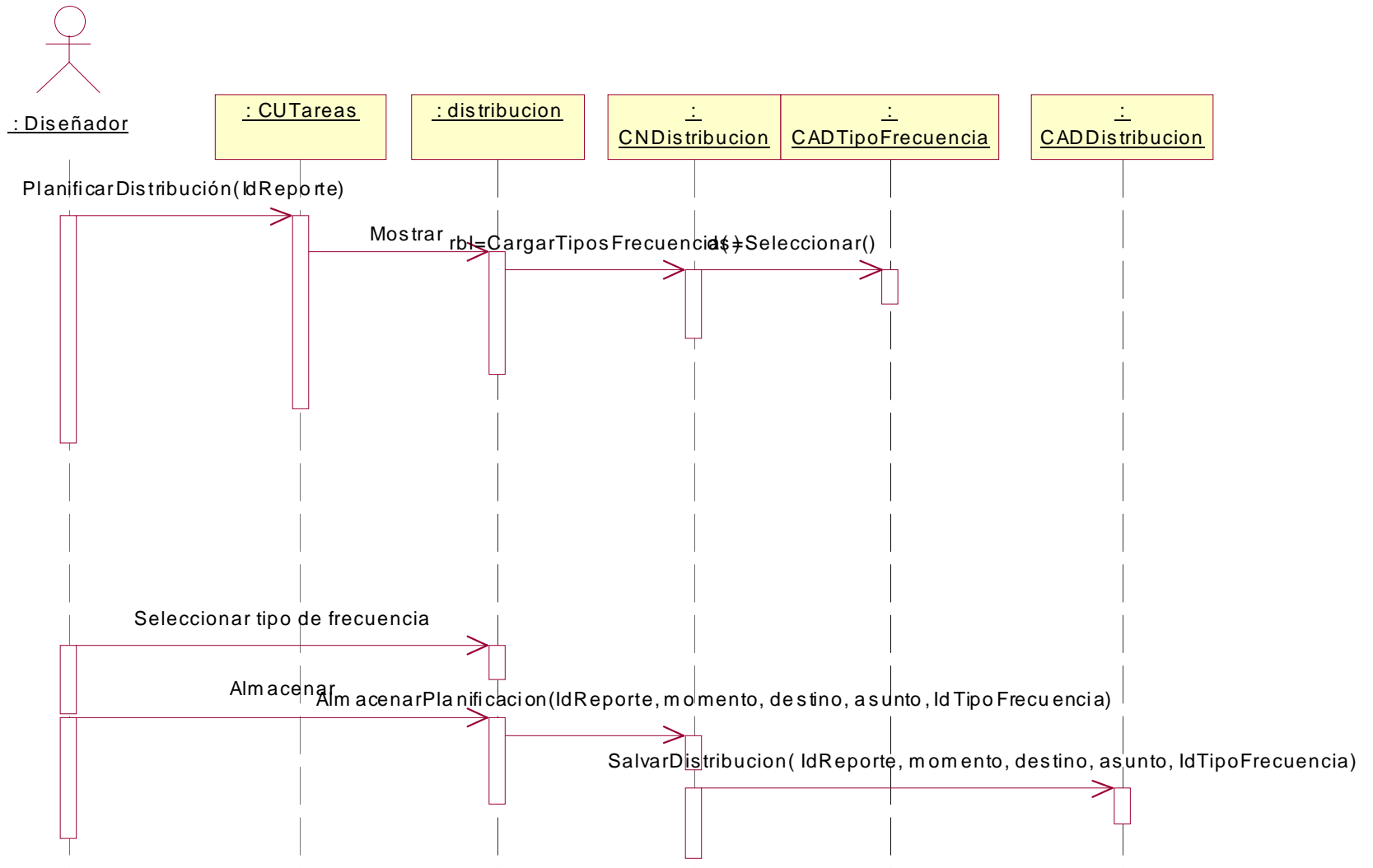


3.2.7 Paquete: Planificar Distribución

3.2.7.1 Diagrama de Clases: Planificar Distribución

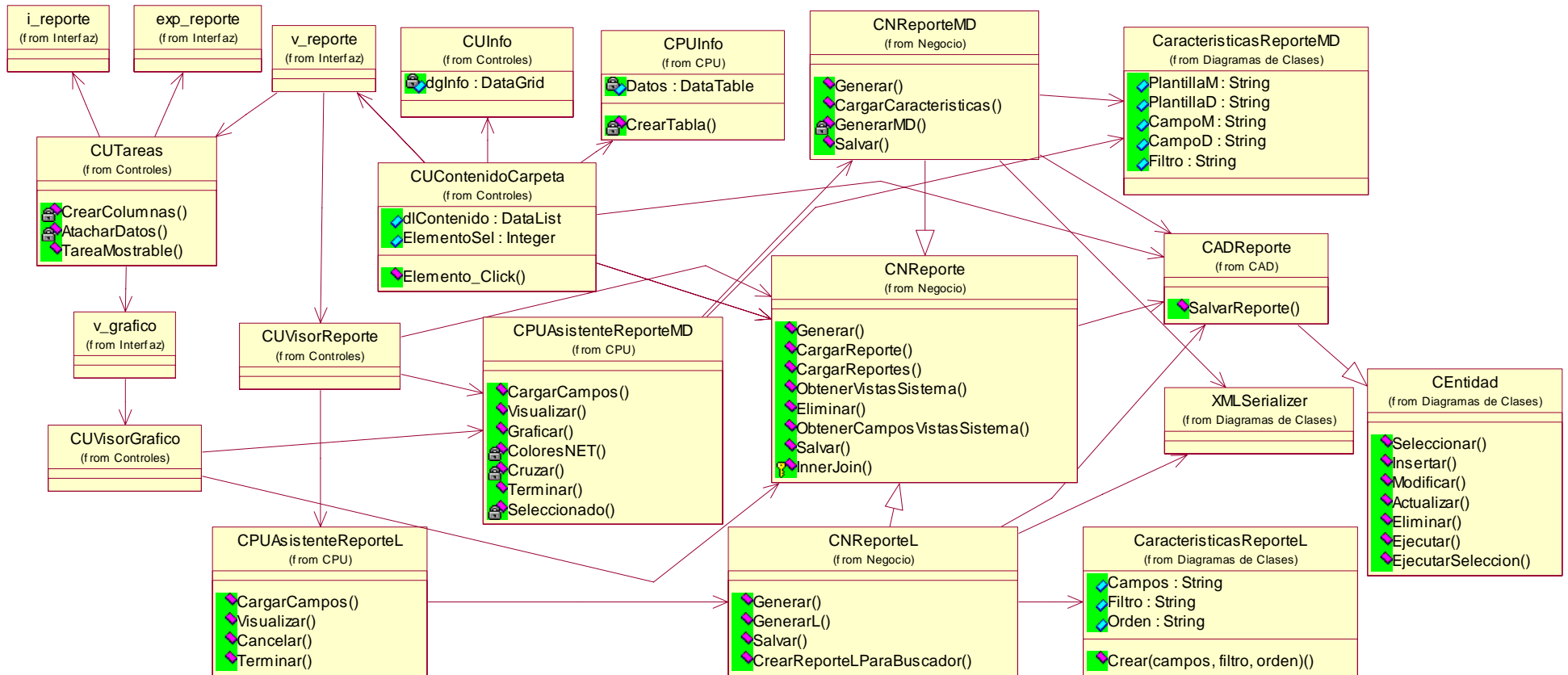


3.2.7.2 Diagrama de Secuencia: Planificar Distribución

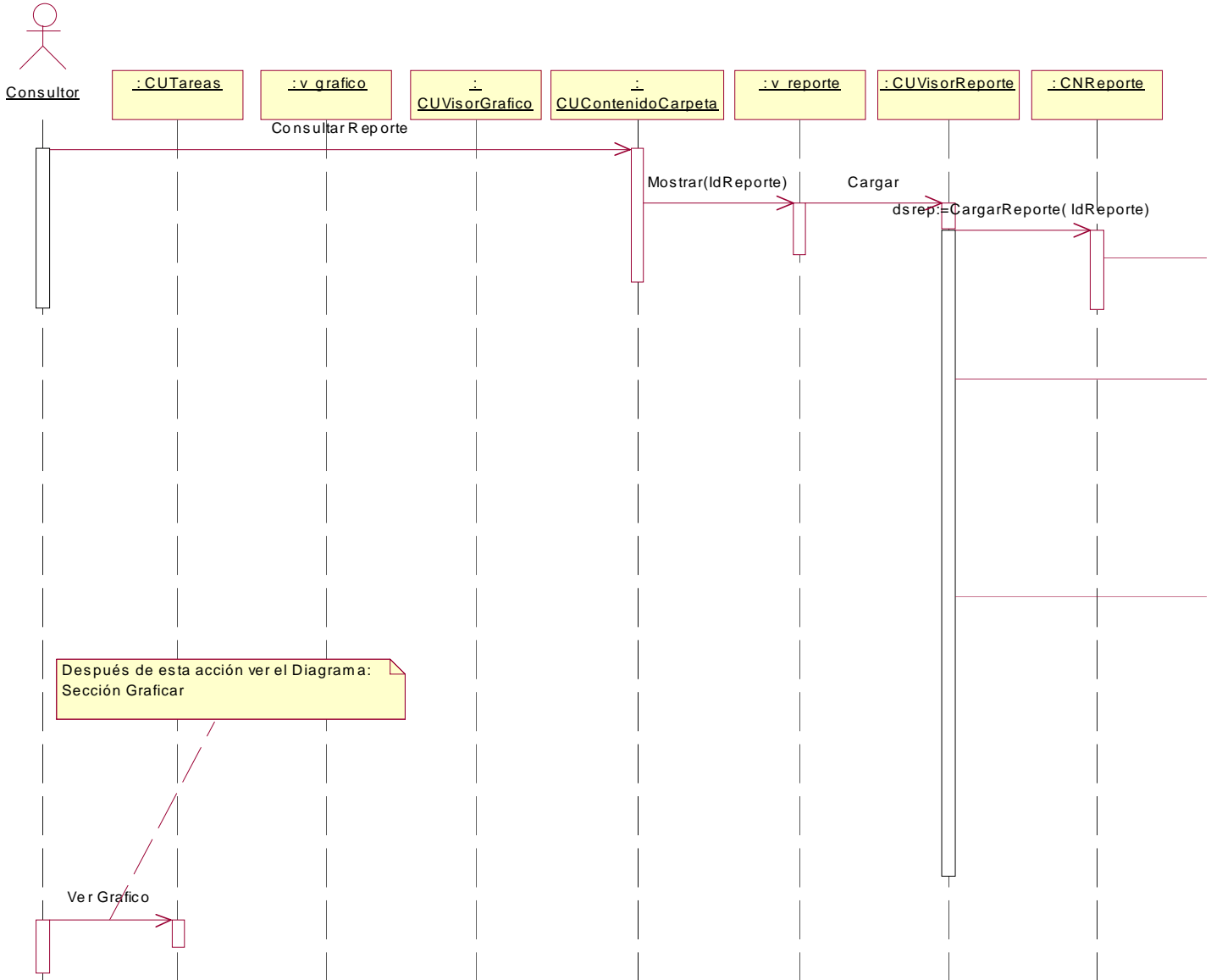


3.2.8 Paquete: Consultar Reporte

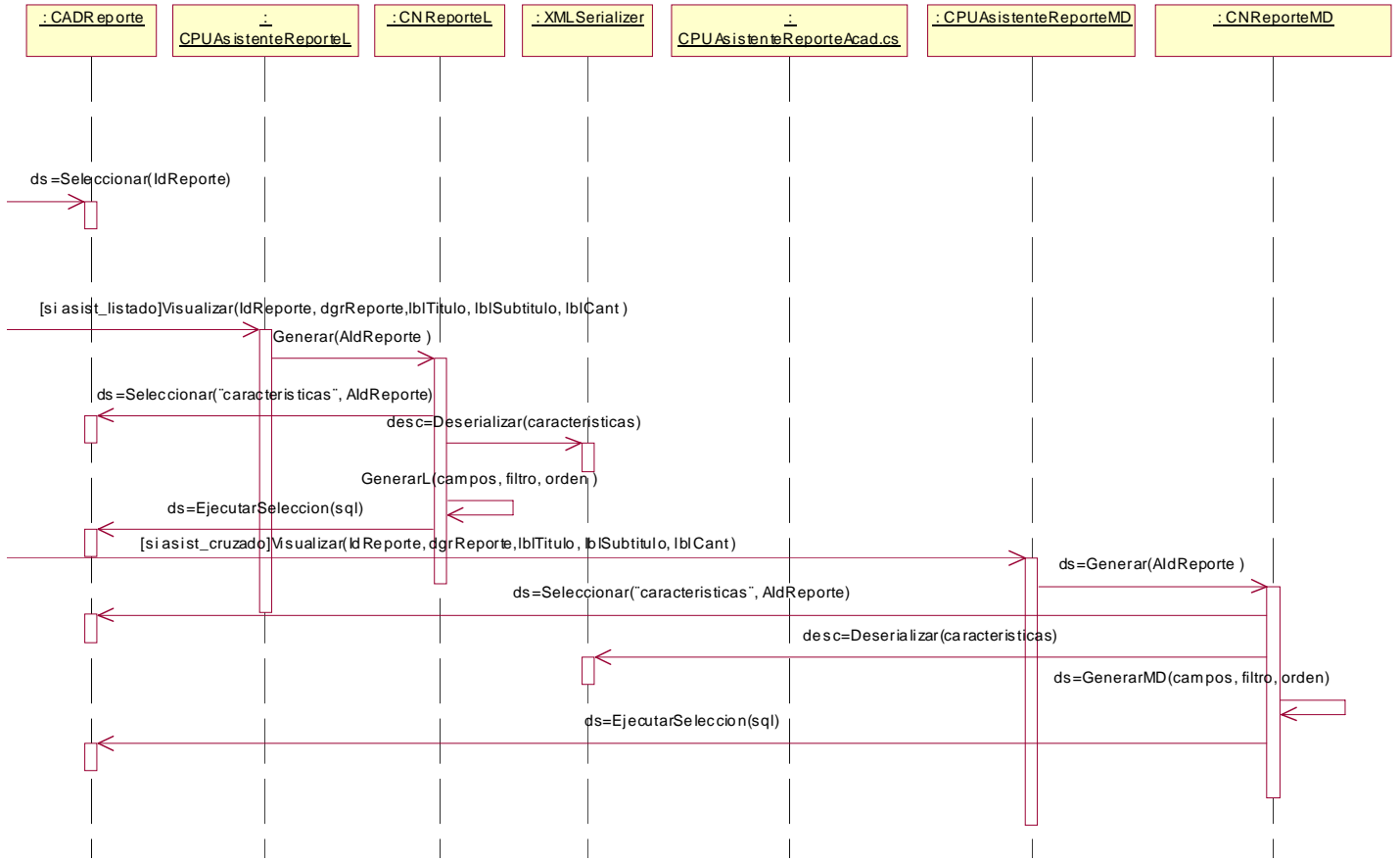
3.2.8.1 Diagrama de Clases: Consultar Reporte



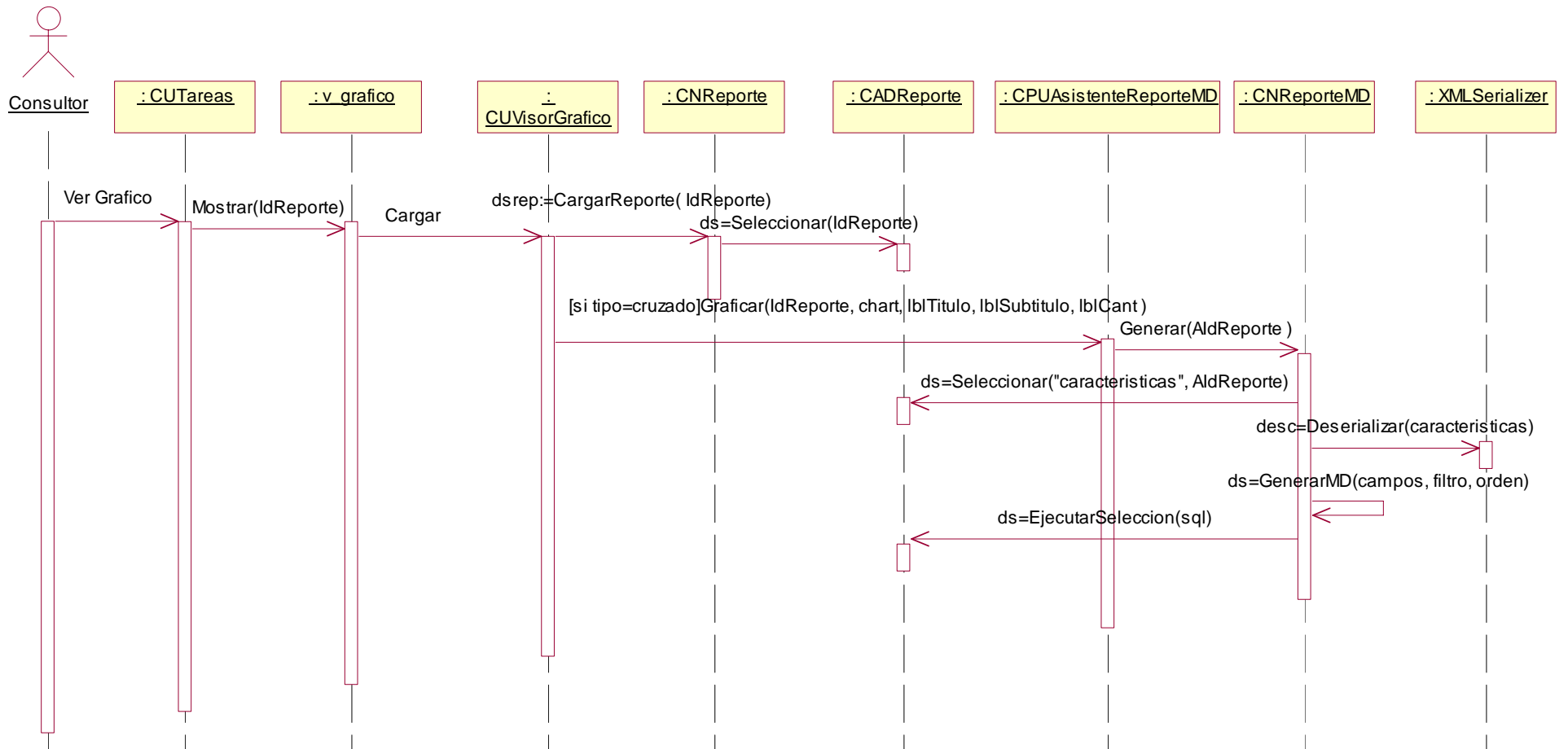
3.2.8.2 Diagrama de Secuencia: Consultar Reporte_Graficar



CAPÍTULO III CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA



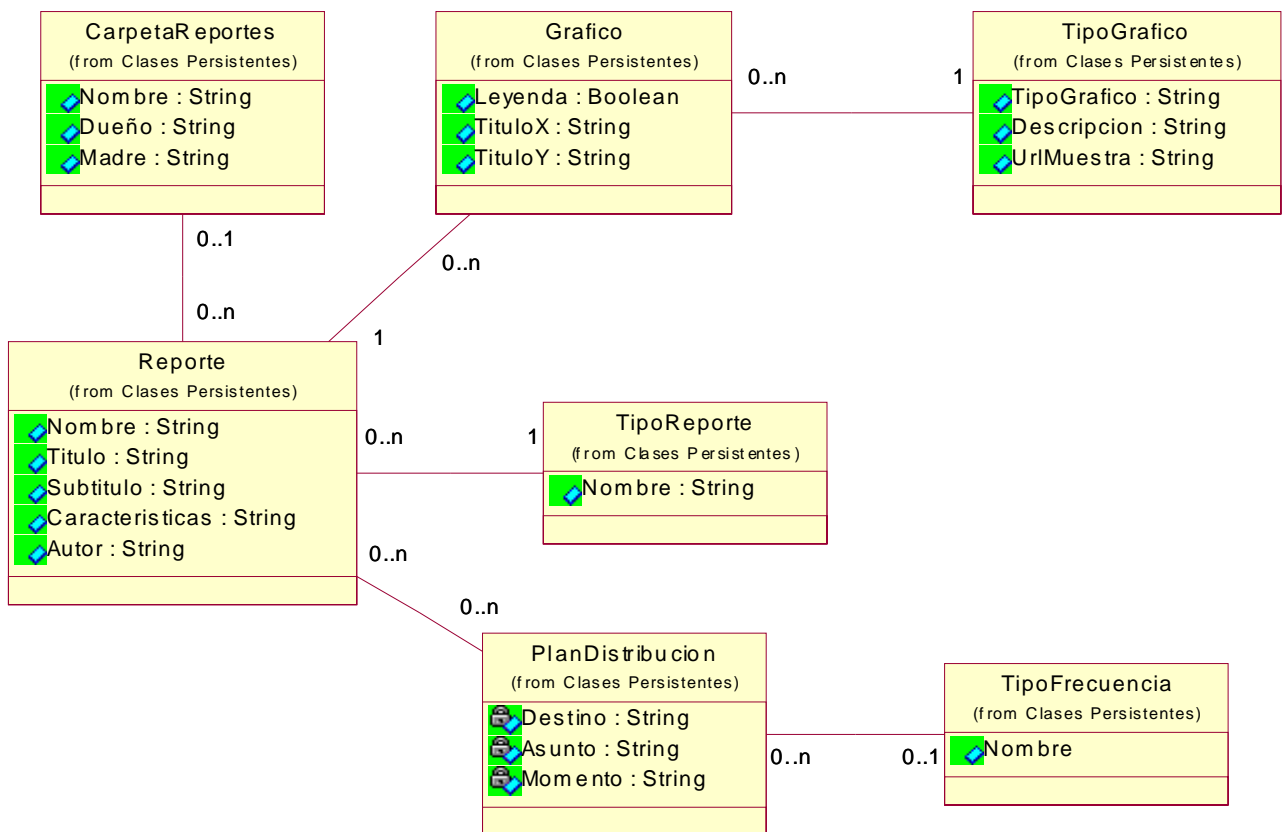
3.2.8.3 Diagrama de Secuencia: Sección Graficar



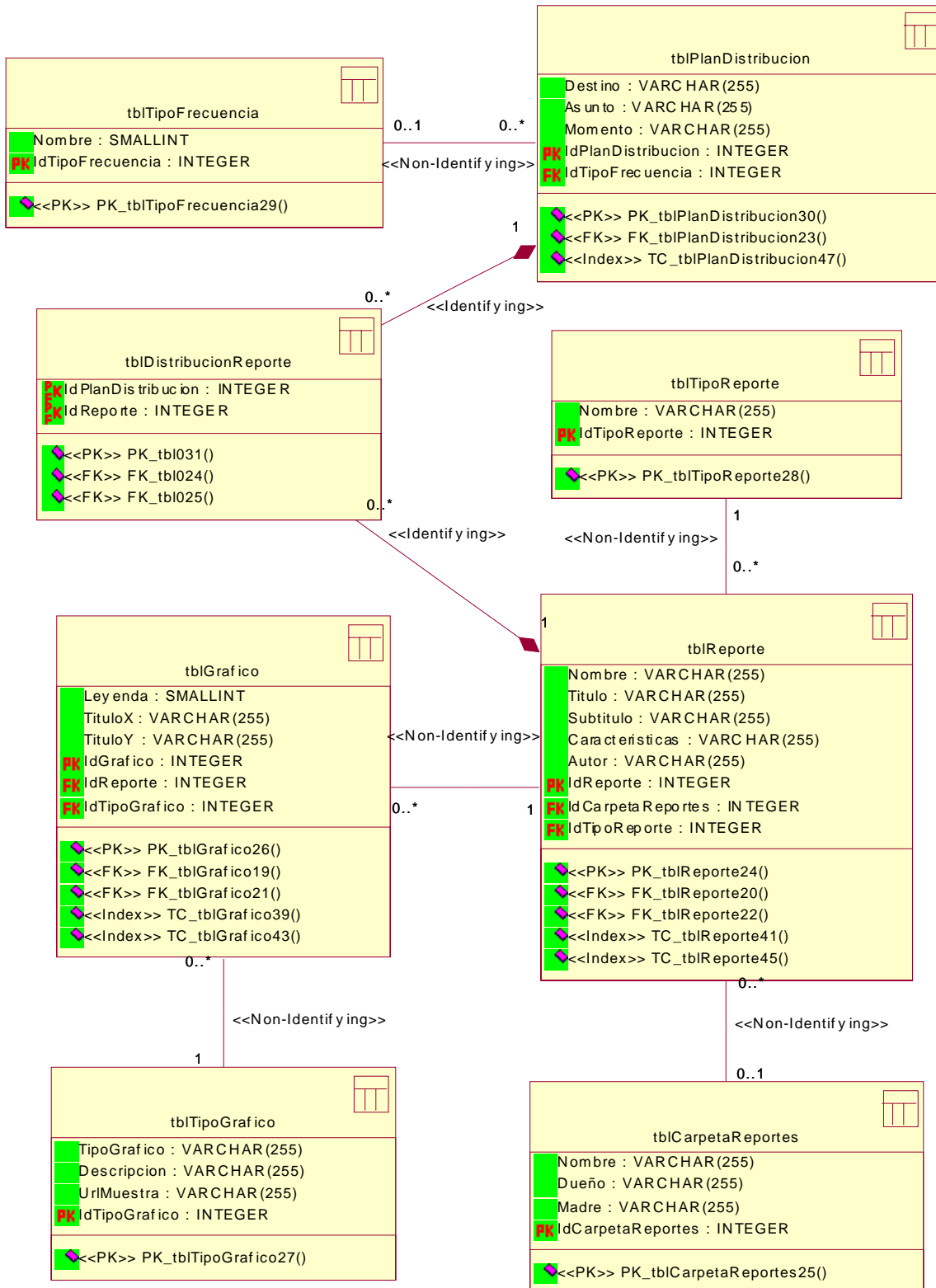
3.3 Diseño de la Base de Datos

3.3.1 Diagrama de Clases Persistentes

La Arquitectura de Aplicaciones de .NET, la cual ha sido empleada para el desarrollo de este sistema, implementa un tipo de componente llamado entidad empresarial, que no es más que los datos que son necesarios pasar de una capa a otra de la arquitectura, esta última plantea que las entidades empresariales que se utilizan de forma interna en la aplicación, suelen ser estructuras de datos, como los DataSets que han sido usados en el sistema que se presenta. Por esta razón no existen clases orientadas a objetos personalizadas, que representan entidades del mundo real, sin embargo, partiendo de que estas clases constituyen otra manera de transportar los datos, también válida para esta arquitectura y con el objetivo de llegar al Modelo de Datos se elabora este diagrama.



3.3.2 Modelo de Datos



3.4 Principios de Diseño

3.4.1 Estándares en la interfaz de la aplicación

La interfaz del sistema favorece su usabilidad, ya que la información que se exhibe en ella, es fácil de ubicar y leer, pues todas las tareas que puede efectuar el usuario aparecen juntas en la parte izquierda de la pantalla. Además el contraste de colores entre las letras y el fondo, en este caso verde claro y negro, es adecuado para la lectura y para refrescar la vista.

También esta interfaz cumple con el principio de la anticipación, pues antes de que el usuario se interese y esfuerce en buscar la información general referente a un reporte o carpeta, como es: el nombre, el autor y el título, el sistema se lo muestra al seleccionar uno de estos elementos. De la misma manera que ofrece la descripción de los campos de las plantillas cuando se marcan.

Además la navegación de la aplicación es bastante reducida. El usuario siente que se mantiene en un único lugar y que el que va variando es su trabajo. Esto no solamente elimina la necesidad de mantener mapas u otras ayudas de navegación, sino que además brindan al usuario una sensación de autonomía, lo cual se pone de manifiesto en los asistentes que tiene la aplicación y que son usados por el usuario para crear reportes o gráficos.

Por otra parte se ha tratado que exista consistencia entre la interfaz y la plataforma que más utilizan los usuarios del sistema: Windows. Por esta razón se ha apostado por una interfaz del tipo “explorador”, muy recurrida en esta plataforma y con probada utilidad para la búsqueda rápida de información, que puede organizarse jerárquicamente, en este caso, los reportes. La decisión de utilizar este estándar de interfaz se basa en que muchas personas querrán y podrán crear reportes y llegará el momento en que será una cantidad muy numerosa con tendencia a incrementar, por lo que es conveniente distribuirlos en carpetas.

3.4.2 Concepción general de la ayuda

La ayuda de este sistema se basa en ventanas que aparecen cuando el usuario la solicita en un momento determinado. Esto es uno de los principios de diseño que existe y que hay que tener en cuenta, porque de lo contrario, el sistema de ayuda tapanía el área de trabajo y por tanto se estaría forzando al usuario a memorizar el contexto en orden para interpretar el mensaje de ayuda, y una vez comprendido este, también tendría que recordarlo mientras regresa al lugar donde fue invocado, es válido aclarar que esta llamada se realiza siempre de la misma manera y al salir de la ayuda todo queda en el mismo estado en que se encontraba. Por otra parte, el sistema ayuda al usuario activando la etiqueta que describe los campos de las plantillas cuando son seleccionados, lo que facilita y agiliza el diseño de los reportes.

3.4.3 Tratamiento de excepciones

Los mensajes que aparecen a raíz de una excepción son ventanas al igual que el sistema de ayuda y generalmente brindan información acerca de la posible causa del error, además de señalar alguna solución al mismo.

El sistema cumple con la recuperación de errores que es tan importante para que el usuario pierda el miedo a fracasar en el uso de la aplicación, haciéndola, por tanto, fácil de aprender. Para esto, usa dos de los tipos más comunes de este principio: el cancelar y el deshacer. En cada paso de los asistentes el diseñador puede cancelar las acciones que ha realizado. El deshacer lo emplea de la manera más sencilla, lanzando preguntas de confirmación ante la solicitud de borrar un reporte o carpeta.

3.5 Estándares de Codificación

Los estándares de codificación que se siguen coinciden con el estilo de codificación que debe aplicarse a la hora de programar en C#.

1. Identación

Para la indentación no se utilizaron espacios en blanco, sino Tabs, los cuales proporcionan algunas ventajas:

- ❖ Es sólo un carácter y no 2, 4, 8 como en el caso de los espacios en blanco.
- ❖ Si se quiere incrementar o decrementar la sangría, basta con marcar un bloque y presionar Tab para incrementar y Shift-Tab para decrementar.

2. Comentarios

- Bloques de Comentarios

Para las descripciones se usaron tres slashes.

- Comentarios de una sola línea

Se usaron dos slashes para comentar una línea o una sección de código. Además la longitud del comentario no excede el tamaño del código explicado por él.

- Comentarios de Documentación

En la aplicación se le da gran uso a la generación de un sistema de documentación basado en comentarios XML que introdujo Microsoft y que siguen el siguiente patrón:

```
///
///
```

3. Declaraciones

- Se usa una sola declaración por línea, aunque las variables sean del mismo tipo.

```
int IdReporte
int IdPlantilla
```

- Se trató de inicializar las variables en el mismo momento en que se declaraban.
- En la declaración de las clases se usaron las siguientes reglas:
 - ❖ No espacio entre el nombre de un método y el paréntesis abierto para comenzar la lista de parámetros.

- ❖ La llave abierta aparece en la próxima línea a la declaración.
- ❖ La llave cerrada comienza una línea, donde sólo parecerá ella, y coincide con la alineación de la llave abierta.

4. Instrucciones

- Cada línea contiene una sola instrucción
- Se sigue el siguiente patrón para las instrucciones if, if-else, if else-if else:

```
If (condición) {  
    Hacer algo;  
}
```

```
If (condición) {  
    Hacer algo;  
} else{  
    hacer otra cosa;  
}
```

```
If (condición) {  
    Hacer algo;  
} else If (condición) {  
    hacer otra cosa;  
} else  
    hacer otra cosa de nuevo;  
}
```

- Se sigue el siguiente patrón para las instrucciones for/ foreach:

```
For (int i = 0; i<5; ++i) {  
    ....  
}
```

o simplemente

```
for (inicialización; condición; modificación);
```

```
for (int i in IntList) {  
    ...  
}
```

- Las instrucciones switch tienen la siguiente forma:

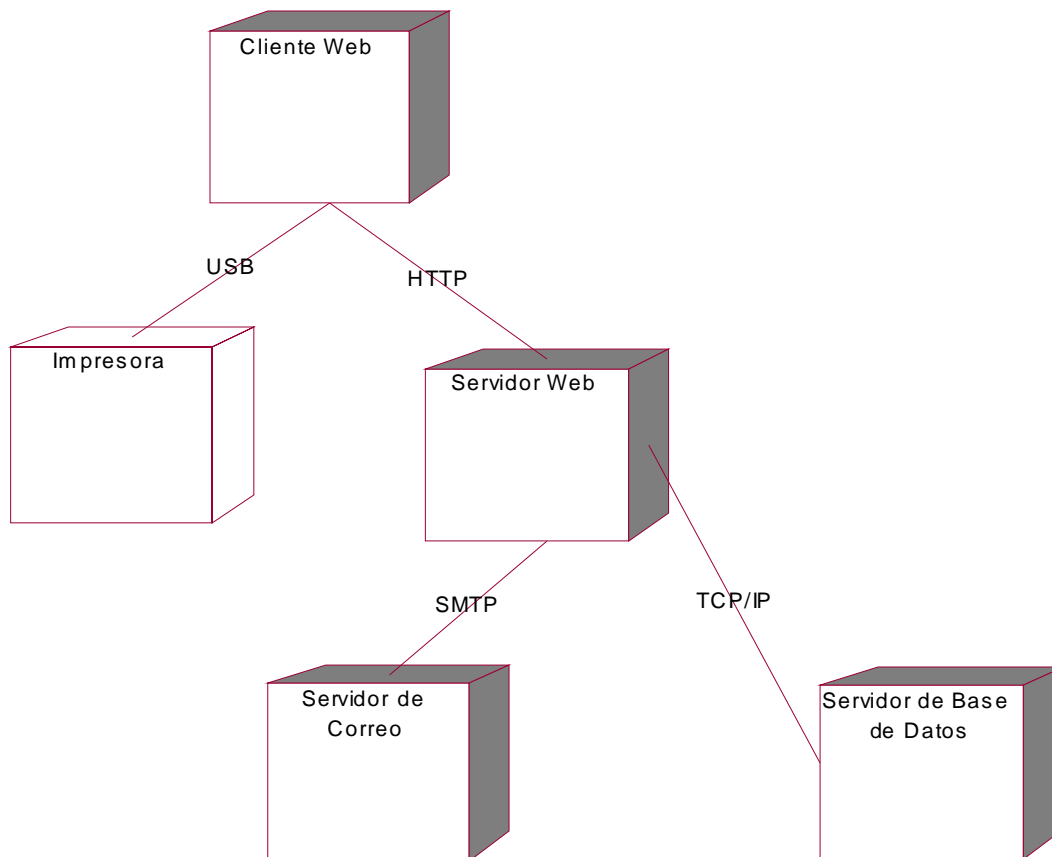
```
Switch (condición) {  
    case A:  
        ...
```

```
        break;
    case B:
        ...
        break;
    case C:
        ...
        break;
}
```

5. Nombramiento de clases: según la notación Pascal, es decir se coloca en mayúscula la primera letra de cada palabra, además el nombre es un sustantivo.
6. Nombramiento de los métodos: son verbos o frases verbales, usando la notación de Pascal.

3.6 Modelo de Despliegue

3.6.1 Diagrama de Despliegue



Estructura de Nodos

En el Diagrama de Despliegue representado anteriormente aparece el nodo Cliente Web que representa a todos los ordenadores de los usuarios que accederán al sistema para hacer uso de él, cada uno de estos nodos se comunica, a través del protocolo http, con el Servidor Web donde está publicada la aplicación, este a su vez está conectado al Servidor de Base de Datos y por medio del protocolo TCP/IP se comunica con él, con el objetivo de obtener y actualizar los datos del sistema. En el diagrama también aparece el nodo Servidor de Correo, el cual es utilizado por el Servidor Web para distribuir los reportes por correo electrónico, el protocolo que media entre ellos es el SMTP. Además se representa un dispositivo conectado al Cliente Web, la impresora, que no tiene que existir en todos los casos, sólo cuando se desee imprimir un determinado reporte.

3.6.2 Arquitectura de Capas

La arquitectura de capas que se presenta se corresponde con la Arquitectura de Aplicaciones de .NET, la cual establece la existencia de cuatro capas, a través de las cuales se distribuye toda la aplicación.

Capa de Presentación: esta capa contiene los Componentes de Interfaz de Usuario, los cuales interactúan directamente con el mismo y los Componentes de Proceso de Usuario que se encargan de controlar dichas interacciones; ambos se ejecutan en el Servidor Web.

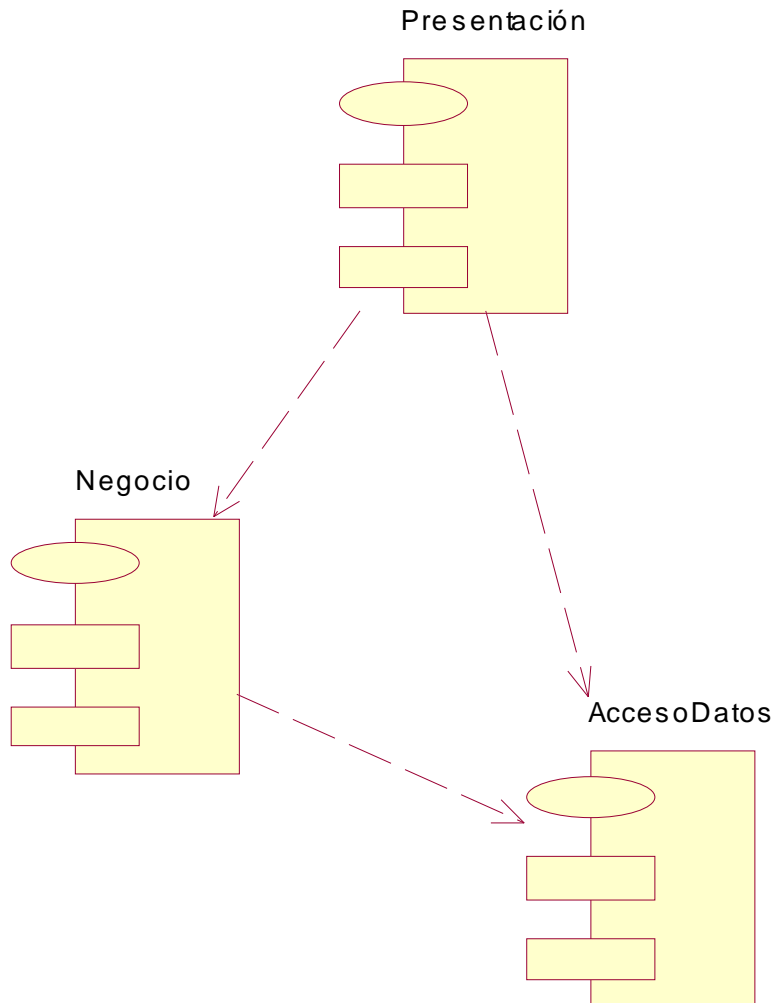
Capa Empresarial: en ella se encuentran ubicados los componentes que se encargan de gestionar toda la lógica del negocio de la aplicación. En este caso se usaron los Componentes y las Entidades Empresariales, también se ejecutan en el Servidor Web.

Capa de Acceso a Datos: a esta capa pertenecen los componentes que acceden a los almacenes de datos, por tanto ellos son los que se encargan de consultar, insertar, modificar y eliminar la información propia del sistema. Para esta aplicación se emplearon los Componentes Lógicos de Acceso a Datos, los cuales se encuentran en el Servidor Web.

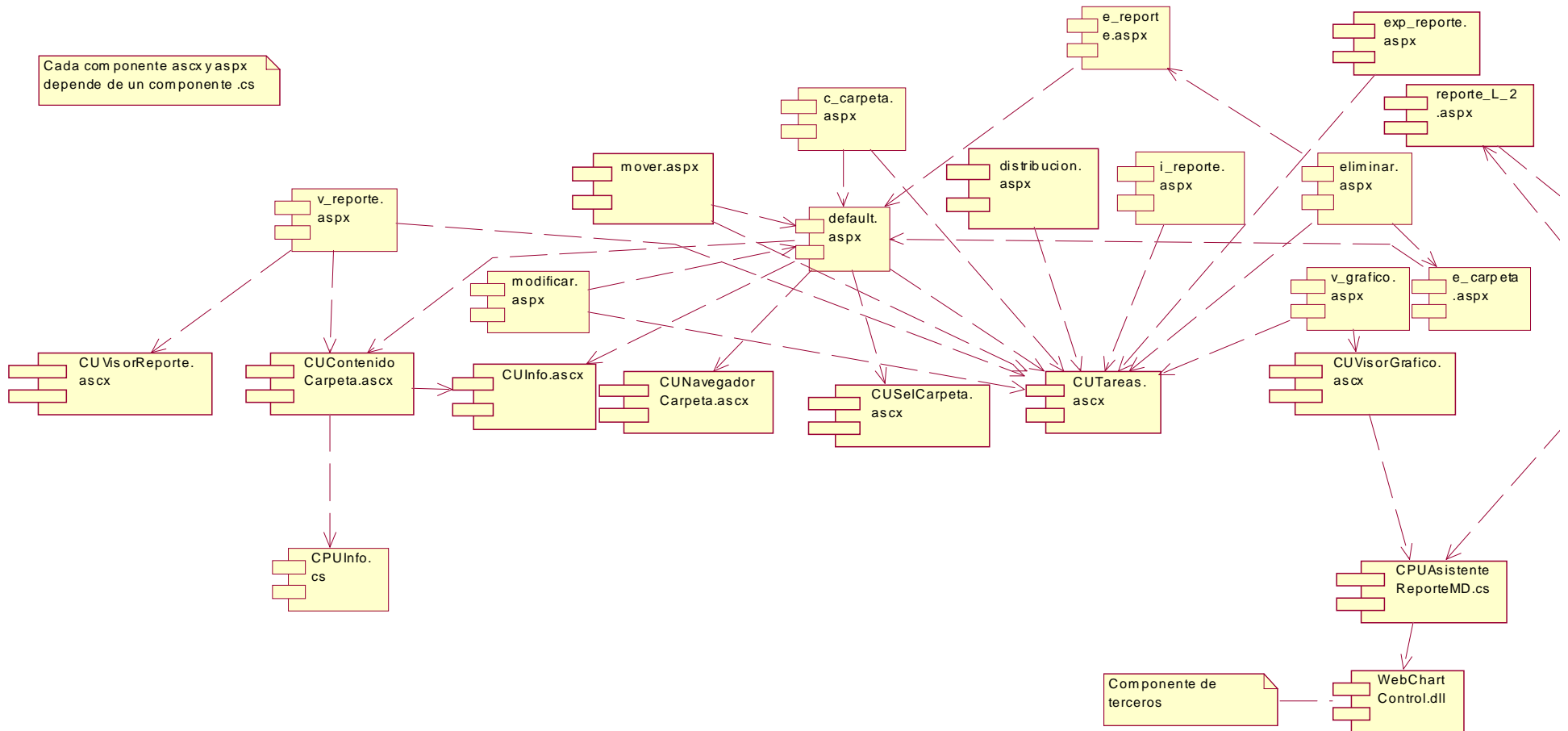
Capa de Datos: esta capa es la que contiene los datos del sistema como tal, en este caso las tablas pertenecientes a la Base de Datos Akademos, con toda la información que almacena cada una de ellas, lógicamente se encuentra en el Servidor de Base de Datos.

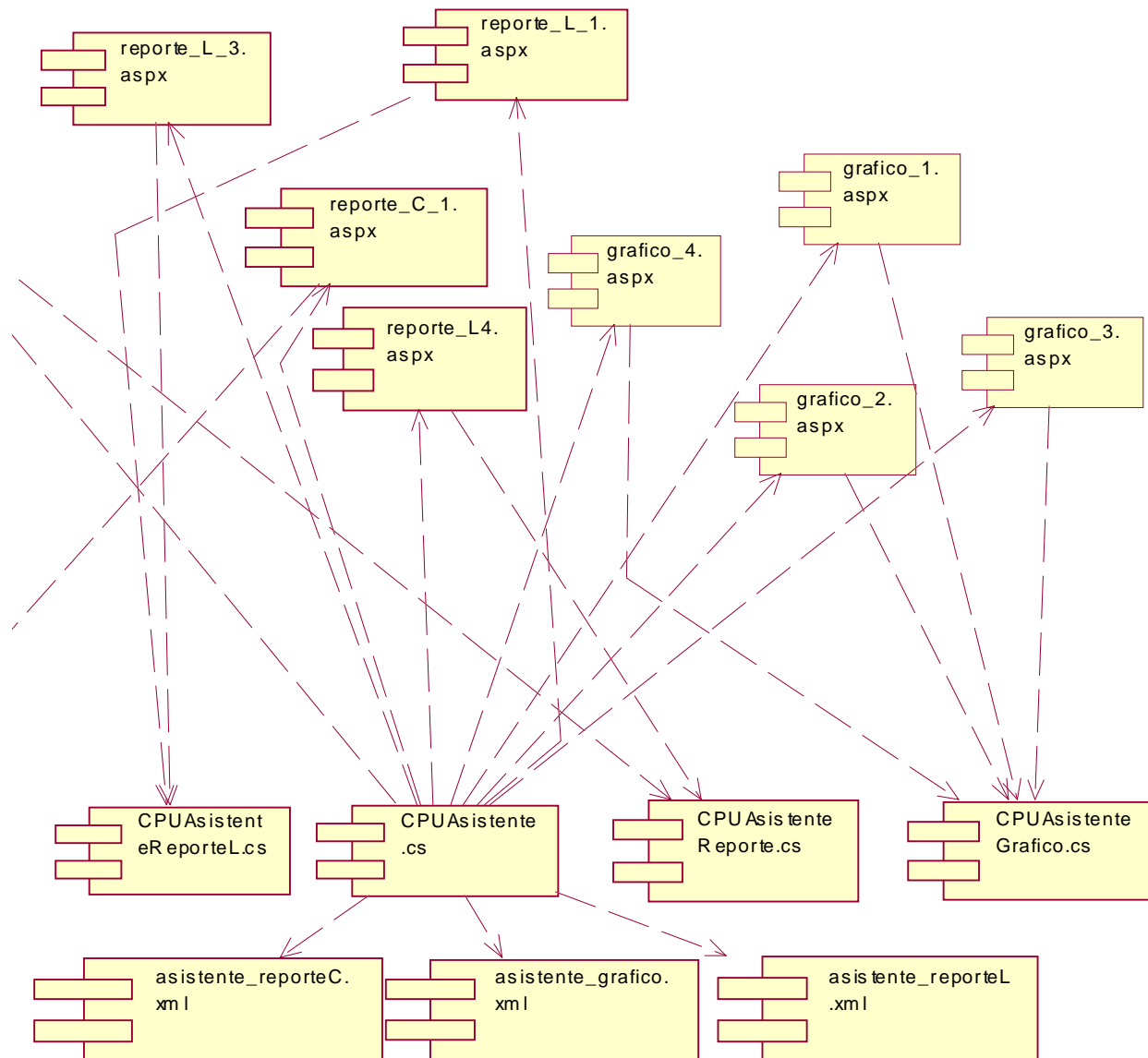
3.7 Modelo de Implementación

3.7.1 Diagrama de Paquetes de Componentes

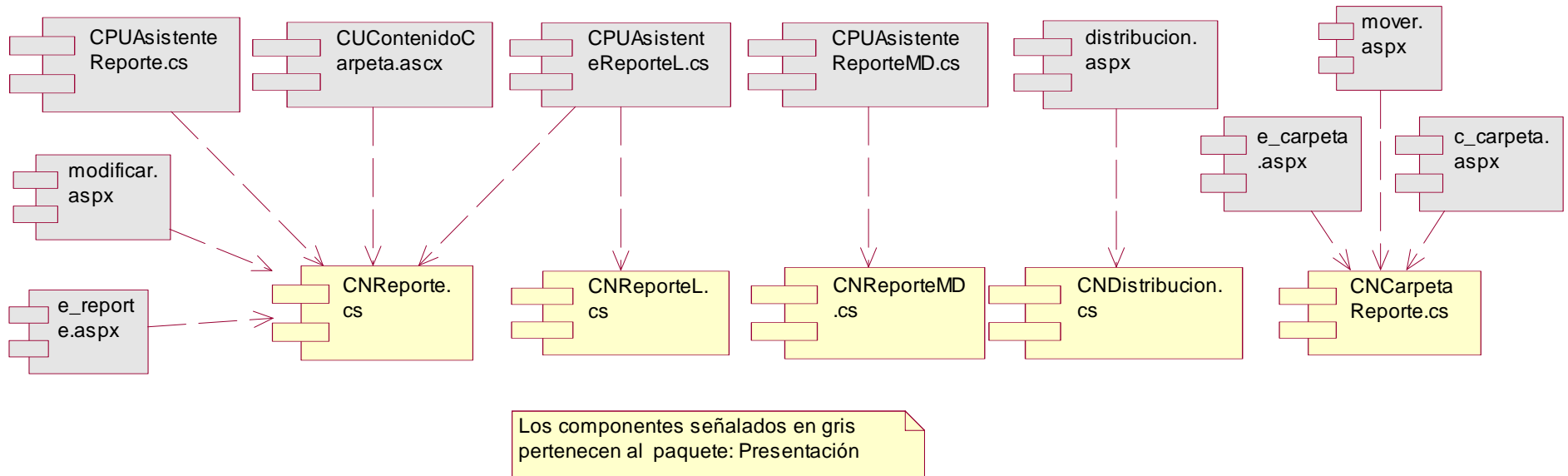


3.7.1.1 Diagrama de Componentes del Paquete Presentación

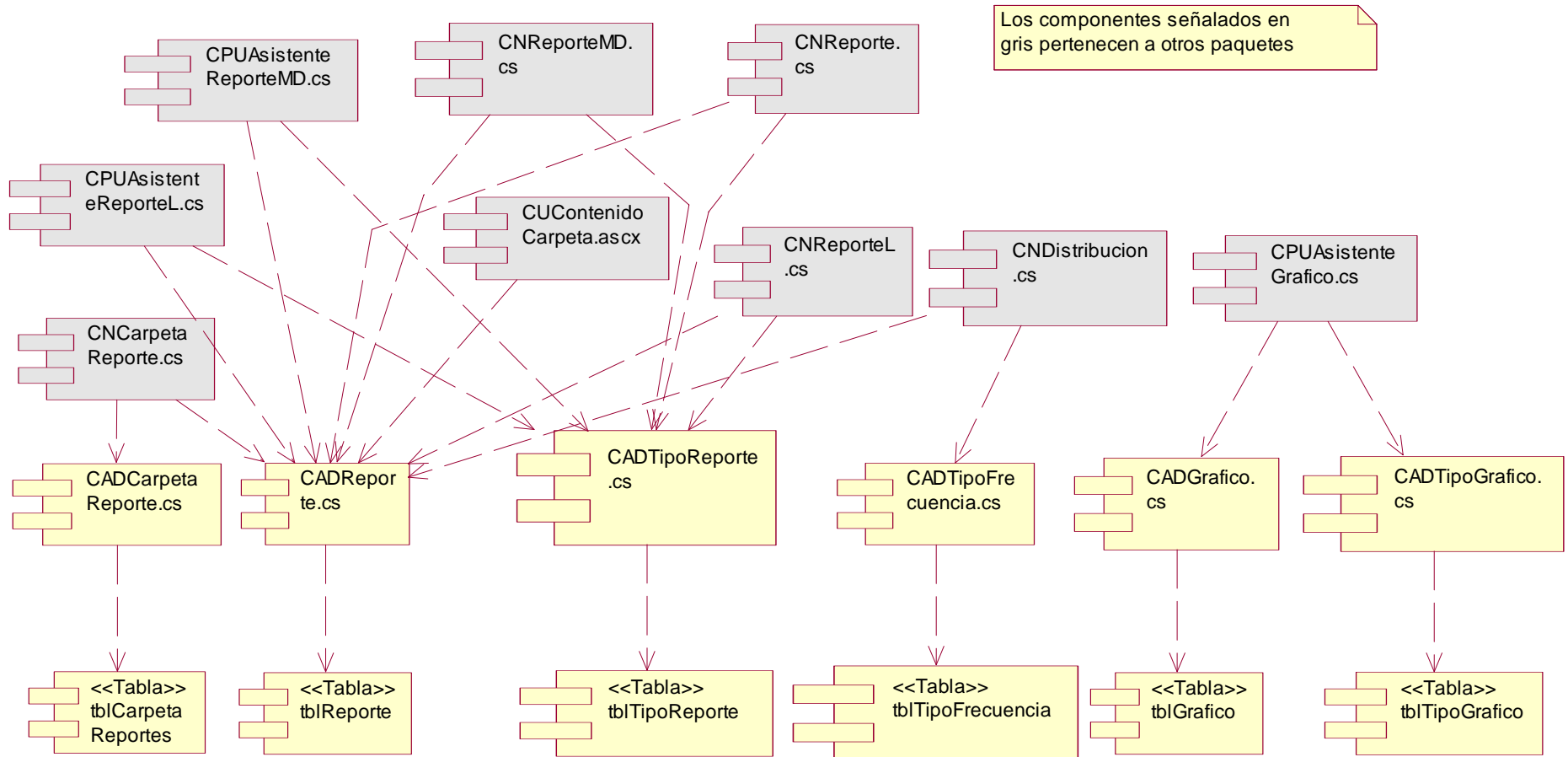




3.7.1.2 Diagrama de Componentes del Paquete Negocio



3.7.1.3 Diagrama de Componentes del Paquete Acceso a Datos



3.7.2 Descripción de los componentes

Componente	Propósito	Contenido
default.aspx	Mostrar el explorador que contiene los reportes distribuidos por carpetas y las tareas que puede realizar el usuario.	
default.aspx.cs	Controlar el componente default.aspx	
reporte_C_1.aspx	Mostrar las plantillas y los campos de cada una. Adquirir y validar los seleccionados por el usuario para las filas y las columnas de su reporte.	
reporte_C_1.aspx.cs	Controlar el componente reporte_C_1.aspx	
reporte_L_1.aspx	Mostrar las plantillas y los campos de cada una. Adquirir y validar los seleccionados por el usuario como información de su reporte	
reporte_L_1.aspx.cs	Controlar el componente reporte_L_1.aspx	
reporte_L_2.aspx	Mostrar las partes que puede tener el filtro de un reporte. Adquirir y validar las seleccionadas por el usuario para conformar el mismo	
reporte_L_2.aspx.cs	Controlar el componente reporte_L_2.aspx	
reporte_L_3.aspx	Mostrar los campos del reporte y adquirir el orden de los mismos que ha sido indicado por el usuario	
reporte_L_3.aspx.cs	Controlar el componente reporte_L_3.aspx	
reporte_L_4.aspx	Adquirir el nombre, título y subtítulo que da el usuario a los reportes diseñados.	
reporte_L_4.aspx.cs	Controlar el componente reporte_L_4.aspx	
grafico_1.aspx	Mostrar los tipos de gráficos y adquirir el que el usuario desea construir	
grafico_1.aspx.cs	Controlar el componente	

	grafico_1.aspx	
grafico_2.aspx	Mostrar las filas y las columnas de un reporte cruzado. Adquirir y validar las que el usuario seleccionó como series del gráfico	
grafico_2.aspx.cs	Controlar el componente grafico_2.aspx	
grafico_3.aspx	Mostrar las características de un gráfico y adquirir las seleccionadas por el usuario	
grafico_3.aspx.cs	Controlar el componente grafico_3.aspx	
grafico_4.aspx	Adquirir el nombre, título y subtítulo que da el usuario a los gráficos diseñados	
grafico_4.aspx.cs	Controlar el componente grafico_4.aspx	
v_reporte.aspx	Mostrar la información de un reporte determinado y las opciones imprimir, exportar y graficar.	
v_reporte.aspx.cs	Controlar el componente v_reporte.aspx	
v_grafico.aspx	Mostrar el gráfico predefinido de un reporte determinado	
v_grafico.aspx.cs	Controlar el componente v_grafico.aspx	
distribución.aspx	Mostrar los tipo de frecuencia de una distribución. Adquirir y validar la planificación hecha por el usuario	
distribución.aspx.cs	Controlar el componente distribución.aspx	
i_reporte.aspx	Mostrar los datos de impresión. Adquirir y validar los seleccionados por el usuario	
i_reporte.aspx.cs	Controlar el componente i_reporte.aspx	
exp_reporte.aspx	Mostrar las herramientas a las que se puede exportar un reporte	
exp_reporte.aspx.cs	Controlar el componente exp_reporte.aspx	
eliminar.aspx	Decidir la página a mostrar,	

	según el elemento seleccionado por el usuario para eliminar: carpeta o reporte	
eliminar.aspx.cs	Controlar el componente eliminar.aspx	
e_reporte.aspx	Mostrar un mensaje de confirmación para eliminar un reporte. Adquirir la respuesta del usuario	
e_reporte.aspx.cs	Controlar el componente e_reporte.aspx	
e_carpeta.aspx	Mostrar un mensaje de confirmación para eliminar una carpeta con todo su contenido. Adquirir la respuesta del usuario	
e_carpeta.aspx.cs	Controlar el componente e_carpeta.aspx	
mover.aspx	Mostrar las posibles ubicaciones que puede tener un reporte al desear moverlo. Adquirir y validar el movimiento efectuado por el usuario	
mover.aspx.cs	Controlar el componente mover.aspx	
modificar.aspx	Mostrar los datos de un reporte determinado. Adquirir y validar los datos modificados	
modificar.aspx.cs	Controlar el componente modificar.aspx	
CUTareas.ascx	Interacción del usuario con las tareas que puede realizar dentro de la aplicación en un contexto determinado.	
CUTareas.ascx.cs	Controlar el componente CUTareas.ascx	CPCollcono, CPColTexto
CUInfo.ascx	Mostrar la información general de elemento señalado por el usuario.	
CUInfo.ascx.cs	Controlar el componente CUInfo.ascx	
CUContenidoCarpeta.ascx	Interacción del usuario con los reportes del sistema	
CUContenidoCarpeta.ascx.cs	Controlar el componente CUContenidoCarpeta.ascx	

CUNavegadorCarpeta.ascx	Interacción del usuario con las carpetas del sistema	
CUNavegadorCarpeta.ascx.cs	Controlar el componente CUContenidoCarpeta CUNavegadorCarpeta.ascx	
CUSelCarpeta.ascx	Interacción del usuario con las carpetas a donde puede mover un reporte determinado	
CUSelCarpeta.ascx.cs	Controlar el componente CUSelCarpeta.ascx	
CUVisorReporte.ascx	Mostrar la información de un reporte seleccionado por el usuario	
CUVisorReporte.ascx.cs	Controlar el componente CUVisorReporte.ascx	
CUVisorGrafico.ascx	Mostrar el gráfico de un reporte determinado	
CUVisorGrafico.ascx.cs	Controlar el componente CUVisorGrafico.ascx	
CPUAsistente.cs	Sincronización y organización de las interacciones del usuario con cualquier tipo de asistente.	CPUAsistente
CPUAsistenteReporte.cs	Sincronización y organización de las interacciones del usuario con un asistente para crear reportes.	CPUAsistenteReporte
CPUAsistenteReporteMD.cs	Sincronización y organización de las interacciones del usuario con un asistente para crear reportes cruzados.	CPUAsistenteReporteMD
CPUAsistenteReporteL.cs	Sincronización y organización de las interacciones del usuario con un asistente para crear reportes de listado.	CPUAsistenteReporteL
CPUAsistenteGrafico.cs	Sincronización y organización de las interacciones del usuario con un asistente para crear gráficos.	CPUAsistenteGrafico
CPUInfo.cs	Gestionar el proceso de mostrar información.	CPUInfo
asistente_reporteC.xml	Describe los pasos de un asistente para crear reportes cruzados.	

asistente_reporteL.xml	Describe los pasos de un asistente para crear reportes de listado.	
asistente_grafico.xml	Describe los pasos de un asistente para crear gráficos a partir de un reporte.	
CNReporte.cs	Implementación de la lógica del negocio referente a un reporte, independizándola del almacén de datos y de la interacción con el usuario .	CNReporte
CNReporteL.cs	Implementación de la lógica del negocio referente a un reporte de listado, independizándola del almacén de datos y de la interacción con el usuario .	CNReporteL, CaracteristicasReporteL
CNReporteMD.cs	Implementación de la lógica del negocio referente a un reporte cruzado, independizándola del almacén de datos y de la interacción con el usuario .	CaracteristicasReporteMD, CNReporteMD
CNGrafico.cs	Implementación de la lógica del negocio referente a un gráfico, independizándola del almacén de datos y de la interacción con el usuario .	CNGrafico
CNDistribucion.cs	Implementación de la lógica referente a la distribución de un reporte, independizándola del almacén de datos y de la interacción con el usuario .	CNDistribucion
CNCarpetaReporte.cs	Implementación de la lógica referente a la lista de reportes del sistema, independizándola del almacén de datos y de la interacción con el usuario	CNCarpetaReporte
CADReporte.cs	Gestionar el acceso a los datos de los reportes	CADReporte
CADGrafico.cs	Gestionar el acceso a los datos de los gráficos	CADGrafico
CADTipoReporte.cs	Gestionar el acceso a los	CADTipoReporte

	tipos de reportes que pueden crearse	
CADTipoGrafico.cs	Gestionar el acceso a los tipos de gráficos que pueden crearse	CADTipoGrafico
CADTipoFrecuencia.cs	Gestionar el acceso a los tipos de frecuencia que puede tener la distribución de un reporte	CADTipoFrecuencia
CADCarpetaReporte.cs	Gestionar el acceso a la relación de los reportes distribuidos en carpetas.	CADCarpetaReporte
tblReporte	Almacenar los datos de los reportes	
tblGrafico	Almacenar los datos de los gráficos	
tblTipoReporte	Almacenar los tipos de reportes	
tblTipoGrafico	Almacenar los tipos de gráficos	
tblTipoFrecuencia	Almacenar los tipos de frecuencia que puede tener la distribución de un reporte	
tblCarpetaReporte	Almacenar la ubicación de los reportes por carpetas	

3.8 Conclusiones

La construcción de este capítulo ha sido de gran importancia pues los Diagramas de Clases y de Interacción constituyen una guía excelente y confiable para la implementación de cada caso de uso. Además se logró arribar con facilidad al diseño de la Base de Datos que almacenará la información del sistema. Por otra parte se llega a la conclusión de que es necesario que la aplicación cumpla con estándares de codificación y principios de diseño que ya han sido estudiados y probados, lo cual garantiza que pueda ser mejor comprendido por cualquier programador y que tenga un mayor éxito entre los usuarios.

CONCLUSIONES

Como resultado de la investigación llevada a cabo para el desarrollo de este documento y del propio sistema que se describe en su contenido, se arriba a las siguientes conclusiones:

- Los procesos académicos son de naturaleza dinámica y por tanto están sujetos a constantes cambios a tener en cuenta a la hora de automatizarlos.
- El sistema desarrollado garantiza a Akademos un período de explotación prolongado, pues no se limite a brindar sólo un conjunto de reportes fijos sino que permite diseñar los reportes de acuerdo a la necesidad de información de cada persona que usará el sistema.
- Cada asistente de reportes presenta sus características en dependencia del tipo de información y de formato que se va a mostrar como resultado de su empleo.
- El empleo de los gráficos en la aplicación enriquece la información brindada por la misma.
- La distribución de información a través de correo electrónico que permite el sistema, reduce el tiempo que demora un usuario en obtenerla después de haberla solicitado.
- Planificando con antelación la distribución de un determinado reporte se garantiza que llegue la información a su destino en el momento preciso.
- La utilización del Proceso Unificado de Desarrollo de Software asegura una mejor comprensión de los requisitos y su formalización.
- El empleo de la plataforma .NET y en especial el lenguaje C#, acelera la implementación del sistema.

RECOMENDACIONES

Al concluir esta iteración en el desarrollo del Sistema de Diseño y Distribución de Reportes, es necesario hacer las siguientes recomendaciones:

- Diseñar los Casos de Uso que no se detallaron en esta iteración.
- Implementar el asistente para la creación personalizada de gráficos.
- Implementar la herramienta para la distribución de los reportes.
- Incluir información referente a los profesores en el diseño de los reportes.
- Migrar el sistema a un elemento libre teniendo en cuenta la política definida por la dirección del país de basar el desarrollo de la industria del software cubana en las plataformas libres.

REFERENCIAS BIBLIOGRÁFICAS

- [1] COELLO, Sayda y VAZQUEZ, Rosa Alicia, *Enfoque Curricular de una Universidad de Nuevo Tipo en Cuba. La Universidad de Ciencias Informáticas*.
En: Pedagogía 2005,
http://www.pedagogia2005.rimed.cu/resumen_datos.asp?select=20&Submit=Buscar&offset=90, (12/4/2005).
- [2] TELOT, Julio A. y otros, *Sistema de Gestión Académica GESTACAD versión 1.0*.
En: Conferencia Metodológica de la Facultad de Ingeniería Industrial y Economía,
<http://www.umcc.cu/eu/Educacion%20Universitaria%202002/Sistema%20de%20gestion%20academica%20GESTACAD.htm>, (7/11/2004).
- [3] Corda Technologies, *Viewing Dynamic Data for Real-Time Manufacturing*,
<http://www.dmreview.com/whitepaper/wp.cfm?topicId=230006>, (24/3/2005).
- [3] FYI Corporation, *Introduction to Interactive Data Visualization*,
<http://www.dmreview.com/whitepaper/wp.cfm?topicId=230006>, (24/3/2005).
- [4] PATTON, Robert y OGLE, Jennifer, *Designing SQL Server 2000 Database for .NET Enterprise Servers*, 2001.
- [5] y [8] FERGUSON, Jeff y otros, *La Biblia de C#*, 2003.
- [6] y [7] GONZALEZ, José Antonio, *El lenguaje de programación C#*.
- [9], [10], [11], [12], [13] Microsoft Corporation, *Arquitectura de aplicaciones de .NET: Diseño de aplicaciones y servicios*, 2002,
http://download.microsoft.com/download/e/d/1/ed1bd192-a665-49e1-b97d-e283318c4294/Chapter2_Components.doc, (15/12/2004).
- [14], [15] CHARTE, Francisco, *Programación con Visual C# .NET*.
- [16] JACOBSON, Ivar y otros, *El Proceso Unificado de Desarrollo de Software*, edición en español, 2000.
- [17] www.creangel.com/uml/intro.php, (5/2/2005).
- [18] <http://www.recrytallize.com/merchant/crystal-reports/buy-crystal-reports.htm>, (23/11/2004).

BIBLIOGRAFÍA

OSA, de la José A., *Fin de curso en la Universidad de Ciencias Informáticas*, <http://www.granma.cu/espanol/2004/julio/sabado17/fin-e.html> (2/2/2005).

IZAGUIRRE, Rafael, 2002, *Guía Metodológica para la Elaboración de Informes*.

Universidad Carlos III de Madrid, *Puntos de información y atención al alumno. Situación y horarios*, <http://www.uc3m.es/uc3m/gral/IA/iasa.html> (2/2/2005).

Universidad de Murcia, *Área de Gestión Académica*, <http://www.um.es/academic> (2/2/2005).

PROGRAN Software Ganadero, <http://www.insolca.com/progran/asistente.htm>(2/2/2005).

Certia Business Intelligence Team, *Microsoft SQL Server 2000 Reporting Services vs. Business Objects Crystal Reports / Crystal Enterprise*, 2004.

TURTSCHI, Adrian y otros, *C# .NET Web Developer's Guide*, 2002.

AHMED, Mesbah y otros, *ASP .NET Web Developer's Guide*, 2002.

KRUGER, Mike, *C# Coding Style Guide*, 2003.

VILARNAU, Joan, *Las listas de distribución como respuesta a las necesidades de gestión de la información*, 2001.
http://www.uoc.edu/web/esp/articles/vilarnau/llistes_dist.html.

AGUILAR, Carlos, *Ayuda de Web Chart Control*, 2005,
<http://www.carlosaq.net/Tools/WebChart/Default.aspx>.

GLOSARIO DE SIGLAS Y TÉRMINOS

ASP: Active Server Page.

Actor: Alguien o algo, fuera del sistema o negocio que interactúa con él. Puede ser una persona u otro sistema.

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.

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.

Crystal Reports: Herramienta de la empresa Crystal Desisions para la generación de reportes que tradicionalmente ha tenido una alta integración con las plataformas de desarrollo de Microsoft.

GUI: Graphical User Interface.

Intranet: Es una adaptación de las mismas tecnologías que existen en Internet, para que sean utilizadas dentro de la red internan de una empresa u organización de forma tal que sus miembros puedan intercambiar información de todo tipo, utilizando el Web como interfaz común.

JIT: Just In Time.

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

Reporte Cruzado o de Totales: cuyo resultado es una tabla que representa en las filas un dato y en las columnas otro y las casillas del cuerpo de la tabla son los datos numéricos correspondientes al cruce de las filas y las columnas.

RAD: Rapid Application Desing.

RAM (Random Access Memory): Memoria de acceso directo. La RAM se usa para mantener los programas mientras se están ejecutando, y los datos mientras se los procesa. La RAM es volátil, lo que significa que la información escrita en la RAM desaparecerá cuando se apague la alimentación de energía del ordenador.

SQL (Structured Query Language): Conjunto estándar de comandos para gestionar bases de datos relacionales por sus mismas características relacionales.

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.

SMTP: Simple Mail Transfer Protocol.

SSL: Secure Socket Layer.

TCP/IP: Transmission Control Protocol/Internet Protocol.

Software: Palabra en inglés utilizada para indicar a los sistemas informáticos.

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 acceden a él para realizar una función determinada. Un computador funcionando como servidor puede tener operando varios software servidores para prestar servicios, por ejemplo: servidor de WWW, de FTP, de correo, etc .

Usuario: Persona que usa algo ordinariamente.

ANEXOS

Anexo # 1 Primera Pantalla del CU: Crear Reporte de Listado.

Akademos2005 Inicio Matrícula Plan de estudio Registro Profesor Reportes

Plan de estudio activo
CRD

Tareas
Dirección de Informatización
UCI

ASISTENTE PARA REPORTES (PASO 1 DE 4)
Seleccione los campos que desea incluir en el reporte

Plantillas
Hoja de matricula

Campos de la plantilla

- CI
- N.S. CI
- P. Nombre
- S. Nombre
- P. Apellido**
- S. Apellido
- Organización
- Residencia
- Número
- Apartamento

Campos seleccionados

- Hoja de matricula.CI
- Hoja de matricula.P. Nombre
- Hoja de matricula.P. Apellido

Cancelar Atrás Siguiente

Anexo # 2 Pantalla del CU: Conformar Filtro

Akademoss2005 Inicio Matrícula Plan de estudio Registro Profesor Reportes

Tareas
Dirección de Informatización
UCI

ASISTENTE PARA REPORTES (PASO 2 DE 4)

Especifique el filtro a aplicar al reporte

{ Hoja de matricula.Sexo es igual a Femenino
} Y
{ Matricula.Facultad es igual a Facultad 2
}

() Y O

Plantilla: Matricula
Campo: Facultad
Operador: es igual a
Valor: Facultad 2

Cancelar Atrás Siguiente

Anexo # 3 Segunda Pantalla del Caso de Uso: Crear Reporte de Listado

Akademoss2005 Inicio Matrícula Plan de estudio Registro Profesor Reportes

Tareas
Dirección de Informatización
UCI

ASISTENTE PARA REPORTES (PASO 3 DE 4)

Especifique el orden del reporte

Campos disponibles

Hoja de matricula.CI
Hoja de matricula.P. Nombre
Hoja de matricula.P. Apellido

Campos seleccionados

Hoja de matricula.P. Apellido
Hoja de matricula.P. Nombre

Cancelar Atrás Siguiente

Anexo # 4 Tercera Pantalla del Caso de Uso: Crear Reporte de Listado

Akademos2005 Inicio Matrícula Plan de estudio Registro Profesor Reportes

Tareas

Dirección de Informatización
UCI

ASISTENTE PARA REPORTES (PASO 4 DE 4)

Diga los datos generales del reporte:

Nombre:

Título:

Subtítulo:

Anexo # 5 Primera Pantalla del Caso de Uso: Crear Reporte Cruzado

Akademos2005 Inicio Matrícula Plan de estudio Registro Profesor Reportes

Plan de estudio activo

Tareas

ASISTENTES PARA REPORTES CRUZADOS (PASO 1 DE 2)

Seleccione los campos que desea incluir en el reporte

Filas:

Columnas:

Anexo # 6 Tercera Pantalla del Caso de Uso: Crear Reporte Cruzado

Akademoss2005 Inicio Matrícula Plan de estudio Registro Profesor Reportes

Tareas
Dirección de Informatización
UCI

ASISTENTE PARA REPORTES (PASO 4 DE 4)

Diga los datos generales del reporte:

Nombre:

Título:

Subtítulo:

Cancelar Atrás Terminar

Anexo # 7 Primera Pantalla del Caso de Uso: Consultar Reporte

Generar reporte de estado docente
Crear una carpeta
Eliminar
Mover

Información
Nombre: Mujeres de la facultad 2
Autor: UCI\emil
Título: Mujeres de la facultad 2
Dirección de Informatización
UCI

Facultad 1 Facultad 2 Facultad 3 Facultad 4 Facultad 5
Facultad 6 Facultad 7 Facultad 8 Facultad 9 Facultad 10
marian emil julioc Darien(Pepe) Prueba

co p males en bd cantidadn por prov cantida x prov- borges
d Matriulados por Facultad grupos por sexo datos delfin
listado Mujeres de la facultad 2 MARI informacion provincias

Anexo # 8 Segunda Pantalla del Caso de Uso: Consultar Reporte

Akademoss2005 Inicio Matrícula Plan de estudio Registro Profesor Reportes

Ver grafico Imprimir Ir atrás Inicio

MUJERES DE LA FACULTAD 2
TOTAL: 318

CI	P. Nombre	P. Apellido
84122921752	Yisel	Aguilera
84070829050	Ailin	Alarcón
84051412496	Iris	Alemán
84072910734	Dadienka	Alfonso
84041017618	Yislen	Aljoe
84041625653	Yanela	Almaguer
85011810019	Niurys	Alvarez
85093027456	Yisleivis	Alvarez
85081312976	Marilyn	Amador
86032205210	María	Andricáin
84111019990	Katiuska Virgen	Apiau
84091926875	Yelenis	Aponte
86121816172	AYLIENN	AQUINO

Anexo # 9 Pantalla del Caso de Uso: Actualizar Carpetas, Sección: Crear Carpeta

Akademoss2005 Inicio Matrícula Plan de estudio Registro Profesor Reportes

Tareas

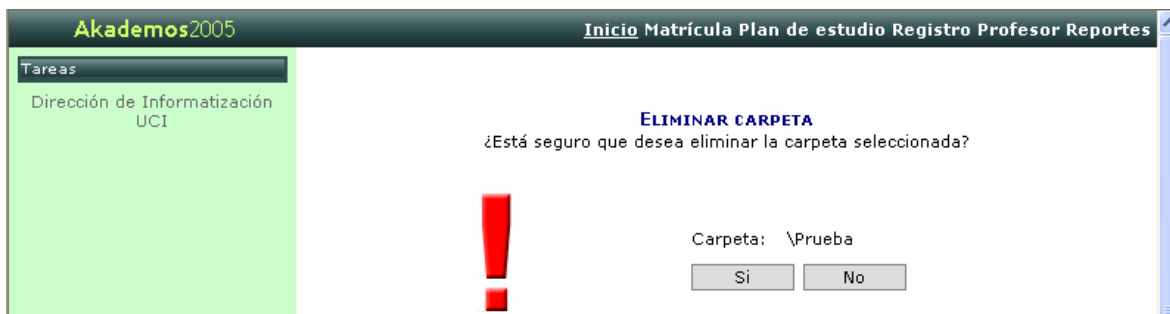
Dirección de Informatización
UCI

CREAR CARPETA

En:\[Raiz]

Nombre:

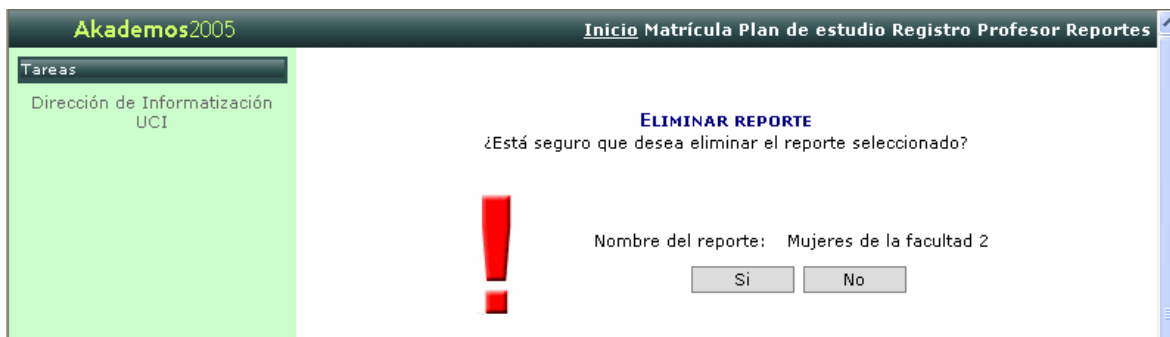
Anexo # 10 Pantalla del Caso de Uso: Actualizar Carpetas, Sección: Eliminar Carpeta



Anexo # 11 Pantalla del Caso de Uso: Actualizar Carpetas, Sección: Mover



Anexo # 12 Pantalla del Caso de Uso: Actualizar Reportes, Sección: Eliminar Reporte



Anexo # 13 Primera Pantalla del Caso de Uso: Crear Gráfico



Anexo # 14 Segunda Pantalla del Caso de Uso: Crear Gráfico

Akademos2005

Dirección de Informatización
UCI

ASISTENTE PARA GRÁFICOS (PASO 2 DE 4)

Seleccione las series de su gráfico

Series en:

filas

columnas

Campos del reporte

Series del gráfico

Cancelar Atrás Siguiete

Anexo # 15 Tercera Pantalla del Caso de Uso: Crear Gráfico

Akademos2005

Tareas

Dirección de Informatización
UCI

ASISTENTE PARA GRÁFICOS (PASO 3 DE 4)

Diga las características de su gráfico:

Leyenda: mostrar

Título del Eje X:

Título del Eje Y:

Cancelar Atrás Terminar

Anexo # 16 Cuarta Pantalla del Caso de Uso: Crear Gráfico

Akademoss2005

Dirección de Informatización
UCI

ASISTENTE PARA GRÁFICOS (PASO 4 DE 4)

Diga los datos generales del gráfico:

Nombre:

Título:

Subtítulo:

Cancelar Atrás Terminar

Anexo # 17 Pantalla del Caso de Uso: Planificar Distribución

Akademoss2005

Dirección de Informatización
UCI

PLANIFICACIÓN DE LA DISTRIBUCIÓN DE UN REPORTE

Especifique los datos de la distribución:

A:

CC:

Asunto:

Fecha y Hora:

June 2005						
<						>
Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

Periódica
 Frecuencia:
 diario
 semanal
 mensual
 anual

Hora: 06 : 25 | AM

Cancelar Aceptar