

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
DIRECCIÓN DE SOFTWARE EDUCATIVO**

# **SIRC: SISTEMA DE INCORPORACIÓN DE RECURSOS Y CONTENIDOS**

Trabajo de diploma para optar por el título de Ingeniero Informático

**Autores: Harold Ordaz Valdés  
Edier García Gutiérrez**

**Tutor: Ing. Abel Ernesto Lorente Rodríguez**

Ciudad de la Habana, abril de 2006

## DECLARACIÓN DE AUTORÍA

Por este medio declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los \_\_\_ días del mes de abril del 2006.

\_\_\_\_\_  
Firma del Autor

\_\_\_\_\_  
Firma del Autor

\_\_\_\_\_  
Firma del Tutor

## OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado “SIRC: Sistema de Incorporación de Recursos y Contenidos”, 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:

---

---

---

---

---

Y para que así conste, se firma la presente a los \_\_\_\_ días del mes  
\_\_\_\_\_ del año \_\_\_\_\_

\_\_\_\_\_  
Representante de la entidad

\_\_\_\_\_  
Cargo

\_\_\_\_\_  
Firma

\_\_\_\_\_  
Cuño

# OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

**Título:** SIRC: Sistema de Incorporación de Recursos y Contenidos.

**Autores:** Edier García Gutiérrez y Harold Ordaz Valdés.

El tutor del presente Trabajo de Diploma considera que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan.

<Aquí el tutor debe expresar cualitativamente su opinión y medir (usando la escala: muy alta, alta, adecuada) entre otras las cualidades siguientes:

- Independencia
- Originalidad
- Creatividad
- Laboriosidad
- Responsabilidad

>

<Además, debe evaluar la calidad científico-técnica del trabajo realizado (resultados y documento) y expresar su opinión sobre el valor de los resultados obtenidos (aplicación y beneficios) >

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingenieros Informáticos y propongo que se le otorgue al Trabajo de Diploma la calificación de \_\_\_\_ puntos. <Además, si considera que los resultados poseen valor para ser publicados, debe expresarlo también>

\_\_\_\_\_  
Firma

\_\_\_\_\_  
Fecha

*"No progresas mejorando lo que ya está hecho,  
sino esforzándote por lograr lo que aún queda por hacer."  
Khalil Gibrán*

## AGRADECIMIENTOS

---

Por sobre todas las cosas quiero agradecer a mi familia y muy especialmente a mis padres que me lo han dado todo para que yo este hoy aquí, siempre esperando lo mejor de mí. A Fabio y muy especialmente a Matilde por el cariño y la preocupación. A todos los profesores que he conocido hasta ahora, especialmente a mi profesora de computación del IPVCE "Federico Engels" que me motivo a estudiar la carrera. A Abel por la ayuda de todos estos años, te doy las gracias mi hermano. Y por último a nuestro Comandante por darme la oportunidad de formar parte de un programa de la Revolución tan bello y digno como lo es la Universidad de las Ciencias Informáticas.

Harold Ordaz Valdés

En primer lugar quería agradecer a mi familia y en especial a mis padres y mis abuelos que me han dado lo mejor que tengo en mi vida: su amor.

Le doy gracias a todos los profes que he tenido en mi vida de estudiante, especialmente a mi profe Catalina por considerarme su hijo, por quererme mucho y por motivarme a estudiar.

A Fabio y a Matilde, por todo el apoyo que sin dudas ha sido el mejor que hemos tenido.

Agradezco también a Abel por su ayuda en estos 2 años. Muchas gracias.

Un agradecimiento muy, muy grande a nuestro Comandante, por la obra maravillosa que es la Revolución, que nos hace hombres de bien y nos da la oportunidad de estudiar en una escuela tan bella como la UCI.

Edier García Gutiérrez

## *DEDICATORIA*

---

A mi mama por todo el amor y el cariño que me ha regalado.

A mi hermana por ser la niña de mis ojos y por todos sus triunfos.

A mi papa por ser mi guía y luz en todos los caminos que he tomado.

A la memoria de mi abuela teresita, que se que estaría muy contenta si me pudiera ver.

A Alexey, Edier, Yancy y Sasha por todos los momentos compartidos y las tantas cosas que hicimos en equipo.

A todos mis amigos por aceptarme como soy.

A todos los que han ocupado un lugar en mi vida.

Harold Ordaz Valdés

A mi padres por su amor y entrega. Por estar siempre pendientes de mi y de mis estudios.

En especial a mis abuelos, que son mis segundos padres.

A mis 3 hermanos que los quiero con la vida.

A mi prima Leye, que también es mi hermana.

A mi familia completa.

A mi segunda familia: Yami, Iraida, Julio y Yamisleidy.

A Aljadis por la amistad de siempre.

A Harold, Yancy, Alexey y Sasha por los 2 años de alegrías y sufrimientos...y los que faltan.

A Fals, Machado y Landrián...no pueden faltar.

A mis compañeros de 5to, incluyendo los graduados por la lucha cotidiana de estos años.

A mis otros amigos, que son muchos.

Edier García Gutiérrez

## RESUMEN

Este trabajo se propone estudiar la propuesta de un sistema que automatice los procesos que intervienen en la producción y que actualmente no cuentan con un soporte informático en la Dirección de Software Educativo (DSE) de la UCI. Entre los más importantes desde el punto de vista de los intereses de la producción, se encuentran los procesos de incorporación de recursos y contenidos al producto plantilla de una colección y asociado a ello la gestión de tareas asignadas a los desarrolladores para llevar un control de las actividades que realiza cada uno. También el poder gestionar Grupos de Proyecto con el objetivo de agrupar a los desarrolladores en equipos de trabajo para una mejor distribución de las tareas a realizar durante la producción. Un sistema informático garantizaría mayor eficiencia y control en todos estos procesos.

Por tanto el objetivo concreto de este trabajo consiste en desarrollar una herramienta, segura y de interfaz amigable, que dé soporte a los procesos de incorporación de recursos y contenidos, la gestión de las tareas y asignación de las mismas para brindar seguridad a los procesos, así como la gestión de Grupos de Proyecto con el objetivo de brindar facilidades para la producción y llevar un control estricto de la misma.

Este documento recoge los resultados de todo el trabajo investigativo realizado. Se identifican y describen los procesos de la DSE, especialmente aquellos que se van a automatizar; y se dan argumentos para demostrar que la situación problemática requiere de un sistema nuevo ya que ninguna otra solución satisface todas sus necesidades. Posteriormente se hace un análisis comparativo acerca de las tecnologías existentes y se seleccionan las más apropiadas. Se muestran los resultados del diseño de la propuesta del sistema y finalmente se incluye el estudio de factibilidad del proyecto.

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>1 FUNDAMENTACIÓN DEL TEMA.....</b>	<b>7</b>
1.1    Introducción.....	7
1.2    El Grupo de Plataformas Educativas dentro de la Dirección de Software Educativos..	7
1.2.1    Flujo actual de los procesos involucrados en el campo de acción.....	8
1.3    Propuesta de solución: SIRC .....	10
1.4    Objetivos propuestos.....	11
1.4.1    Objetivo general. ....	11
1.4.2    Objetivos específicos. ....	11
1.5    Conclusiones.....	11
<b>2 TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR .....</b>	<b>12</b>
2.1    Introducción.....	12
2.2    Descripción de las tendencias y tecnologías actuales sobre las que se apoya la propuesta.....	12
2.2.1    Plataforma .NET .....	12
2.2.2    Visual Studio .NET .....	13
2.2.3    .NET Framework .....	13
2.2.4    Servicios Web basados en XML. ....	16
2.2.5    XML. ....	18
2.2.6    Arquitectura de Aplicaciones.....	19
2.3    Fundamentación de la metodología utilizada.....	19
2.3.1    RUP.....	19
2.3.2    UML.....	21
2.4    Fundamentación de los lenguajes que serán utilizados.....	22
2.4.1    C#.....	22
2.4.2    Lingo.....	23
2.4.3    SQL. ....	23
2.5    Fundamentación de la selección del Gestor de Bases de Datos.....	24
2.6    Herramientas utilizadas.....	25
2.7    Conclusiones.....	25
<b>3 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA .....</b>	<b>27</b>
3.1    Introducción.....	27
3.2    Estado actual del negocio. ....	27
3.3    Modelo de Dominio. ....	28

3.3.1	Descripción del Modelo de Dominio.....	28
3.3.2	Definición de los conceptos principales del entorno del sistema. ....	29
3.4	Requisitos funcionales. ....	29
3.5	Requisitos no funcionales. ....	32
3.6	Descripción del sistema propuesta: .....	34
3.6.1	Concepción general del sistema: .....	34
3.6.2	Definición de los actores del sistema a automatizar. ....	35
3.6.3	Modelo de casos de uso del sistema. ....	36
3.6.4	Descripción de los casos de uso.....	36
3.7	Conclusiones.....	49
<b>4</b>	<b>CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA .....</b>	<b>50</b>
4.1	Introducción.....	50
4.2	Modelo de Diseño. ....	50
4.2.1	Paquete <Acceso a Datos>.....	52
4.2.2	Paquete <Gestión de Colecciones> .....	53
4.2.3	Paquete <Gestión de Ficheros del Repositorio>.....	53
4.2.4	Paquete <Gestión de Grupos de Proyecto> .....	54
4.2.5	Paquete <Gestión de Contenido> .....	55
4.2.6	Paquete <Gestión de Recursos> .....	61
4.2.7	Paquete <Gestión de Reportes e Incidencias> .....	62
4.2.8	Paquete <Gestión de Tareas> .....	63
4.2.9	Paquete <Gestión de Usuarios> .....	64
4.2.10	Paquete <Global> .....	65
4.2.11	Descripción de las principales clases del diseño. ....	66
4.3	Diagramas de secuencia.....	73
4.3.1	Diagramas de secuencia del CU “Realizar Tareas Asignadas” .....	73
4.3.2	Diagrama de secuencia del CU “Incorporar Contenido”.....	73
4.3.3	Diagrama de secuencia del CU “Incorporar Recursos”.....	73
4.3.4	Diagramas de secuencia del CU “Administrar Tareas” .....	73
4.3.5	Diagramas de secuencia del CU “Administrar Usuarios” .....	73
4.3.6	Diagramas de secuencia del CU “Mostrar Información de Reportes” .....	73
4.3.7	Diagrama de secuencia del CU “Autenticar Usuario”.....	73
4.3.8	Diagramas de secuencia del CU “Gestionar Ficheros del Repositorio” .....	73
4.3.9	Diagramas de secuencia del CU “Administrar Grupos de Proyecto” .....	73
4.4	Diseño de la base de datos.....	74

## ÍNDICE

---

4.4.1	Diagrama de clases persistentes. ....	74
4.4.2	Modelo de datos. ....	75
4.4.3	Descripción de las tablas de la bases de datos. ....	76
4.5	Principios de diseño. ....	80
4.5.1	Estándares en la interfaz de la aplicación. ....	80
4.5.2	Formatos de reportes. ....	80
4.5.3	Tratamiento de excepciones. ....	80
4.6	Estándares de codificación. ....	81
4.7	Modelo de despliegue. ....	81
4.8	Modelo de implementación. ....	82
4.8.1	Diagrama de Componentes <Paquete Acceso a Datos> ....	83
4.8.2	Diagrama de Componentes <Paquete Global> ....	83
4.8.3	Diagrama de Componentes <Paquete Servicios Web> ....	84
4.8.4	Diagrama de Componentes <Paquete Negocio> ....	85
4.8.5	Diagrama de Componentes <Paquete Presentación> ....	86
4.9	Descripción de los principales componentes. ....	87
4.10	Conclusiones. ....	87
<b>5</b>	<b>ESTUDIO DE FACTIBILIDAD</b> .....	<b>88</b>
5.1	Introducción. ....	88
5.2	Planificación. ....	88
5.3	Beneficios tangibles e intangibles. ....	91
5.4	Análisis de costos y beneficios. ....	92
5.5	Conclusiones. ....	92
	<b>CONCLUSIONES</b> .....	<b>93</b>
	<b>RECOMENDACIONES</b> .....	<b>94</b>
	<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	<b>95</b>
	<b>GLOSARIO DE TÉRMINOS</b> .....	<b>96</b>
	<b>ANEXOS</b> .....	<b>98</b>

## ÍNDICE DE FIGURAS

---

Fig. 2-1. Componentes del .NET Framework .....	14
Fig. 2-2. Los servicios Web pueden conectar a diversos tipos de aplicaciones.....	18
Fig. 3-1. Modelo de objetos del dominio .....	28
Fig. 4-1. Diagrama de paquetes del sistema .....	51
Fig. 4-2. Diagrama de clases del paquete Acceso a Datos.....	52
Fig. 4-3. Diagrama de clases del paquete Gestión de Colecciones.....	53
Fig. 4-4. Diagrama de clases del paquete Gestión de Ficheros del Repositorio.....	53
Fig. 4-5. Diagrama de clases del paquete Gestión de Grupos de Proyecto .....	54
Fig. 4-6. Diagrama de clases del paquete Gestión de Contenido Sección 1 .....	55
Fig. 4-7. Diagrama de clases del paquete Gestión de Contenido Sección 2 .....	56
Fig. 4-8. Diagrama de paquetes del paquete Gestión de Contenido .....	57
Fig. 4-9. Diagrama de clases del paquete Gestión de Contenido Temas .....	57
Fig. 4-10. Diagrama de clases del paquete Gestión de Contenido Principal .....	58
Fig. 4-11. Diagrama de clases del paquete Gestión de Contenido Maestro .....	58
Fig. 4-12. Diagrama de clases del paquete Gestión de Contenido Ejercicios.....	59
Fig. 4-13. Diagrama de clases del paquete Gestión de Contenido Biblioteca .....	60
Fig. 4-14. Diagrama de clases del paquete Gestión de Recursos .....	61
Fig. 4-15. Diagrama de clases del paquete Gestión de Reportes e Incidencias.....	62
Fig. 4-16. Diagrama de clases del paquete Gestión de Tareas .....	63
Fig. 4-17. Diagrama de clases del paquete Gestión de Usuarios .....	64
Fig. 4-18. Diagrama de clases del paquete Global.....	65
Fig. 4-19. Diagrama de clases persistentes.....	74
Fig. 4-20. Modelo de datos .....	75
Fig. 4-21. Iconos usados en la interfaz de usuario .....	80
Fig. 4-22. Modelo de despliegue del sistema .....	81
Fig. 4-23. Diagrama de paquetes del modelo de implementación .....	82
Fig. 4-24. Diagrama de componentes del paquete Acceso a Datos .....	83
Fig. 4-25. Diagrama de componentes del paquete Acceso a Datos .....	83
Fig. 4-26. Diagrama de componentes del paquete Servicios Web.....	84
Fig. 4-27. Diagrama de componentes del paquete Negocio .....	85
Fig. 4-28. Diagrama de componentes del paquete Presentación.....	86
Fig. 6-1. Modelo de casos de uso del sistema.....	98
Fig. 6-2. Realizar Tareas Asignadas (Sección "Principal").....	99
Fig. 6-3. Realizar Tareas Asignadas (Sección "Realizar Tarea") .....	100
Fig. 6-4. Realizar Tareas Asignadas (Sección "Terminar Tarea") .....	101
Fig. 6-5. Incorporar Contenido .....	102
Fig. 6-6. Incorporar Recursos .....	103
Fig. 6-7. Administrar Tareas (Sección "Principal").....	104

## ÍNDICE DE FIGURAS

---

Fig. 6-8. Administrar Tareas (Sección "Agregar Tarea" Escenario Mostrar Pantalla).....	105
Fig. 6-9. Administrar Tareas (Sección "Agregar Tarea" Escenario Agregar) .....	106
Fig. 6-10. Administrar Tareas (Sección "Eliminar Tarea") .....	107
Fig. 6-11. Administrar Tareas (Sección "Modificar Tarea" Escenario Mostrar Pantalla).....	108
Fig. 6-12. Administrar Tareas (Sección "Modificar Tarea" Escenario Modificar).....	109
Fig. 6-13. Administrar Usuarios (Sección "Principal").....	110
Fig. 6-14. Administrar Usuarios (Sección "Agregar Usuario" Escenario Mostrar Pantalla).....	111
Fig. 6-15. Administrar Usuarios (Sección "Agregar Usuario" Escenario Agregar) .....	112
Fig. 6-16. Administrar Usuarios (Sección "Eliminar Usuario") .....	112
Fig. 6-17. Administrar Usuarios (Sección "Modificar Usuario" Escenario Mostrar Pantalla).....	113
Fig. 6-18. Administrar Usuarios (Sección "Modificar Usuario" Escenario Modificar).....	113
Fig. 6-19. Mostrar Información de Reportes (Sección "Reporte del estado de las tareas").....	114
Fig. 6-20. Mostrar Información de Reportes (Sección "Reporte del estado de los productos") .....	115
Fig.6-21. Autenticar Usuario .....	116
Fig.6-22. Gestionar Ficheros del Repositorio (Sección "Bajar Ficheros") .....	117
Fig. 6-23. Gestionar Ficheros del Repositorio (Sección "Subir Ficheros") .....	117
Fig. 6-24. Administrar Grupos de Proyecto (Sección "Principal").....	118
Fig. 6-25. Administrar Grupos de Proyecto (Sección "Agregar Proyecto" Escenario Mostrar Pantalla)..	119
Fig. 6-26. Administrar Grupos de Proyecto (Sección "Agregar Proyecto" Escenario Agregar) .....	120
Fig. 6-27. Administrar Grupos de Proyecto (Sección "Eliminar Proyecto") .....	121
Fig. 6-28. Administrar Grupos de Proyecto (Sección "Modificar Proyecto" Escenario Mostrar Pantalla)	121
Fig. 6-29. Administrar Grupos de Proyecto (Sección "Modificar Proyecto" Escenario Modificar) .....	122
Fig. 6-30. Pantalla de Menú Desarrollador .....	123
Fig. 6-31. Pantalla Realizar Tareas Asignadas.....	123
Fig. 6-32. Pantalla Incorporar Contenido .....	123
Fig. 6-33. Pantalla Incorporar Recursos .....	124
Fig. 6-34. Pantalla de Menú Jefe de Proyecto.....	124
Fig. 6-35. Pantalla de Administración de Tareas .....	124
Fig. 6-36. Pantalla Asignar Tarea .....	125
Fig. 6-37. Pantalla Modificar Tarea (Por Comenzar) .....	125
Fig. 6-38. Pantalla Modificar Tarea (En Desarrollo y Reasignada) .....	126
Fig. 6-39. Pantalla Modificar Tarea (Terminada) .....	126
Fig. 6-40. Pantalla de Menú Administrador.....	127
Fig. 6-41. Pantalla de Administración de Proyectos .....	127
Fig. 6-42. Pantalla Agregar Proyecto.....	127
Fig. 6-43. Pantalla Modificar Proyecto .....	128
Fig. 6-44. Pantalla de Administración de Usuarios.....	128
Fig. 6-45. Pantalla Agregar Usuario.....	129

## ÍNDICE DE FIGURAS

---

Fig. 6-46. Pantalla Modificar Usuario.....	129
Fig. 6-47. Pantalla de Menú Personal de Proyecto .....	130
Fig. 6-48. Pantalla Modificar Contraseña.....	130
Fig. 6-49. Pantalla Menú Reporte .....	130
Fig. 6-50. Pantalla Reporte del estado de las tareas.....	130
Fig. 6-51. Pantalla de Autenticación .....	131
Fig. 6-52. Pantalla de Menú Repositorio.....	131
Fig. 6-53. Pantalla Repositorio Bajar Ficheros .....	131
Fig. 6-54. Pantalla Repositorio Subir Ficheros .....	132
Fig. 6-55. Diagrama de flujo de procesos para la creación de productos de una colección .....	136
Fig. 6-56. Arquitectura del Sistema de Multimedia Educativa (SISME) .....	137

## ÍNDICE DE TABLAS

---

Tabla 3-1. Actores del sistema. ....	35
Tabla 3-2. Descripción del CU Autenticar Usuario. ....	36
Tabla 3-3. Descripción del CU Administrar Usuarios. ....	37
Tabla 3-4. Descripción del CU Administrar Grupos de Proyecto.....	39
Tabla 3-5. Descripción del CU Administrar Tareas. ....	41
Tabla 3-6. Descripción del CU Realizar Tareas Asignadas.....	43
Tabla 3-7. Descripción del CU Incorporar Recursos. ....	45
Tabla 3-8. Descripción del CU Incorporar Contenidos. ....	46
Tabla 3-9. Descripción del CU Gestionar Ficheros del Repositorio. ....	47
Tabla 3-10. Descripción del CU Mostrar Información de Reportes. ....	48
Tabla 4-1. Descripción de la clase OleDbHelper.....	66
Tabla 4-2. Descripción de la clase SqlHelper.....	66
Tabla 4-3. Descripción de la clase GestorColeccion.....	66
Tabla 4-4. Descripción de la clase GestorProyecto.....	67
Tabla 4-5. Descripción de la clase GestorUsuario.....	67
Tabla 4-6. Descripción de la clase GestorTarea.....	68
Tabla 4-7. Descripción de la clase GestorReporte.....	69
Tabla 4-8. Descripción de la clase GestorRecursos.....	70
Tabla 4-9. Descripción de la clase GestorContenido.....	71
Tabla 4-10. Descripción de la clase Contenido.....	71
Tabla 4-11. Descripción de la clase GestorRepositorio.....	71
Tabla 4-12. Descripción de la clase GestorPlantilla.....	72
Tabla 4-13. Descripción de la tabla tblProyecto.....	76
Tabla 4-14. Descripción de la tabla tblColeccion.....	76
Tabla 4-15. Descripción de la tabla tblProducto.....	76
Tabla 4-16. Descripción de la tabla tblUsuario.....	77
Tabla 4-17. Descripción de la tabla tblTipoUsuario.....	77
Tabla 4-18. Descripción de la tabla tblTarea.....	77
Tabla 4-19. Descripción de la tabla tblOperacionTarea.....	78
Tabla 4-20. Descripción de la tabla tblEstadoTarea.....	78
Tabla 4-21. Descripción de la tabla tblTareaUsuarioo.....	79
Tabla 4-22. Descripción de la tabla tblTareaProducto.....	79
Tabla 5-1. Entradas externas.....	88
Tabla 5-2. Salidas externas.....	88
Tabla 5-3. Ficheros lógicos internos.....	88
Tabla 5-4. Consulta externa.....	89
Tabla 5-5. Puntos de Función desajustados.....	89
Tabla 5-6. Tabla de características.....	89

## ÍNDICE DE TABLAS

---

Tabla 5-7. Multiplicadores de esfuerzo: .....	90
Tabla 5-8. Factores de Escala .....	90
Tabla 5-9. Cálculos .....	91
Tabla 6-1. Descripción de los principales componentes .....	133

## INTRODUCCIÓN

### Situación problemática.

La Universidad de las Ciencias Informáticas (UCI), como ente productor y exportador de software, establece como una de sus principales líneas de la producción el desarrollo de aplicaciones multimedia educativas, tanto para su propio desarrollo de formación y aprendizaje de sus estudiantes como para las diferentes enseñanzas adscritas al Ministerio de Educación y otras instituciones tanto nacionales como extranjeras.

El Grupo de Plataformas Educativas, proyecto subordinado a la Dirección de Software Educativos, es líder en la producción de software multimedia y el poder contar con la experiencia, anteriormente acumulada, en el desarrollo de plataformas educativas para el montaje de cursos CBT como el de CorelDraw 12, se dio a la tarea de automatizar el proceso de desarrollo de aplicaciones multimedia educativas pertenecientes a las colecciones *Multisaber* y *Aprender Técnica (ETP)* para la escuela cubana. Realizado un estudio de los productos de las diferentes colecciones y principalmente de la colección *Aprender Técnica (ETP)*, se llegó a la conclusión que estos productos presentan una estructura muy similar en cuanto a diseño y funcionalidad. Dichos productos presentan módulos predeterminados con diferentes funcionalidades, dentro de los que se encuentran: Presentación (Tiene como objetivo identificar la aplicación), Temas (Permitir al usuario seleccionar y consultar los temas que trata el producto), Profesor (Permite a los maestros consultar artículos de actualización, programas de una asignatura, etc.), Biblioteca (Contiene una serie de elementos como imágenes, videos, palabras del glosario, efemérides etc., que pueden ser consultados por el usuario), Ejercicios (Permite la evaluación de los estudiantes en los contenidos consultados por los mismos) y Resultados (Permite visualizar las evaluaciones de los estudiantes en los ejercicios realizados).

Actualmente se está diseñando una plataforma de desarrollo de aplicaciones multimedia, SISME (Sistemas de Multimedia Educativas), que automatice los diferentes procesos para el desarrollo de los productos multimedia educativos de las diferentes colecciones (*Multisaber* y *ETP*). Uno de los procesos que debe ser automatizado es el de incorporación de recursos y contenidos.

Hasta el momento el proceso de incorporación de recursos y contenidos de un producto se lleva a cabo de forma manual, como fase final de desarrollo de un producto correspondiente a una colección. Esta actividad se realiza mediante la asignación de tareas por parte del Jefe de Proyecto a su equipo de desarrolladores para su posterior cumplimiento y control de las mismas. Para el desarrollo de un nuevo producto los desarrolladores editan el producto plantilla

cambiando todos los recursos de forma manual por nuevos recursos, por tanto ese desarrollador tiene que estar familiarizado con la estructura interna y elementos que componen el producto plantilla o previamente haber una capacitación de cómo se realiza dicho proceso manualmente, de cuáles son los recursos que pueden ser cambiados y cómo deben ser incorporados. Por otra parte la incorporación del contenido en la base de datos del producto es un trabajo muy engorroso y propenso a introducir errores e incongruencias, ya que para esto se transfiere información, referente al contenido, de la base de datos de gestión de todos los productos a la base de datos del producto plantilla, copiando los contenidos directamente en las tablas correspondientes. En este caso también es necesario contar con un desarrollador que tenga conocimiento del modelo físico de la base de datos, es decir que haya participado en el diseño de la misma o al menos tenga una noción de las relaciones entre las entidades del modelo físico e identificar cuáles son los contenidos a transferir donde corresponde. Esto da una idea de que se requiere mucho esfuerzo en recursos humanos, medios y tiempo para la preparación de los recursos humanos en caso que se necesite incorporar personal nuevo a la producción.

La situación anteriormente planteada está dada porque la Dirección de Software Educativo no cuenta con una herramienta que automatice los procesos de incorporación de recursos y contenidos a los productos que se desarrollan de las colecciones *ETP* y *Multisaber*.

### **Problema a resolver.**

Por tanto el **problema** a resolver es el siguiente: ¿Cómo facilitar los procesos de incorporación de recursos y contenidos a una plantilla producto de la colección, de forma tal, que se designen los estudiantes responsables durante el proceso productivo de la aplicación multimedia y que no necesariamente tengan un conocimiento de las herramientas que se utilizaron para su desarrollo?

### **Antecedentes.**

InstEd Software es la entidad nacional que dirige la creación del software educativo que produce el Ministerio de Educación de la República de Cuba. Consta con 16 centros de estudios y producción de software, distribuidos en las 16 Universidades Pedagógicas cubanas. Esta red de centros de software, además de poseer cerca de 2 centenas de productos educativos se dedica la investigación en el área del empleo de la computadora como medio de enseñanza-aprendizaje. Sus temas de investigación orbitan, entre otros, alrededor de la creación de modelos pedagógicos para el logro de aprendizajes desarrolladores así como las teorías de aprendizaje que los sustentan, las herramientas de autor factibles para la

implementación de este tipo de producto educativo, el impacto en la aplicación de las nuevas tecnologías en el sector educacional, etc. Los trabajos más recientes se proyectan sobre la concepción de modelos de hiperentornos de aprendizaje basados en tecnología hipertexto. Con esta concepción pedagógica se han elaborado hasta el momento 2 colecciones de software: La colección "Multisaber" para la Educación Primaria que consta de 32 software y la colección "El Navegante" para la educación Secundaria. Se prevén para fines de este curso la salida de las colecciones "Futuro" de Preuniversitario, la colección "Aprender Técnica" para la Enseñanza Técnica y Profesional (ETP) y una colección para la Enseñanza Especial. [1]

Cuando se inició la gran tarea de aprovechar los medios tecnológicos en el proceso de aprendizaje, nuestro país se vio envuelto en la tarea de producir una serie de cursos educativos destinados en un principio a la enseñanza primaria. Los 32 productos desarrollados hasta ahora de la colección "Multisaber" para la Educación Primaria fueron desarrollados por diferentes instituciones y/o universidades por todo el país. Estas instituciones desarrollaron sus productos de acuerdo a las experiencias que tenían en el uso de una u otra herramienta de software, por citar solo algunos ejemplos, se puede decir que se emplearon herramientas como Visual C, Visual Basic, Delphi, Macromedia Flash, Macromedia Director, ToolBook, entre otras. Esto implica una dificultad para la actualización y mantenimiento de los productos, pues habría que buscar a sus creadores, distribuidos en todo el país quienes tendrían que modificar el código fuente del producto puesto que los productos no fueron diseñados para permitir una actualización dinámica. Por otra parte, dada las características de cada una de las herramientas usadas, era muy difícil seguir pautas que ya estaban preestablecidas para todos los productos de la colección, por lo existen diferencias en cuanto a este aspecto de un producto a otro.

### **Objeto de estudio.**

El **objeto de estudio** lo constituyen los procesos de incorporación de recursos y contenidos dentro de la producción de software educativos de las colecciones existentes.

### **Campo de acción.**

El **campo de acción** que abarca este trabajo, incluye los siguientes aspectos:

- Analizar las etapas de la producción y definir de qué elementos depende la automatización de los procesos de incorporación de recursos y contenidos.
- Cómo se realiza la gestión de usuarios, así como la asignación de roles específicos dentro de un equipo de proyecto, para controlar los privilegios sobre el sistema.

- Cómo realizar la gestión de errores durante la incorporación de los recursos y los contenidos de los productos de la colección.
- Cómo realizar la gestión de los procesos de incorporación de recursos y contenidos a un producto plantilla de una colección.

### **Hipótesis.**

Si se desarrolla una herramienta para automatizar los procesos de incorporación de recursos y contenidos, mediante la asignación de tareas dentro de un equipo de proyecto que determinan el proceso a realizar en la producción de aplicaciones multimedia educativas de las colecciones existentes, entonces será posible facilitar, controlar y mejorar dichos procesos, además de alcanzar una mayor rapidez en la producción, con un uso menor de recursos humanos y materiales.

### **Objetivo general.**

El **objetivo general** de esta investigación es: desarrollar una propuesta de aplicación que permita, de una forma cómoda e intuitiva, facilitar a los desarrolladores del equipo de proyecto el proceso de incorporación de recursos y contenidos a los productos multimedia de las diferentes colecciones (Multisaber y ETP).

De ahí se derivan los siguientes **objetivos específicos**:

- Garantizar que se pueda gestionar el proceso de asignación de tareas a los desarrolladores, que le brinda todas las funcionalidades necesarias para realizar los procesos de incorporación de recursos y contenidos de un producto.
- Garantizar que se pueda gestionar la distribución de los equipos de proyecto: qué desarrolladores lo integran y cuál es su jefe de proyecto.
- Garantizar que se pueda gestionar la asignación de roles a los usuarios.
- Garantizar que se pueda gestionar los procesos de incorporación de recursos y contenidos a un producto.
- Garantizar el control de las incidencias que puedan ocurrir en el proceso de incorporación de recursos.

Con vistas al cumplimiento de los objetivos se propone la realización de las siguientes **Tareas**:

- Estudio y descripción de las formas actuales de gestión de procesos en la Dirección de Software Educativo para la producción.
- Análisis de cómo se encuentran en la arena internacional las tecnologías que se utilizan para llevar a cabo sistemas como el que se pretende desarrollar.

- Selección de la Metodología de Análisis y Diseño de Sistemas Informáticos, que facilite la creación y garantice la calidad del sistema.
- Selección de las herramientas para llevar a cabo el proyecto y la selección de la plataforma en la que se desarrollará la aplicación, fundamentando dicha selección.
- Diseño de una Base de datos que soporte las funcionalidades del sistema.
- Implementación de una herramienta que facilite la gestión de los procesos de incorporación de recursos y contenidos mediante la asignación de tareas que determinan el proceso a realizar durante la producción.
- Analizar si es posible aplicar los procesos de incorporación de recursos y contenidos a cualquier plantilla de una colección y modelar el sistema de tal forma que sea genérico en este sentido.

El desarrollo de esta aplicación y su posterior utilización, minimizará el tiempo de trabajo necesario para llevar a cabo cualquier operación concerniente a la incorporación teniendo como punto de partida los recursos y contenidos del producto y controlará a su vez qué persona, ubicada en un grupo de proyecto, tiene la responsabilidad de realizar dichas operaciones a través de la gestión previa de las tareas que le serán asignadas por su Jefe de Proyecto. Se logrará una mejor organización en los procesos que el sistema automatiza. Además, la interfaz amigable hará posible que cualquier persona sin necesidad de una alta calificación pueda interactuar con el sistema.

El presente documento está estructurado en cinco capítulos:

En el **Capítulo 1** se describe la situación actual, se hace referencia a sistemas automatizados que existen vinculados a este trabajo. Se tratan los principales problemas que motivan el desarrollo de este trabajo, se describe brevemente en qué consiste la propuesta y se fundamentan los objetivos que la misma se plantea.

El **Capítulo 2** hace referencia a las tendencias y tecnologías existentes en la actualidad que se deben considerar para hacer la selección de aquellas que se van a utilizar en el proyecto. Finalmente se plantea dicha selección a modo de propuesta, con cada uno de sus aspectos debidamente fundamentado.

En el **Capítulo 3** se describen los procesos actuales a través de un modelo de dominio, el cual sirve de base para determinar qué es lo que se va a desarrollar. Se determinan las funcionalidades del sistema propuesto y se describen en detalle.

El **Capítulo 4** aborda aspectos relacionados con la construcción de la solución propuesta, se modelan los diagramas de clases de diseño, se plantea el modelo de datos, y se especifican los principios para el diseño gráfico y la implementación.

El **Capítulo 5** y último, contiene el estudio de factibilidad realizado para este sistema, se enumeran los beneficios tangibles e intangibles y se analizan los costos que representa la elaboración de la propuesta de solución.

# CAPÍTULO 1

## FUNDAMENTACIÓN DEL TEMA

### 1.1 Introducción.

En el presente capítulo se hace alusión a las tres líneas fundamentales de la producción en la Dirección de Software Educativo y se profundiza en una de ellas haciendo una descripción de los procesos que intervienen. Se identifican los principales problemas que motivan esta investigación, se plantea la propuesta de solución y finalmente se fundamentan los objetivos propuestos.

### 1.2 El Grupo de Plataformas Educativas dentro de la Dirección de Software Educativos.

La Dirección de Software Educativo (DSE) de la Universidad de las Ciencias Informáticas tiene como misión fundamental la producción de software educativo según las solicitudes de la propia universidad y de otras instituciones externas como el Ministerio de Educación (MINED). Esta Dirección se ha planteado tres líneas fundamentales en la producción, las cuales son:

1. Desarrollo de los productos multimedia educativos de las diferentes colecciones para la escuela cubana.
2. Desarrollo de Multimedia o Catálogo Electrónico.
3. Desarrollo de Publicaciones Electrónicas.

Actualmente la primera línea de producción es la que mayor demanda tiene por parte del principal cliente, el Ministerio de Educación (MINED). Teniendo en cuenta que el número de productos multimedia de las diferentes colecciones es considerable y la complejidad que estos presentan de acuerdo a su estructura, contenido y requerimientos funcionales es alta, se creó el Grupo de Plataformas Educativas que tiene como principal objetivo cumplir con esta línea de producción en el menor tiempo posible, con el mínimo de recursos a utilizar tanto humanos como tecnológicos y con una alta calidad de los productos multimedia educativos para la escuela cubana.

El GPE realizó un estudio de los diferentes procesos de la producción y cómo debían desarrollarse, para facilitar su automatización. Como parte del estudio se identificaron tres procesos principales durante el desarrollo de los productos de las diferentes colecciones que responden a la necesidad de automatizar la producción, los cuales son:

- Realizar la estructuración y gestión de los contenidos de aplicaciones multimedia educativas pertenecientes a las colecciones *Multisaber* y *ETP*.
- Desarrollar o reutilizar una plantilla para el montaje dinámico de los productos de la colección.
- Y por último realizar la incorporación de recursos y contenidos al producto plantilla y así obtener un nuevo producto de la colección.

Actualmente se cuenta con dos sistemas, como parte del diseño de una plataforma de desarrollo de aplicaciones multimedia nombrada *Sistemas de Multimedia Educativas* (SISME), que automatizan las dos primeras etapas del proceso de producción respectivamente. Estos sistemas se nombran:

- *Sistema de Gestión de la Configuración y los Contenidos* (SGCC).
- *Plantilla para el montaje dinámico de los productos de la colección*.

El GPE tiene como tarea fundamental en estos momentos automatizar los procesos de incorporación de recursos y contenidos, para lo cual propone como solución el *Sistema de Incorporación de Recursos y Contenidos*, siglas SIRC, como parte del SISME.

### 1.2.1 Flujo actual de los procesos involucrados en el campo de acción.

La DSE recibe las solicitudes de los clientes para el desarrollo de un producto multimedia. En el caso del MINED, emite la solicitud y pone a su disposición los especialistas (guionistas) para el desarrollo del guión multimedia dando inicio al proceso de ***Elaboración del Guión Multimedia*** del producto de la colección. Una vez concluido el documento - Guión Multimedia - con una descripción detallada del producto, plasmando las pautas para el diseño y sus particularidades en cuanto a estructura y funcionalidad de manera general, así como los elementos de medias que debe tener, dígame textos, imágenes, sonidos, animaciones y videos, se procede a analizar dicho documento por parte de los especialistas de la dirección de audiovisual y diseñadores. Los especialistas en audiovisual se encargan de la elaboración de elementos de media que contendrá el software final y los diseñadores de diseñar interfaces para la aplicación, que agrupan los elementos que conformarán la aplicación como: animaciones, botones, etc. en prototipos de pantalla funcional. Ya concluida la elaboración de las medias y los diseños, se da por terminado los procesos de ***Elaboración de las medias*** y ***Elaboración del diseño de las Interfaces*** respectivamente.

A medida que se vaya avanzando en el proceso de ***Elaboración del diseño de las Interfaces*** y se obtengan los prototipos de pantallas de los diferentes módulos del software -

Presentación, Temas, Profesor, Biblioteca, Ejercicios y Resultados en el caso de la colección de *ETP* - se le da comienzo al proceso de **Estructuración y gestión de los contenidos del producto** dependiendo de lo que está plasmado en el documento - Guión Multimedia - y los diseños de interfaz del producto en particular. Durante este proceso se realiza la estructuración y gestión, a partir de los módulos y los prototipos de pantallas que conforman la colección, de la configuración de dichos módulos y prototipos de pantallas más los contenidos del producto en particular. Para esto el Grupo de Plataformas Educativas, implementó la herramienta *Sistema de Gestión de la Configuración y los Contenidos (SGCC)* para automatizar este proceso de la producción. Dentro de este proceso, el resultado de la estructuración y gestión de la configuración de los módulos y los prototipos de pantalla de la colección es reutilizable para todos los productos de una misma colección, ya que se diseña a nivel de la colección, debido a los elementos comunes que poseen sus productos en cuanto a estructura y funcionamiento, mientras que la gestión de los contenidos difiere en cada producto de acuerdo a sus recursos y los contenidos.

El SGCC almacena en una base de datos, llamada BD de Gestión de las Colecciones, los datos que se obtienen de la estructuración y gestión de los contenidos de un producto agrupados por colección. Además, los elementos elaborados, es decir recursos de media y diseño, son almacenados en un directorio de carpetas previamente definido por el sistema y a su vez es almacenado en un servidor de control de versiones conservando su estructura. A este servidor sólo pueden acceder por la red los usuarios que se designen.

Una vez elaborados y almacenados los elementos de medias y del diseño, imágenes, sonidos, videos, botones, fondos de pantalla, animaciones, etc., y procesada toda la información necesaria en el SGCC, se comienza con el proceso de **Incorporación de Recursos y Contenidos al producto plantilla**. Este proceso depende de la existencia o no del producto plantilla de la colección. De no estar implementada se procede a su desarrollo dando inicio al proceso de **Implementación del producto plantilla de la colección**, teniendo como fin el poder realizar un montaje dinámico de los productos de la colección. [Ver ANEXO V]

Este montaje dinámico depende del proceso de **Incorporación de Recursos y Contenidos al producto plantilla**. Actualmente dicho proceso se está llevando a cabo de forma manual y está organizado por Grupos de Proyectos, integrados por un número de desarrolladores – programadores – que son los responsables de realizar dicho proceso mediante tareas asignadas por su Jefe de Proyecto. Teniendo en cuenta la situación problemática se presenta una propuesta de solución con el objetivo de resolver el problema que se formuló.

### 1.3 Propuesta de solución: SIRC

Teniendo en cuenta la situación problemática planteada, se propone la elaboración de un sistema informático capaz de automatizar los procesos descritos en el problema formulado. A esta aplicación se le ha denominado **Sistema de Incorporación de Recursos y Contenidos** e incluirá fundamentalmente las siguientes funcionalidades:

- Mostrar los productos y la colección a la que pertenecen, que hayan pasado por el proceso de estructuración y gestión de los contenidos, mediante el uso del SGCC; almacenados en la BD de Gestión de las Colecciones. Sólo los Jefes de Proyecto están autorizados a ver esta información.
- Conformar Grupos de Proyecto eligiendo los usuarios desarrolladores que pertenecerán al mismo y un usuario Jefe del Grupo de Proyecto. Almacenar los datos referentes a la gestión en la base de datos del sistema. Sólo los usuarios administradores del sistema están autorizado a gestionar los datos referentes a los Grupos de Proyectos.
- Gestionar los usuarios del sistema y guardar sus datos en la BD del sistema.
- Asignar roles a los usuarios del sistema y permitir cambiarlos cada vez que se desee. Los usuarios sólo deben tener asignado un rol y no varios al mismo tiempo. El rol determina las acciones que pueden realizar sobre el sistema.
- Asignar a los Grupos de Proyectos los productos que van a desarrollar y sólo ellos pueden realizarlo. Los únicos responsables de esta gestión son los administradores del sistema.
- Gestionar y asignar una tarea a un usuario desarrollador para ser cumplida en el plazo que se le indique. La tarea determina qué proceso, incorporar recursos o contenidos, se debe realizar y a qué producto en particular. Los usuarios Jefes de Proyecto están autorizados a asignar las tareas solamente a los desarrolladores del grupo que dirigen.
- Mostrar a los usuarios desarrolladores las tareas que le fueron asignadas por su Jefe de Grupo de Proyecto. Sólo el usuario desarrollador está autorizado a realizar el proceso que determina la tarea asignada.

El SIRC debe consultar, en todo momento, la información contenida en la base de datos de Gestión de las Colecciones, contenedora de toda la información referente a la configuración de los módulos y los prototipos de pantallas de todas las colecciones almacenadas, mas los datos del contenido del todos los productos de las colecciones. Como el sistema que constituye la propuesta de solución se sustenta en la información contenida en la base de datos de Gestión de las Colecciones, requiere que dicha información sea lo más completa posible, de

modo que los procesos de incorporación de recursos y contenidos arrojen resultados correctos en beneficio de la producción. [Ver ANEXO VI]

### **1.4 Objetivos propuestos.**

Para darle respuesta a la situación problemática planteada, se propone para este trabajo un conjunto de objetivos para cumplimentar la propuesta de solución planteada en el epígrafe anterior.

#### **1.4.1 Objetivo general.**

Desarrollar una propuesta de aplicación que permita, de una forma cómoda e intuitiva, facilitar a los desarrolladores del equipo de proyecto el proceso de incorporación de recursos y contenidos a los productos multimedia de las diferentes colecciones (Multisaber y ETP).

#### **1.4.2 Objetivos específicos.**

- Diseñar un módulo para facilitar el proceso de gestión y asignación de tareas a los desarrolladores, que le brinde todas las funcionalidades necesarias para realizar los procesos de incorporación de recursos y contenidos de un producto.
- Diseñar un módulo que brinde todas las funcionalidades administrativas, que dé la posibilidad de gestionar toda la información con la que opera la aplicación: los datos de los usuarios del sistema, la organización de los grupos de proyecto, los roles de los usuarios que interactúan con el sistema.
- Implementar las funcionalidades priorizadas, sobre todo las relativas a la gestión de usuarios y grupos de proyectos, asignación de tareas y la realización de una tarea, dándole cumplimiento al proceso de incorporar los recursos y contenidos a un producto plantilla según el tipo de tarea.

### **1.5 Conclusiones.**

En este capítulo se ha descrito el objeto de estudio de este trabajo, determinándose los problemas que presenta. Se ha demostrado la necesidad de facilitar el proceso de producción de los productos de las colecciones existentes y se ha planteado una propuesta de solución, junto con los objetivos generales y específicos de este trabajo.

## CAPÍTULO 2

### TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR

#### 2.1 Introducción.

En el presente capítulo, se hace un análisis de cómo se encuentran a nivel internacional las tecnologías que pueden ser adecuadas para llevar a cabo el sistema que se pretende desarrollar. Igualmente se analiza la metodología a utilizar para el análisis y diseño del sistema, teniendo en cuenta las facilidades que puede aportar al trabajo. Y finalmente se exponen las propuestas de herramientas y tecnologías a utilizar para el desarrollo de la aplicación, teniendo en cuenta el resultado de los análisis anteriores.

#### 2.2 Descripción de las tendencias y tecnologías actuales sobre las que se apoya la propuesta.

##### 2.2.1 Plataforma .NET

Es un proyecto de Microsoft que tiene como objetivo 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 del sistema operativo, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados.

Provee los cimientos para la nueva generación de software. Utiliza los Servicios Web como un medio para poder conectar los distintos sistemas operativos, información y usuarios. Le da a los desarrolladores las herramientas y tecnologías para hacer rápidamente soluciones de negocios que involucran distintas aplicaciones, dispositivos físicos y organizaciones.

La idea central detrás de la plataforma .NET es la de servicio. Más concretamente software como servicio y de cómo construir, instalar, consumir, integrar o agregar (en federaciones) estos servicios para que puedan ser accedidos mediante Internet. Esto es posible debido a que tenemos la infraestructura de comunicación global que es Internet cada vez mas rápida y a un costo cada vez menor y además, a la capacidad de los procesadores que continúa incrementándose año tras año.

La plataforma .NET no es un solo producto. Es un conjunto de productos. Desde sistemas operativos como Windows XP, servidores de aplicaciones como SQL Server 2000, productos de oficina como Office XP, herramientas de desarrollo como Visual Studio .NET hasta servicios Web provistos por Microsoft como .NET Passport.

Entre las interfaces de programación y herramientas para diseñar, crear, ejecutar e instalar soluciones basadas en .NET se encuentran Visual Studio .NET y .NET Framework. [2]

### **2.2.2 Visual Studio .NET**

Visual Studio .NET es el entorno RAD (*Rapid Application Development* o Desarrollo Rápido de Aplicaciones), que permite utilizar .NET Framework para desarrollar aplicaciones de una forma rápida y visual, incluyendo características de desarrollo como el *IntelliSense*. Visual Studio .NET utiliza por tanto, .NET Framework. [3] Es una completa herramienta para generar e integrar con rapidez aplicaciones y servicios Web XML, lo que mejora notablemente la productividad del programador y abre las puertas a nuevas oportunidades empresariales. Está basado en la más reciente plataforma de servidor de Microsoft Windows®, lo que incorpora escalabilidad, confiabilidad y seguridad a las aplicaciones. Se han simplificado la administración y la implementación de las aplicaciones en un ambiente de producción, reduciendo así los costes totales del ciclo. [4]

Es un conjunto de aplicaciones completo para la creación tanto de aplicaciones de escritorio como de aplicaciones Web de empresa para trabajo en equipo. Aparte de generar aplicaciones de escritorio de alto rendimiento, se pueden utilizar las eficaces herramientas de desarrollo basado en componentes y otras tecnologías de Visual Studio para simplificar el diseño, desarrollo e implementación en equipo de soluciones para empresa.

Visual Basic .NET, Visual C++ .NET, Visual C# .NET y Visual J# .NET utilizan el mismo entorno de desarrollo integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes aprovechan las funciones de .NET Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y servicios Web XML. [5]

### **2.2.3 .NET Framework**

.NET Framework es el marco de trabajo con el cuál se desarrollan aplicaciones. En él se incluyen las diferentes partes del lenguaje (clases, objetos, tipos, etc.) que permiten desarrollar soluciones informáticas. [3]

Soporta múltiples lenguajes de programación y, es posible desarrollar cualquier tipo de aplicación con cualquiera de estos lenguajes: C# (C Sharp), Visual Basic, C++, J# (Java #), Jscript, además, se han adaptado lenguajes existentes, como Perl, Phyton, Fortran, Cobol, entre otros.” [6]

Si algo es el .NET Framework, es un entorno de desarrollo de nuestro tiempo, un entorno de desarrollo moderno. En él, se pueden conjugar los aspectos modernos y actuales

para cubrir las necesidades de los desarrolladores. Es un entorno capaz de resolver las necesidades de los desarrolladores de hoy, capaz incluso de trabajar con los errores que se sucedan en los programas a modo de excepciones. Es decir, se puede trabajar con los errores de una aplicación que se producen en tiempo de ejecución en cualquier momento. El trabajo con excepciones, ofrece capacidades de gestión de errores muchos mayores a las que se estaba acostumbrado con los antiguos entornos de desarrollo de Microsoft. Otra de las características de .NET Framework como entorno abierto, es que ha sido desarrollado con la pretensión de cumplir con todos los estándares actuales, siguiendo patrones de estandarización ya aprobados como el ECMA (<http://www.ecma.ch/>). Todo en el .NET Framework, cumple con los patrones de la normalización y se apoya en estándares abiertos. [3]



**Fig. 2-1. Componentes del .NET Framework**

### 2.2.3.1 CLR

*Common Language Runtime* (Motor Común de Ejecución) es el núcleo de .NET Framework. Es el motor encargado de gestionar la ejecución de aplicaciones y a las que ofrece numerosos servicios que simplifican su desarrollo y favorecen su fiabilidad y seguridad. Las principales características y servicios que ofrece el CLR son:

- Modelo de programación consistente.
- Eliminación del “infierno de las DLL”.
- Ejecución multiplataforma.
- Integración de lenguajes.
- Gestión de memoria.
- Seguridad de tipos.
- Aislamiento de procesos.
- Tratamiento de excepciones.
- Soporte multihilo.

- Distribución transparente.
- Seguridad avanzada.
- Interoperabilidad con código antiguo.

La particularidad del CLR es que se tiene un CLR distinto por cada plataforma, ya sea una plataforma Windows, Linux, etc. Esto significa que una aplicación desarrollada bajo Microsoft Windows 2000 Advanced Server por ejemplo, debería poder ejecutarse en un entorno que dispusiese de un CLR para Linux y un CLR para Windows.

Todas las aplicaciones .NET, son compiladas a un lenguaje neutral denominado IL (Intermediate Language, es decir, Lenguaje Intermedio). El CLR es capaz de compilar ese lenguaje intermedio a lenguaje máquina, específico para cada sistema en el cuál se ejecuta. Ese es el "truco" que utiliza .NET Framework para poder ejecutar una misma aplicación en Windows o Linux por ejemplo. [3]

### **2.2.3.2 MSIL**

*Microsoft Intermediate Language* (Lenguaje Intermedio de Microsoft) es el lenguaje en el que todos los compiladores generan código para la plataforma .NET en vez de generar código máquina para CPU x86 o para algún otro tipo de CPU concreta.

Antes de poder ejecutar código, se debe convertir MSIL al código específico de la CPU, normalmente mediante un compilador Just-In-Time (JIT). Common Language Runtime proporciona uno o varios compiladores JIT para cada arquitectura de equipo compatible, por lo que se puede compilar y ejecutar el mismo conjunto de MSIL en cualquier arquitectura compatible.

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.

### **2.2.3.3 BCL**

*Base Class Library* (Biblioteca de Clases Base) es importante dentro de la estructura de desarrollo .NET. Dentro del BCL se encuentra una extensa biblioteca formada por clases que proporcionarán la posibilidad de acceder a una gran cantidad de servicios. [3] Es una librería incluida en el .NET Framework formada por cientos de tipos de datos que permiten acceder a los servicios ofrecidos por el CLR y a las funcionalidades más frecuentemente usadas a la hora de escribir programas. Además, a partir de estas clases prefabricadas se pueden crear nuevas

clases que mediante herencia extiendan su funcionalidad y se integren a la perfección con el resto de clases de la BCL.

Esta librería está escrita en MSIL, por lo que puede usarse desde cualquier lenguaje cuyo compilador genere MSIL. A través de las clases suministradas en ella es posible desarrollar cualquier tipo de aplicación, desde las tradicionales aplicaciones de ventanas, consola o servicio de Windows NT hasta los novedosos servicios Web.

### **2.2.3.4 Seguridad**

La faceta más importante de cualquier entorno de desarrollo de aplicaciones distribuida es cómo maneja la seguridad. Con .NET la seguridad comienza tan pronto como se carga una clase por el CLR, ya que el cargador de clases es parte del esquema de seguridad de .NET. Cuando se carga una clase en el entorno de ejecución .NET, se comprueban los factores relacionados con la seguridad tales como reglas de accesibilidad y requisitos de autoconsistencia. Además, los chequeos de seguridad aseguran que un trozo de código tenga las credenciales apropiadas para acceder a ciertos recursos. El código de seguridad asegura la determinación de los roles y la información de identidad. Estos chequeos de seguridad incluso se extienden a procesos y más allá de los límites de la máquina para asegurar que los datos sensibles no estén comprometidos en entornos de ejecución distribuidos. En relación con la seguridad, dentro de .NET Framework, se puede trabajar con el cifrado de información según los algoritmos SHA-1 y MD5. [3]

### **2.2.4 Servicios Web basados en XML.**

Nos encontramos en un momento especial en la industria de computación. Estamos en el inicio de una nueva manera de hacer y de integrar las aplicaciones y el elemento integrador será Internet. El tema de fondo es romper barreras entre distintas aplicaciones que tienen información, barreras entre sistemas, barreras entre los sistemas y la gente que los utiliza, barreras entre las organizaciones. A comienzos de este siglo es clara la necesidad de permitir a las computadoras conectadas a Internet comunicarse entre ellas. Desde entonces se va dando forma al nuevo modelo de computación distribuida llamado servicios Web basados en XML que se están convirtiendo en la plataforma de integración de aplicaciones gracias a los estándares abiertos y al énfasis en la comunicación y colaboración entre personas y aplicaciones. El objetivo es permitir comunicarse entre sí a sistemas heterogéneos. Esta comunicación es independiente del sistema operativo, lenguaje o modelo de programación. Para conseguir esto se desarrollaron estándares. El consorcio de Internet <http://www.w3c.org> fue el encargado de crear y mantener estos estándares.

Estos son algunos de los estándares que permiten hacer uso de los Servicios Web basados en XML:

- XML: (Lenguaje de Marcado Extensible) Es un formato universal para representar los datos.
- SOAP: (Protocolo Simple de Acceso a Objetos) Es un protocolo que permite mover los datos entre aplicaciones y sistemas. Es el mecanismo por medio del cual los servicios Web son invocados e interactúan.
- UDDI: (Descubrimiento, Descripción e Integración Universal) Lenguaje que permite publicar, encontrar y usar los Servicios Web basados en XML. Es la 'Página Amarilla' de los servicios Web, es decir un directorio para poder encontrarlos. Puede ser accedido con un explorador en <http://www.uddi.org>.
- WSDL: (Lenguaje de Descripción de Servicios Web) Lenguaje por medio del cual un servicio Web describe entre otras cosas qué hace o qué funcionalidad implementa.

La competencia es por proveer de las mejores herramientas basadas en estándares y las más fáciles y más productivas herramientas para construir las aplicaciones. La plataforma .NET es la infraestructura, los servicios y productos que Microsoft ofrece.

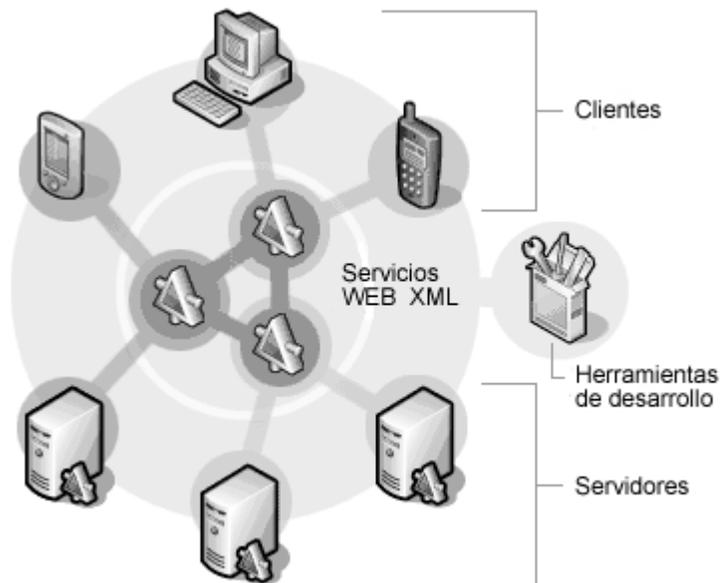
El objetivo de los servicios Web XML es lograr la interoperatividad, o sea la capacidad de que sistemas dispares puedan comunicarse y compartir información. Las aplicaciones pueden ahora, comunicarse entre sí y con los sistemas de sus socios, proveedores y clientes de forma estándar.

En resumen, con el uso de los servicios Web se integra la información que puede ser accedida desde distintos dispositivos, desde distintas plataformas de hardware o software y que puede estar guardada en distintos formatos. El lenguaje estándar para lograr esta integración es XML. [2]

La primera ventaja de los servicios Web es que SOAP es bastante menos complejo que sus antecesores, de modo que resulta más fácil que una implementación cumpla con los estándares de SOAP. Otra ventaja importante es que los servicios Web funcionan con protocolos Web estándar, como XML, HTTP y TCP/IP.

Si bien se han realizado muchos progresos para convertir los servicios Web basados en XML en una realidad, queda mucho por hacer. Actualmente, las personas obtienen resultados satisfactorios de estos servicios pero quedan aspectos pendientes como la seguridad, la administración operativa, las transacciones, el intercambio seguro de mensajes, etc. La arquitectura global de los servicios Web basados en XML ayudará a que estos servicios

avancen al siguiente nivel proporcionándoles un modelo coherente y general para agregar nuevas capacidades avanzadas a estos servicios para que sean modulares y ampliables. [7]



**Fig. 2-2. Los servicios Web pueden conectar a diversos tipos de aplicaciones.**

### 2.2.5 XML.

*EXtensible Markup Language* (Lenguaje extensible de etiquetas) (<http://www.w3.org/XML/>) ha revolucionado a los desarrolladores en los últimos años. Es un lenguaje de marcas cuya particularidad reside en que está aceptado por el W3C, es decir, es un lenguaje estándar lo cual significa que un sistema Windows puede entenderse con cualquier otro sistema mediante XML y al revés. Esta es una noticia especialmente esperanzadora para los desarrolladores, ya que no importa la plataforma en la cuál se desarrollen sus aplicaciones ni tampoco dónde se ejecuten. Lo más importante es que el lenguaje XML es un lenguaje universal. [3] El XML con todas las tecnologías relacionadas, representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. Así pues, el XML juega un papel importantísimo para los servicios Web en la actualidad, ya que es la tecnología que permitirá compartir la información de una manera segura, fiable, fácil. Además, XML permite al programador y los soportes dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas tediosas como la validación de estos o el recorrido de las estructuras corre a cargo del lenguaje y está especificado por el estándar, de modo que el programador no tiene que preocuparse por ello.

### **2.2.6 Arquitectura de Aplicaciones.**

Los éxitos pasados de los sistemas cliente/servidor, en su mayoría están relacionados con soluciones en pequeña escala, que proporcionan acceso a sistemas basados en un legado anticuado. Las cada vez más crecientes necesidades de los clientes, así como también el crecimiento acelerado de la Internet y las redes amplias, han forzado a las organizaciones a moverse hacia las empresas de mercado de soluciones. Con este movimiento se crean expectativas de crecimiento del desempeño, de la flexibilidad, y de la escalabilidad; así como de una disminución del tiempo de desarrollo, de la administración y de los costos a largo plazo. Más y más organizaciones se están moviendo a soluciones de tres capas para alcanzar estas metas. La arquitectura de tres capas supera limitaciones del diseño de dos capas, separando la presentación, el procesamiento y los datos en soportes separados. La presentación de las herramientas no necesariamente debe cambiar, sino que los cálculos y los accesos de datos se mueven a la capa intermedia para aumentar el desempeño y la seguridad, así como para disminuir los esfuerzos de implementación de las nuevas versiones o de los cambios en el código. La arquitectura de tres capas incrementa la habilidad para responder a los cambios y permite reutilizar código, simplifica el mantenimiento y hace más fácil la migración a nuevas plataformas. [8]

### **2.3 Fundamentación de la metodología utilizada.**

La calidad en el desarrollo y mantenimiento del software se ha convertido hoy en día en uno de los principales objetivos estratégicos de las organizaciones, debido a que cada vez más, los procesos principales dependen de los sistemas informáticos para su buen funcionamiento. En los últimos años se han publicado diversos estudios y estándares en los que se exponen los principios que se deben seguir para la mejora de los procesos de software.

Una metodología para el desarrollo de un proceso de software es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de Sistemas Informáticos. Por ello escoger la metodología que va a guiar el proceso de desarrollo del sistema es un paso tan importante.

#### **2.3.1 RUP.**

Para desarrollar un software se necesita una forma coordinada de trabajo. Un proceso que integre las múltiples facetas del desarrollo. Se necesita un método común, un proceso que:

1. Proporcione una guía para ordenar las actividades de un equipo.
2. Dirija las tareas de cada desarrollador por separado y del equipo como un todo.
3. Especifique los artefactos que deben desarrollarse
4. Ofrezca criterios para el control y la medición de los productos y actividades de proyectos.

El Proceso Unificado es el producto final de tres décadas de desarrollo y uso práctico. Su desarrollo sigue un camino desde 1967 con la Metodología Ericsson (Ericsson Approach), una aproximación de desarrollo basada en componentes, que introdujo el concepto de caso de uso; luego el proceso Objectory que continuó una serie de versiones desde Objectory 1.0 en 1988 a Objectory 3.8 en 1995, pasando por el proceso Objectory de Rational (publicado en 1997) hasta el Proceso Unificado de Rational (RUP: Rational Unified Process, publicado en junio de 1998).

El Proceso Unificado es una propuesta de proceso para el desarrollo de software orientado a objeto que utiliza Unified Model Language (UML) para describir todo el proceso. Está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. Sus características principales son:

- Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso. El usuario no solo representa a una persona sino que comprende además a otros sistemas.
- Centrado en arquitectura.: El concepto de arquitectura incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa como la perciben los usuarios y los inversores, y se refleja en los casos de uso. Esta además se ve influida por otros factores como la plataforma en que debe funcionar el software, arquitectura *hardware*, sistema operativo, etc. En resumen, es una vista del diseño completo con las características más importantes resaltadas, dejando los detalles de lado
- Iterativo e Incremental: RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. Propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

El Proceso Unificado es un proceso de desarrollo de Software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organización, diferentes niveles de aptitud y diferentes tamaños de proyecto.

Este proceso de desarrollo considera que el desarrollo de un sistema software debe pasar por cuatro fases, terminando cada una de ellas en un hito principal que determina decisiones importantes antes de que el trabajo pueda continuar con la siguiente fase. [9]

### **2.3.2 UML.**

El desarrollo del Unified Modeling Lenguaje (UML) empezó en octubre de 1994, cuando Grady Booch y James Rumbaugh en la Rational Software Corp. empezaron a trabajar para unificar el método Booch y la OMT (Object Modelling Technique). Un proyecto versión 0.8 del Método Unificado (UML), como se llamó desde un comienzo, salió al público en octubre de 1995. En el otoño de 1995, Ivar Jacobson se unió a la compañía y unió su esfuerzo al nuevo modelo, uniendo el OOSE al UML.

En resumen UML es el resultado de la experiencia sumada, anotaciones, y conceptos. Ya que todas las metodologías bases han tenido una aplicación extensa en el campo del la POO se han desarrollado en la práctica, tienen su historia, y han sido aplicados en una gran variedad de industrias y problemas por lo que pueden ser clasificadas por como muy maduras. UML no es una salida revolucionaria de Booch, OMT, y OOSE, sino una evolución y síntesis de estos tres.

El UML es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. UML proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto las cosas conceptuales, tales como procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes software reutilizables.

Permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar *de facto* de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y James Rumbaugh. Tiene como objetivo entregar un material de apoyo que le permita al lector poder definir diagramas propios como también poder entender el modelamiento de diagramas ya existentes.

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación. [10]

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. [9]

### **2.4 Fundamentación de los lenguajes que serán utilizados.**

#### **2.4.1 C#.**

C# es un lenguaje de programación orientado a objetos desarrollado por Microsoft y estandarizado, como parte de su plataforma .NET. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes. C# fue diseñado para combinar el control a bajo nivel de lenguajes como C y la velocidad de programación de lenguajes como Visual Basic. C#, como parte de la plataforma .NET, está normalizado por ECMA desde diciembre de 2001 (ECMA-334 "Especificación del Lenguaje C#"). El 7 de Noviembre de 2005 acabó la beta y salió la versión 2.0 del lenguaje que incluye mejoras tales como tipos genéricos, métodos anónimos, iteradores, tipos parciales y tipos anulables.

Aunque C# forma parte de la plataforma .NET, ésta es una interfaz de programación de aplicaciones; mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. [11]

C# es un lenguaje de programación poderoso, diseñado para escribir aplicaciones empresariales. Es una evolución de los lenguajes C y C++. Utiliza muchas de las características de C++ en las áreas de instrucciones, expresiones y operadores.

C# presenta considerables mejoras e innovaciones en áreas como seguridad de tipos, control de versiones, eventos y recolección de elementos no utilizados (liberación de memoria).

El lenguaje proporciona la capacidad de generar componentes de sistema duraderos en virtud de las siguientes características:

- Total compatibilidad entre COM y plataforma para integración de código existente.
- Gran robustez, gracias a la recolección de elementos no utilizados (liberación de memoria) y a la seguridad en el tratamiento de tipos.
- Seguridad implementada por medio de mecanismos de confianza intrínsecos del código.

- Plena compatibilidad con conceptos de metadatos extensibles.
- Además, es posible interactuar con otros lenguajes, entre plataformas distintas, y con datos heredados, en virtud de las siguientes características:
- Plena interoperabilidad por medio de los servicios de COM+ 1.0 y .NET Framework con un acceso limitado basado en bibliotecas.
- Compatibilidad con XML para interacción con componentes basados en tecnología Web.
- Capacidad de control de versiones para facilitar la administración y la implementación.

Como tal, C# ha demostrado ser un lenguaje adecuado para la creación de software comercial de alta calidad. Gran parte de las características del lenguaje C# se crearon con cuatro objetivos de diseño diferentes en mente:

- Disponer de un sistema de tipos unificado y simplificar el modo en que el lenguaje utiliza los tipos de valor y referencia.
- Utilizar un diseño basado en componentes establecido a través de características como comentarios XML, atributos, propiedades, eventos y delegados.
- Disponer de espacio práctico para los desarrolladores establecido a través de las capacidades únicas del lenguaje C#, entre las que se incluye la manipulación segura del puntero y la comprobación de desbordamiento, entre otras.
- Utilizar construcciones de lenguaje pragmáticas, como las instrucciones foreach y using, que aumentan la productividad del desarrollador. [12]

### **2.4.2 Lingo.**

Es el lenguaje de programación de Director. Lingo es la herramienta de Director más poderosa para crear una enriquecedora e interactiva experiencia para los usuarios. Agiliza los tiempos de desarrollo y ayuda a integrar a las producciones una interactividad única y de alto nivel. Este lenguaje de programación ofrece posibilidades como: el inspector centralizado de propiedades, el administrador visual del reparto y los comportamientos de tipo arrastrar-soltar, para realizar las producciones rápidamente y con el menor esfuerzo.

### **2.4.3 SQL.**

Structured Query Language (Lenguaje de Consulta Estructurado) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Aúna características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de

datos, de una forma sencilla. Se habla por tanto de un lenguaje normalizado que permite trabajar con cualquier tipo de lenguaje en combinación con cualquier tipo de base de datos como MS Access, SQL Server, MySQL, DB2, Ingres, Oracle, PervasiveSQL, PostgreSQL, Sybase. [13]

### **2.5 Fundamentación de la selección del Gestor de Bases de Datos.**

Un Sistema de Gestión de Bases de Datos (SGBD) puede definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad. Los SGBD permiten al programador convencional ahorrarse horas de trabajo dedicadas a la seguridad, gestión de los datos, chequeo de errores, etc.

Entre los SGBD comúnmente utilizados en el mundo se encuentran Oracle, MySQL, Microsoft SQL Server, PostgreSQL, Internase, entre otros. Todos estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional.

Como SGBD se seleccionó el Microsoft SQL Server 2000 ya que es un programa completo de base de datos y análisis. Tiene capacidad para consultar la base de datos mediante un explorador y compatibilidad con el Lenguaje de marcado extensible (XML, *Extensible Markup Language*). Además, ostenta marcas de referencia en cuanto a escalabilidad y confiabilidad, que son críticas para el éxito de bases de datos de gran tamaño. El SQL Server permite lograr una gran velocidad en el procesamiento de transacciones, y agilidad en todas sus operaciones.

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Entre sus características figuran:

- Soporte de transacciones.
- Gran estabilidad.
- Gran seguridad.
- Escalabilidad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración.

- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo accedan a la información.
- Además permite administrar información de otros servidores de datos. [14]

Visual C# cuenta con un proveedor ADO.NET nativo para SQL Server 2000. Además Microsoft lo ha desarrollado con el objetivo de explotar al máximo las características de los sistemas operativos Windows.

### **2.6 Herramientas utilizadas.**

- Visual Studio .NET por ser la herramienta más poderosa, y completa para programar cualquier aplicación en .NET, el Visual Studio .NET además permite, mediante un único IDE, la elaboración de aplicaciones Windows, bases de datos, servicios Web y otros tipos de componentes.
- Rational Rose, es la herramienta CASE que soporta de forma completa la especificación del UML. Esta herramienta propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.
- Macromedia Director MX 2004: Macromedia Director es un programa de autor que sirve para crear aplicaciones multimedia, como demostraciones, puntos de información, presentaciones interactivas, juegos, programas formativos, etc. Además, Director es también una herramienta de gran utilidad para crear animaciones en formato de video digital y elementos interactivos para páginas Web y puede incorporar fácilmente una amplia variedad de tipos de medios.
- Embarcadero Studio para la modelación de la Base de Datos. Esta aplicación permite modelar conceptualmente la Base datos y establecer relaciones entre sus entidades, la herramienta luego genera el código necesario del modelo físico, ahorrando tiempo y minimizando las posibilidades de errores.

### **2.7 Conclusiones.**

Una vez analizado las características más importantes de las tecnologías existentes y las tendencias actuales para el desarrollo de software se llega a la decisión de que el sistema que se propone puede desarrollarse .NET, utilizando como lenguaje de programación el C# y como gestor de base de datos SQL Server 2000. La decisión está basada en la potencialidad

## *CAPÍTULO 2. TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR*

---

que brinda .NET en la implementación de aplicaciones seguras y en la gestión de datos. El subsistema de incorporación de recursos se desarrollará con la herramienta de autor Macromedia Director para garantizar la integración con la plantilla general, usando XML para la comunicación entre ellos. El diseño del sistema estará basado en una arquitectura de tres capas, aprovechando las múltiples ventajas que esta brinda. Además para la capa intermedia de la aplicación se usarán varios servicios Web que se encargarán de realizar las consultas a la capa de datos para satisfacer las peticiones del cliente.

## CAPÍTULO 3

### DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

#### 3.1 Introducción.

En este capítulo se realiza el proceso de modelación del dominio para el SIRC, identificando los tipos más importantes de objetos en el contexto del sistema, así como los eventos que suceden en el entorno en el que trabaja el sistema y el diagrama de clases del modelo de objetos. Se describen los requerimientos funcionales y no funcionales del SIRC. Por último se realiza una descripción del sistema propuesto partiendo de la concepción general del sistema para la posterior descripción de los casos de uso del sistema y la representación de los diagramas de casos de uso asociados la mismo, todos estos artefactos separados por paquetes.

#### 3.2 Estado actual del negocio.

Los especialistas de la Dirección de Software Educativo se reúnen y deciden basados en las condiciones existentes, como la cantidad de programadores y la preparación de los mismos, los guiones de contenido entregados, los diseños que se han entregado, qué colección y qué productos de esa colección se desarrollarán. Una vez identificado los productos de la colección se crean Grupos de Proyectos en coordinación con los Vice-Decanos de Producción de las facultades responsables. Los Grupos de Proyectos están integrados por varios desarrolladores y son dirigidos por un Jefe de Proyecto. Se les asigna a los Jefes de Proyecto los productos que su equipo de trabajo va a desarrollar y es el encargado de distribuirlos entre los desarrolladores subordinados. Se escoge el producto plantilla de la colección para el desarrollo de los productos multimedia. Este producto plantilla se basa en las pautas establecidas para los productos de la colección. El Jefe de Proyecto decide qué producto se le asigna a un desarrollador. Se le explica al desarrollador como funciona el producto, si este previamente no trabajó en la confección del producto plantilla, y qué recurso puede cambiar en los cast externos que conforman a la plantilla, contenedores de todos los recursos del producto multimedia. Se le explica al desarrollador dónde encontrar los recursos que debe cambiar. Estos deben estar almacenados en un repositorio dentro de un directorio de carpetas previamente definido para el producto plantilla.

El desarrollador incorpora el contenido en la base de datos definida para el producto plantilla y cambia los recursos por los nuevos recursos correspondientes al producto que se va

a desarrollar. Estos cambios se realizan sobrescribiendo los members de la Ventana de Cast de la plantilla previamente desarrollada en Macromedia Director.

### 3.3 Modelo de Dominio.

Debido al bajo nivel de estructuración que presenta el negocio que se está estudiando y que está altamente centrado en tecnologías informáticas, se propone un modelo conceptual, ya que permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Esto ayuda a los usuarios, clientes y desarrolladores e interesados, a utilizar un vocabulario común para poder entender el contexto en que se sitúa el sistema. Para capturar correctamente los requisitos y poder construir un sistema correcto se necesita tener un firme conocimiento del funcionamiento del objeto de estudio. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema.

#### 3.3.1 Descripción del Modelo de Dominio.

El modelo del dominio se describe mediante diagramas UML, específicamente con un diagrama de clases conceptuales significativas en el dominio del problema.

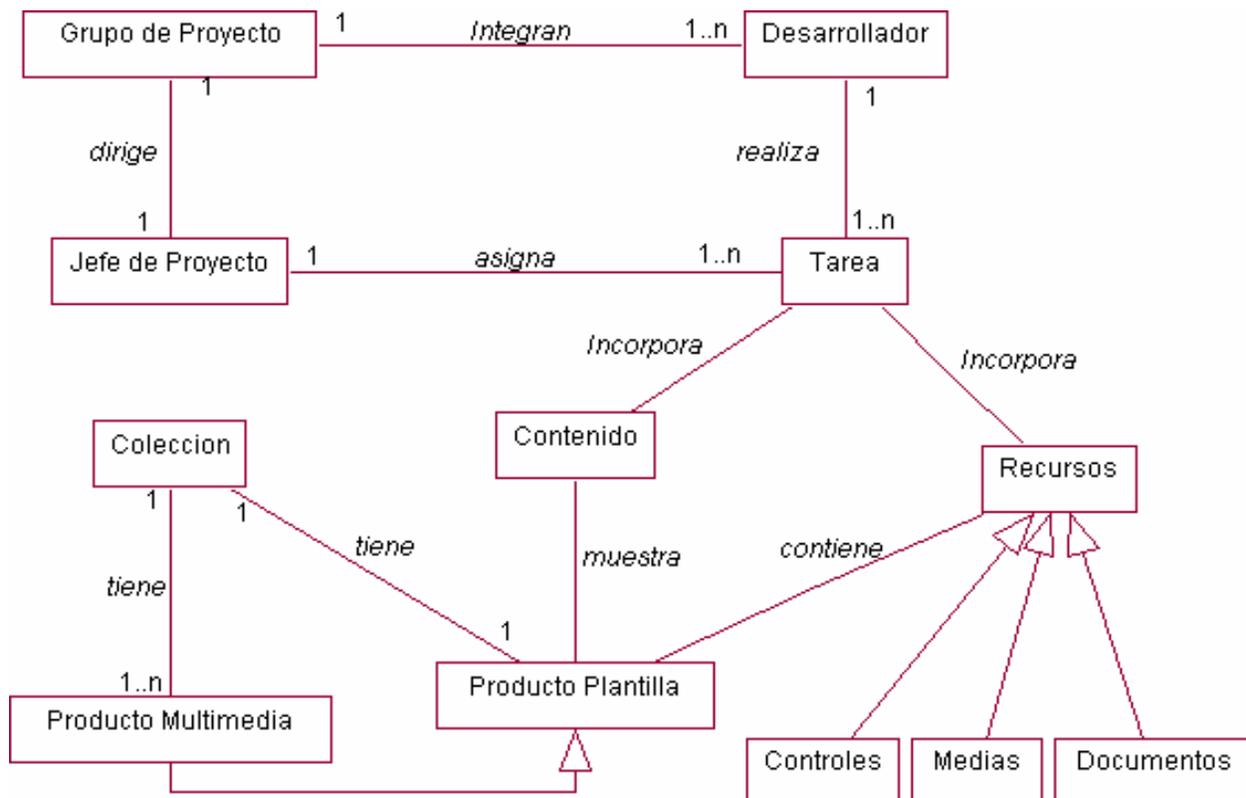


Fig. 3-1. Modelo de objetos del dominio

### 3.3.2 Definición de los conceptos principales del entorno del sistema.

- **Grupo de Proyecto:** Equipo de desarrollo formado por varios desarrolladores y un Jefe de Proyecto.
- **Desarrollador:** Persona que pertenece a un Grupo de Proyecto y que tiene la responsabilidad de realizar tareas de incorporación de recursos y contenidos cuando se le indique.
- **Jefe de Proyecto:** Persona que dirige un Grupo de Proyecto y asigna tareas a los desarrolladores.
- **Tarea:** Actividad que consiste en realizar el proceso de incorporación de recursos o contenidos al Producto Plantilla.
- **Contenido:** Toda la información que será almacenada (proceso de incorporación de contenido) en la base de datos de contenido del Producto Plantilla.
- **Recursos:** Todas las medias, controles y documentos que serán incorporados en el Producto Plantilla
- **Medias:** Imágenes, sonidos, videos y animaciones.
- **Controles:** Objetos que conforman los prototipos de pantalla del diseño de una colección y que interactúan con el usuario. Poseen un diseño particular por cada producto de la colección.
- **Documentos:** Ficheros de texto en formato de Word o PDF.
- **Producto Plantilla:** Herramienta para el montaje dinámico de los productos de la colección. Es un producto de la colección que cumple con todas las pautas generales establecidas y el funcionamiento general previsto.
- **Colección:** Serie de productos (cursos) que siguen o se rigen por pautas previamente establecidas por guionistas, diseñadores y programadores. Las pautas son las mismas en todos los productos de la colección.
- **Producto Multimedia:** El resultado de incorporar al Producto Plantilla sus recursos y contenidos.

### 3.4 Requisitos funcionales.

#### *Autenticación de usuarios.*

##### R.1 Autenticar usuario.

- 1.1 Introducir nombre de usuario y contraseña.
- 1.2 Encriptar contraseña.
- 1.3 Verificar nombre de usuario y contraseña.

R.2 Para una cuenta de usuario brindar privilegios de acceso a las diferentes funciones que realizara el sistema según su rol.

### ***Administración de usuarios.***

R.3 Crear cuenta de usuario.

3.1 Introducir los datos para registrar una cuenta de usuario:

3.1.1 Introducir nombre de usuario, contraseña, nombre y apellidos, correo electrónico y tipo de usuario.

3.1.2 Encriptar la contraseña.

3.2 Seleccionar de una lista el rol que juega en el sistema.

3.3 Enviar un correo electrónico con los datos de la cuenta a dicho usuario.

R.4 Modificar cuenta de usuario.

4.1 Introducir el nombre de la cuenta de usuario.

4.2 Encriptar la nueva contraseña si esta se modifica.

R.5 Eliminar cuenta de usuario.

5.1 Introducir el nombre de la cuenta de usuario.

R.6 Especificar el rol a una cuenta de usuario.

### ***Administración de Grupos de Proyectos.***

R.7 Crear Grupo de Proyecto.

7.1 Introducir los datos necesarios:

7.1.1 Designar el Jefe de Proyecto que asignará las tareas a dicho grupo.

7.1.2 Definir los Desarrolladores que conformarán dicho grupo.

R.8 Eliminar Grupo de Proyecto.

8.1 Introducir el nombre del Grupo de Proyecto.

R.9 Modificar Grupo de Proyecto.

9.1 Cambiar el Jefe de Proyecto

9.2 Agregar un Desarrollador al grupo.

9.3 Eliminar un Desarrollador del grupo.

9.4 Agregar un nuevo producto al proyecto.

9.5 Eliminar un producto asignado al Grupo de Proyecto.

### ***Gestión de Tareas.***

R.10 Asignar nueva tarea.

10.1 Introducir datos de la tarea.

10.1.1 Descripción, fecha de vencimiento, producto a desarrollar.

### CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

---

10.2 Asignar la tarea a un usuario responsable y la operación asociada: incorporar contenido o incorporar recursos.

10.2.1 Enviar un correo electrónico con los datos correspondientes de dicha tarea al usuario responsable.

R.11 Modificar tarea asignada a un Desarrollador.

11.1 Los datos que se pueden modificar son fecha de vencimiento, quien es el desarrollador responsable de realizar la tarea (si la tarea está Por Comenzar) y su descripción. Si la tarea está Terminada, puede cambiarse el estado a Reasignada.

11.2 Enviar un correo electrónico con los nuevos datos al Desarrollador que se le asignó o reasignó dicha tarea. En este último caso también enviar un correo electrónico al Desarrollador que tenía la tarea asignada anteriormente.

R.12 Eliminar tareas.

12.1 Enviar un correo electrónico al Desarrollador si está dentro del plazo para cumplirla, informándole de la cancelación de la misma.

R.13 Mostrar una lista de las tareas asignadas al Desarrollador.

13.1 Mostrar el estado de la tarea (Por comenzar, En desarrollo o Reasignada).

#### ***Gestión de Contenidos.***

R.14 Mostrar una lista de las tareas cuyo tipo sea incorporar contenido y que no se han terminado aún.

14.1 Seleccionar una tarea a realizar.

14.2 Comenzar dicha tarea y pasaría a estado En desarrollo.

14.3 Dar por terminada la tarea y pasaría a estado Terminada.

R.15 Al comenzar la tarea actualizar la base de datos de contenido del producto plantilla con el contenido del producto que se seleccione.

15.1 Definir el fichero de bases de datos que almacenará el contenido correspondiente al producto.

15.2 Notificar que se terminó la tarea asociada.

#### ***Gestión de Recursos.***

R.16 Mostrar una lista de las tareas cuyo tipo sea incorporar recurso y que no se han terminado aún.

16.1 Seleccionar una tarea a realizar.

16.2 Comenzar dicha tarea y pasaría a estado En desarrollo.

16.3 Dar por terminada la tarea y pasaría a estado Terminada.

R.17 Incorporar recursos.

17.1 De un nuevo producto.

17.2 De una versión del producto.

17.3 Localizar los recursos que se incorporarán al producto plantilla.

17.4 Notificar que se terminó la tarea asociada.

R.18 Guardar un reporte con la información de los recursos que no fueron incorporados al producto plantilla.

### ***Gestión de Ficheros del Repositorio.***

R.19 Descargar ficheros del repositorio central del sistema de control de versiones *Subversion* a través del TortoiseSVN.

R.20 Descargar ficheros correspondientes al producto plantilla de una colección para el desarrollo de un nuevo producto.

R.21 Descargar ficheros correspondientes a una versión del producto de la Colección, así como el producto terminado.

R.22 Subir a un servidor central la última versión del producto en desarrollo.

### ***Mostrar Información. Generar Reportes.***

R.23 Consultar el estado de las tareas asignadas a los Desarrolladores.

R.24 Consultar el estado de un producto.

## **3.5 Requisitos no funcionales.**

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

- La interfaz de la aplicación debe ser lo más atractiva y clara posible para el usuario final.
- Debe ser amigable y de fácil comprensión en su funcionamiento permitiendo la utilización del sistema sin mucho entrenamiento.

### ***Rendimiento:***

- La eficiencia y rendimiento del producto se determinará en gran medida por la velocidad de la ejecución de los Servicios Web, que está determinada grandemente por la velocidad de transmisión que se logre en la transferencia de datos de la red.

### ***Confiablez:***

- Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario.

### **Seguridad:**

- La aplicación debe estar protegida de accesos no autorizados.
- Los usuarios autenticados solo tendrán acceso a las funcionalidades del sistema según el rol que le sea asignado.
- Las contraseñas no se guardan permanentemente sin encriptar.

### **Requisitos de Implementación:**

- Emplear la tecnología .Net como plataforma de desarrollo y de apoyo a la solución propuesta la tecnología de Macromedia Director MX 2004 en cualquiera de sus versiones.
- Utilizar Microsoft SQL Server 2000 como Sistema de Gestión de Base de Datos.
- Implementar la gestión de datos a través de Web Services.

### **Requisitos de Software:**

Cliente:

- Sistema operativo con interfaz gráfica y soporte para red.
- El sistema debe correr sobre la plataforma Microsoft .NET Framework 1.1 o versiones superiores.
- Cliente de control de versiones TortoiseSVN.

Servidor:

- Windows 2000 Professional Service Pack 2.0 o superior.
- SQL Server 2000.
- Sistema de control de versiones *Subversión*.
- Servidor de paginas Web IIS.

### **Requisitos de 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:

- 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.

### **3.6 Descripción del sistema propuesta:**

#### **3.6.1 Concepción general del sistema:**

El Sistema de Incorporación de Contenido y Configuración es un sistema que formará parte de la plataforma de desarrollo de multimedia educativas (SISME) para las colecciones Multisaber y ETP. Esta plataforma está formada por tres sistemas:

- El Sistema de Gestión de Configuración y Contenido (SGCC) encargado de la gestión del contenido y la configuración de los prototipos de pantalla de los módulos de un producto de estas colecciones.
- Plantilla General de productos de la colección (Producto Plantilla) que contendrá todas las pautas generales establecidas y el funcionamiento general previsto para todos los productos de una colección.
- Sistema de Incorporación de Contenido y Configuración.

Este sistema permitirá, a partir de toda la información gestionada con el SGCC y tomando como base el producto plantilla, desarrollar un nuevo producto e incorporarle todos los recursos definidos en la configuración de los prototipos de pantallas y actualizar en su base de datos el contenido que mostrará. Esta tarea será realizada por un desarrollador, a quien el Jefe de proyecto previamente le habrá asignado la tarea. El sistema tendrá acceso a la base de datos del SGCC de la colección, donde podrá leer la estructura de un producto, sus módulos y los prototipos de pantalla de cada uno y podrá hacer una agrupación de todos los recursos que necesita el nuevo producto. El sistema además constará de políticas de seguridad otorgando a cada usuario los privilegios que le corresponden según el rol que se le asigne. Partiendo de esto, existirá un Jefe de proyecto que le asigne tareas a los desarrolladores y estos la cumplirán. La tarea consistirá en incorporar recursos o incorporar el contenido de un producto determinado al Producto Plantilla. Por otro lado, existirá un administrador del sistema que gestione todas las cuentas de usuarios y sus privilegios. Estos roles se derivan del proceso de producción definido en la DSE, en el que un administrador puede ser un Vice-decano de Producción, quien es el encargado por la facultad de designar los estudiantes y profesores que formarán un Grupo de Proyecto. Por otro lado los estudiantes constituyen los Desarrolladores

en el contexto del sistema y los profesores serían los Jefes de Proyecto. Se permite con esto que se tenga un control de quién utiliza el sistema y qué es lo que hace, restringiendo su uso a aquellos que estén autorizados a hacerlo.

### 3.6.2 Definición de los actores del sistema a automatizar.

Los actores representan los usuarios del sistema y otras aplicaciones que interactúan con él, es decir, representan terceros fuera del sistema, que interactúan con éste. Estos suelen corresponderse con trabajadores o actores del negocio. En el ANEXO I se puede observar la representación UML de los actores del sistema propuesto, que se describen en la siguiente tabla.

**Tabla 3-1. Actores del sistema.**

Actores	Justificación
Administrador del Sistema	Es el encargado de la administración general del sistema. Tiene todos los privilegios para administrar usuarios y proyectos.
Jefe de Proyecto	Es el encargado de la gestión de las tareas que serán asignadas a su equipo de trabajo.
Desarrollador	Es el encargado de la gestión de contenido y recursos.
Personal de Proyecto	Generalización del Jefe de Proyecto y de los Desarrolladores.
Usuario	Rol ficticio para generalizar los roles que participan en el caso de uso Autenticación de usuario
TortoiseSVN	Es un cliente gratuito de código abierto para el sistema de control de versiones <i>Subversion</i> . Es el encargado de manejar ficheros y directorios locales y por la red. Los ficheros se almacenan en un repositorio central de <i>Subversion</i> .

### 3.6.3 Modelo de casos de uso del sistema.

Los casos de uso del modelo de sistema son fragmentos de funcionalidad del sistema. En ellos se describe la secuencia determinada de eventos que realiza un actor en interacción con la aplicación. En el ANEXO I se muestra el diagrama de casos de uso del sistema.

### 3.6.4 Descripción de los casos de uso.

A continuación se describen los casos de uso del sistema, determinados, para satisfacer los requerimientos funcionales del sistema:

**Tabla 3-2. Descripción del CU Autenticar Usuario.**

<b>Caso de Uso #1</b>	<b>Autenticar Usuario.</b>	
Actores	Usuario	
Resumen	El caso de uso se inicia cuando el usuario inicia la aplicación. El usuario debe entrar su nombre y contraseña, el sistema comprueba si son correctos y en caso de que lo sean envía al usuario a la pantalla que le corresponde según su rol.	
Propósito	Garantizar que en todo momento se conozca quién está accediendo al sistema.	
CU asociados		
Precondiciones	El usuario está registrado en el sistema.	
Descripción		
Interfaz	Ver ANEXO III Fig. 6-51	
<b>Flujo Normal de Eventos</b>		
Acción del Actor	Respuesta del Sistema	
1. El usuario inicia la aplicación.	1.1 Muestra una ventana con los controles necesarios para que el usuario pueda introducir su nombre de usuario y contraseña.	
2. El usuario entra sus datos (nombre de usuario y contraseña) y acepta la operación.	2.1 Valida los datos del usuario y le informa que ya está autenticado remitiéndolo a la pantalla que le corresponde según su rol. En caso de que los datos del usuario no son correctos pasa al CA1.	
Cursos Alternos	CA1: Muestra un mensaje de error y regresa al paso 2.	
Requerimientos	R1, R2	

## CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Funcionales	
Poscondiciones	El usuario queda autenticado.

**Tabla 3-3. Descripción del CU Administrar Usuarios.**

<b>Caso de Uso #2</b>	<b>Administrar Usuarios.</b>	
Actores	Administrador de Sistema.	
Resumen	Para interactuar con la aplicación es necesario estar registrado en el sistema. El caso de uso se inicia cuando el Administrador de Sistema se autentifica y decide crear, modificar o eliminar una cuenta de usuario. En caso de que se cree una cuenta se entran los datos requeridos y se procede al registro. Los usuarios que no son Administrador de Sistema pueden modificar solo su contraseña. El Administrador de Sistema puede modificar los datos de cualquier usuario.	
Propósito	Administrar cuentas de usuario del sistema (crear, modificar y eliminar usuarios).	
CU asociados		
Precondiciones	El Administrador de Sistema seleccionó las opciones de crear, modificar o eliminar usuario.	
Descripción		
Interfaz	Ver ANEXO III Fig. 6-44, 6-45, 6-46, 6-47 y 6-48	
<b>Flujo Normal de Eventos</b>		
Acción del Actor	Respuesta del Sistema	
	<ol style="list-style-type: none"> <li>1. Muestra todos usuarios que ya están registrados y algunos de sus datos.</li> <li>2. Muestra las siguientes opciones: <ol style="list-style-type: none"> <li>a) Crear un nuevo usuario (ver sección "Crear Usuario").</li> <li>b) Modificar una cuenta de usuario existente (ver sección "Modificar Usuario").</li> <li>c) Eliminar una cuenta de usuario (ver sección "Eliminar Usuario").</li> </ol> </li> </ol>	
<b>Sección "Crear Usuario"</b>		
1. Selecciona la opción: Crear Usuario.	1.1 Muestra los controles necesarios para introducir los datos (usuario, contraseña, nombre	

### CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	y apellidos, correo electrónico y tipo de usuario) del nuevo usuario a crear.
2. Introduce los datos requeridos y acepta la operación.	<p>2.1 El sistema encripta la contraseña y guarda los datos del usuario en una base de datos del sistema. En caso de que el usuario ya este registrado en el sistema pasa al CA1.</p> <p>2.2 Le envía un correo electrónico al usuario, informándole que está registrado en el sistema.</p> <p>2.3 Muestra un mensaje notificando que se ha creado un nuevo usuario.</p>
Cursos Alternos	CA1: El sistema muestra un mensaje informando que ya existe un usuario con ese nombre y regresa al paso 1.1.
<b>Sección “Modificar Usuario”</b>	
1. Selecciona un usuario y elige la opción: Modificar Usuario.	1.1 Muestra los datos que pueden ser modificados del usuario seleccionado.
2. Modifica los datos del usuario y acepta la operación.	<p>2.1 Encripta la contraseña y sobrescribe los datos del usuario seleccionado con los nuevos datos en la base de datos del sistema. En caso de que se modifique el nombre del usuario y existe un usuario registrado con ese nombre pasa al CA2.</p> <p>2.2 Notifica mediante un mensaje que se modificaron los datos del usuario seleccionado.</p>
Cursos Alternos	CA2: El sistema muestra un mensaje de error informando que ya existe un usuario registrado con ese nombre y regresa al paso 2.
<b>Sección “Eliminar Usuario”</b>	
1. El actor selecciona un usuario y elige la opción: Eliminar Usuario.	1.1 Muestra una pantalla donde le pregunta si realmente quiere eliminar el usuario seleccionado.
2. Si confirma la petición.	2.1. Elimina la cuenta de usuario de la base de datos del sistema.
3. Si no confirma la petición.	3.1. No elimina la cuenta de usuario y vuelve al paso 1 de la sección principal
Cursos Alternos	

### CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Requerimientos Funcionales	R3, R4, R5, R6
Poscondiciones	Se crea, modifica o elimina un usuario del sistema.

**Tabla 3-4. Descripción del CU Administrar Grupos de Proyecto.**

<b>Caso de Uso #3</b>	<b>Administrar Grupos de Proyecto.</b>	
Actores	Administrador de Sistema	
Resumen	Los Desarrolladores se agrupan en Grupos de Proyecto y se les asigna un Jefe de Proyecto con el objetivo de crear equipos de trabajo responsables de producir una cantidad determinada de productos de una colección. El caso de uso se inicia cuando el Administrador de Sistema solicita crear, modificar o eliminar un Grupo de Proyecto y escoge qué Desarrolladores conformarán el mismo y el Jefe de Proyecto responsable de asignar las tareas a sus Desarrolladores.	
Propósito	Agrupar a los Desarrolladores por Grupos de Proyectos.	
CU asociados		
Precondiciones	El usuario Jefe de Proyecto y los usuarios Desarrollador deben estar registrados en el sistema.	
Descripción		
Interfaz	Ver ANEXO III Fig. 6.40, 6-41, 6-42 y 6-43	
<b>Flujo Normal de Eventos</b>		
Acción del Actor	Respuesta del Sistema	
	<ol style="list-style-type: none"> <li>1. Muestra la lista de Grupos de Proyectos existentes.</li> <li>2. Muestra las siguientes opciones: <ol style="list-style-type: none"> <li>a) Crear Grupo de Proyecto (ver sección).</li> <li>b) Modificar Grupo de Proyecto (ver sección).</li> <li>c) Eliminar Grupo de Proyecto (ver sección).</li> </ol> </li> </ol>	
<b>Sección “Crear Grupo de Proyecto”</b>		
1. El actor selecciona la opción: Crear Grupo de Proyecto.	1.1 Muestra los controles necesarios para introducir los datos del Grupo de Proyecto que se	

### CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	<p>creará:</p> <ul style="list-style-type: none"> <li>- Nombre del Grupo de Proyecto.</li> <li>- Descripción del proyecto</li> <li>- Lista de Desarrolladores que no pertenecen a ningún Grupo de Proyecto.</li> <li>- Jefe de Proyecto que se encargará de asignar las tareas al Grupo de Proyecto.</li> <li>- Colección en la que va a trabajar.</li> <li>- Productos que desarrollará dentro de la colección seleccionada.</li> </ul> <p>En caso de que no haya Jefes de Proyecto y/o Desarrolladores o Colección disponibles para conformar dicho Grupo de Proyecto pasa CA1.</p>
2. Introduce los datos requeridos y acepta la operación.	2.1 Guarda los datos del Grupo de Proyecto y lo crea.
Cursos Alternos	CA1: El sistema muestra un mensaje de error indicando que no hay usuarios (Jefe de Proyectos y/o Desarrolladores) o colección disponibles para conformar el Grupo de Proyecto.
<b>Sección “Modificar Grupo de Proyecto”</b>	
1. Selecciona un Grupo de Proyecto y elige la opción: Modificar Grupo de Proyecto.	1.1 Muestra las opciones de cambiar Jefe de proyecto, agregar o eliminar usuarios, cambiar descripción y agregar o eliminar productos.
2 Realiza los cambios pertinentes y acepta la modificación.	2.1 El sistema guarda los datos del Grupo de Proyecto y lo modifica.
Cursos Alternos	
<b>Sección “Eliminar Grupo de Proyecto”</b>	
1 Selecciona un Grupo de Proyecto y elige la opción: Eliminar Grupo de Proyecto.	1.1 Muestra una pantalla donde le pregunta si realmente quiere eliminar el grupo de proyecto.

### CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

2. Si confirma la petición.	2.1 Elimina el grupo de proyecto.
3 Si no confirma la petición.	3.1 No elimina el grupo de proyecto y regresa al paso 1 de la sección principal.
Cursos Alternos	
Requerimientos Funcionales	R7, R8, R9
Poscondiciones	Se crea, modifica o elimina un proyecto del sistema

**Tabla 3-5. Descripción del CU Administrar Tareas.**

<b>Caso de Uso #4</b>	<b>Administrar Tareas.</b>
Actores	Jefe Proyecto.
Resumen	Los Jefes de Proyecto asignan tareas a los desarrolladores las cuales determinan qué actividad, ya sea incorporar contenido o incorporar recursos, se realiza sobre el producto a desarrollar. El caso de uso se inicia cuando el Jefe de Proyecto decide crear, modificar o eliminar una tarea, validando que esta se puede eliminar sólo cuando no ha sido comenzada o ya se cumplió por el desarrollador. Los Jefes de Proyecto ven las tareas que fueron asignadas por él.
Propósito	Administrar las tareas que serán asignadas a los desarrolladores.
CU asociados	
Precondiciones	El Jefe de Proyecto ha sido autenticado correctamente en el sistema. El Grupo de Proyecto que dirige el Jefe de Proyecto lo integra al menos un Desarrollador.
<b>Descripción</b>	
Interfaz	Ver ANEXO III Fig. 6-34, 6-35, 6-36, 6-37, 6-38 y 6-39
<b>Flujo Normal de Eventos</b>	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> <li>1. Muestra todas las tareas que ya están asignadas y algunos de sus datos.</li> <li>2. Muestra las siguientes opciones: <ol style="list-style-type: none"> <li>d) Crear una tarea nueva (ver sección “Crear Tarea”).</li> <li>e) Modificar una Tarea existente (ver sección “Modificar Tarea”).</li> </ol> </li> </ol>

## CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	f) Eliminar una Tarea (ver sección “Eliminar Tarea”).
<b>Sección “Crear Tarea”</b>	
1. Selecciona la opción: Crear Tarea.	<p>1.1 Muestra los controles necesarios para introducir los datos (tipo de tarea, descripción, fecha de fin, producto a desarrollar, desarrollador responsable) de la nueva tarea a crear.</p> <p>1.2 La fecha que muestra siempre es posterior a la actual en el sistema</p> <p>1.3 Los productos que muestran son sólo aquellos que tienen algún tipo de tarea que no se le ha asignado.</p> <p>1.4 Los desarrolladores que se muestran son sólo aquellos que pertenecen a su grupo de proyecto.</p>
2. Introduce los datos requeridos y acepta la operación.	2.1 El sistema muestra una pantalla con los controles necesarios para enviarle los datos de la tarea al desarrollador responsable por correo electrónico.
3. Introduce su usuario de dominio y contraseña, dirección de correo electrónico del destinatario y remitente y acepta la operación de enviar correo electrónico.	<p>3.1 El sistema valida la identidad del usuario de dominio y envía el correo electrónico al destinatario. En caso de que los datos del usuario de dominio no sean correctos pasa al CA1.</p> <p>3.2 Muestra de nuevo todas las tareas que están asignadas.</p>
Cursos Alternos	CA1: Muestra un mensaje de error y regresa al paso 2.1.
<b>Sección “Modificar Tarea”</b>	
1. El actor selecciona una tarea y elige la opción: Modificar Tarea.	1.1 Muestra los datos que pueden ser modificados de la tarea seleccionada.
2. El actor modifica los datos de la tarea y acepta la operación.	2.1 El sistema muestra una pantalla con los controles necesarios para enviarle los datos de la tarea al desarrollador responsable, por correo electrónico.
3. El actor introduce su usuario de dominio y contraseña, dirección de	3.1 El sistema valida la identidad del usuario de dominio y envía el correo electrónico al

### CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

correo electrónico del destinatario y remitente y acepta la operación de enviar correo electrónico.	destinatario. En caso de que los datos del usuario de dominio no sean correctos pasa al CA2. 3.2 Muestra de nuevo todas las tareas que están aginadas con sus modificaciones.
4. El actor cancela la operación enviar correo electrónico.	4.1 El sistema no envía el correo electrónico y muestra de nuevo todas las tareas que están asignadas.
<b>Cursos Alternos</b>	CA2: Muestra un mensaje de error y regresa al paso 2.1
<b>Sección “Eliminar Tarea”</b>	
1. El actor selecciona una tarea y elige la opción: Eliminar Tarea.	1.1 Muestra una pantalla donde le pregunta si realmente quiere eliminar la tarea.
2. Si confirma la petición.	2.1 Elimina de la base de datos del sistema la tarea que estaba asignada. Si la tarea está en el estado “Comenzada” ver el CA3 2.2 Muestra la nueva lista de tareas que están asignadas.
3. Si no confirma la petición.	3.1 No elimina la tarea y regresa al paso 1 de la sección principal.
<b>Cursos Alternos</b>	CA3: El sistema muestra un error, indicando que una tarea comenzada no se puede eliminar y regresa al paso 1 de la sección principal.
<b>Poscondiciones</b>	Se asignan tareas a los desarrolladores que deberán ser cumplidas en el plazo establecido.
<b>Requerimientos Funcionales</b>	R10, R11, R12

**Tabla 3-6. Descripción del CU Realizar Tareas Asignadas.**

<b>Caso de Uso #5</b>	<b>Realizar Tareas Asignadas</b>
<b>Actores</b>	Desarrollador
<b>Resumen</b>	Este caso de uso se inicia cuando el actor decide realizar las tareas que le fueron asignadas. El sistema le muestra una lista con las tareas que le han sido asignados hasta el momento por su Jefe de Proyecto. Una vez que se le muestren dichas tareas el actor podrá comenzar a realizar las tareas que desee.
<b>Propósito</b>	Los desarrolladores realizan las tareas que tienen pendientes o dan por

### CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	terminadas las tareas que le fueron asignadas.
CU asociados	CU Incorporar Contenido, CU Incorporar Recursos
Precondiciones	- El Desarrollador ha sido autenticado correctamente en el sistema. - El Desarrollador tenga asignada, al menos, una tarea.
Descripción	
Interfaz	Ver ANEXO III Fig. 6-30 y 6-31
<b>Flujo Normal de Eventos</b>	
Acción del Actor	Respuesta del Sistema
	1. El sistema busca todas las tareas asignadas por su Jefe de Proyecto y que no haya terminado y muestra sus datos y el estado que estas presentan (Por Comenzar, En Desarrollo, Reasignada) y muestra los controles necesarios para dar por terminada la tarea (ver sección "Terminar Tarea") o comenzar la realización de la tarea (ver sección "Realizar Tarea")
<b>Sección "Terminar Tarea"</b>	
1. El actor selecciona la tarea y elige la opción "Terminar Tarea"	1.1 El sistema cambia el estado de la tarea asignada a terminada y la elimina de la lista de tareas asignadas, sólo si no ha tenido incidencias en su realización. En caso que tenga incidencias pasa al CA1.
Cursos Alternos	CA1: Muestra un mensaje de error, indicando que la tarea no puede darse por terminada hasta tanto no se solucionen todos los problemas. Vuelve al paso 1 de la sección principal.
<b>Sección "Realizar Tarea"</b>	
1. El actor selecciona la tarea que desea realizar y elige la opción "Comenzar Tarea".	1.1 Cambia el estado de la tarea a "En Desarrollo" 1.2 El sistema determina el tipo de tarea que se va a realizar. Si es del tipo "incorporar contenido" lo remite a la pantalla de incorporación de contenido (ver CU Incorporar Contenido), si es del tipo "incorporar recursos" lo remite a la pantalla de incorporación de recursos (ver CU Incorporar Recursos)

### CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	1.3 Si hubo error en la operación anterior o se cancela, se restaura el estado de la tarea al que tenía antes, de lo contrario se mantiene su estado actual.
Cursos Alternos	
Poscondiciones	El sistema muestra a los Desarrolladores las tareas que le fueron asignadas para su posterior realización.
Requerimientos Funcionales	R13

**Tabla 3-7. Descripción del CU Incorporar Recursos.**

<b>Caso de Uso #6</b>	<b>Incorporar Recursos.</b>
Actores	Desarrollador
Resumen	Una vez que el desarrollador entra al sistema se le muestran todas las tareas que le fueron asignadas. El caso de uso se inicia cuando el desarrollador decide realizar una tarea, cuyo asunto implica incorporar los recursos de un producto. Para incorporar los recursos debe establecer la dirección local donde se encuentran los recursos y la plantilla del producto e indicar si va a desarrollar un producto nuevo o una versión del producto anteriormente generado. Durante la incorporación de recursos se pueden generar incidencias que el sistema debe almacenar para un posterior análisis y corrección.
Propósito	Incorporar recursos a un producto de la colección.
CU asociados	
Precondiciones	<ul style="list-style-type: none"> <li>- Tener asignada, previamente por su Jefe de Proyecto, una tarea que determine la actividad incorporar recursos.</li> <li>- Los recursos y la plantilla están en una carpeta local que el desarrollador previamente decidió al bajarlo del repositorio central de control de versiones.</li> </ul>
Descripción	
Interfaz	Ver ANEXO III Fig. 6-33
<b>Flujo Normal de Eventos</b>	
Acción del Actor	Respuesta del Sistema

### CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	1. El sistema le muestra los controles necesarios para entrar las direcciones locales donde se encuentra los recursos y la plantilla del producto y si va a desarrollar un producto nuevo o una versión ya comenzada.
1. El actor introduce las direcciones locales donde se encuentra la plantilla y sus recursos e indica si va a trabajar o no en una versión ya comenzada y acepta la operación. Si cancela termina el caso de uso.	<p>1.1 Si es un producto nuevo verifica que la plantilla no haya sido modificada. Si está modificada pasa al CA1.</p> <p>1.2 Incorpora los recursos a la plantilla del producto y verifica que el recurso sea el correcto.</p> <p>1.3 En caso de que ocurra alguna incidencia durante el proceso de incorporación de recursos guarda la información referente a la incidencia (una lista con los recursos que no se pudieron incorporar) y la almacena para su posterior visualización.</p> <p>1.4 Al terminar de incorporar los recursos muestra al desarrollador las incidencias que hubo.</p>
Cursos Alternos	CA1: Muestra el mensaje de error “No corresponde con la plantilla para la colección” y regresa al paso 1.
Poscondiciones	- El sistema ha incorporado los recursos existentes al producto.
Requerimientos Funcionales	R16, R17, R18

**Tabla 3-8. Descripción del CU Incorporar Contenidos.**

<b>Caso de Uso #7</b>	<b>Incorporar Contenidos.</b>
Actores	Desarrollador.
Resumen	El caso de uso se inicia cuando el desarrollador decide realizar una tarea, cuyo asunto implica incorporar los contenidos a un producto Para esto debe establecer la dirección local del fichero de base de datos del producto. El contenido se obtiene de un servidor de bases de datos de Gestión de Contenido, y se almacena en la base de datos del producto.
Propósito	Incorporar contenido a un producto de la colección.
CU asociados	
Precondiciones	El Desarrollador ha sido autenticado correctamente en el sistema.

### CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	Tiene asignada una tarea cuyo asunto determina incorporar contenido.
Descripción	
Interfaz	Ver ANEXO III Fig. 6-32
<b>Flujo Normal de Eventos</b>	
Acción del Actor	Respuesta del Sistema
	1. El sistema le muestra los controles necesarios para entrar la dirección local donde se encuentra la base de datos del producto.
1. El actor introduce la dirección local del fichero de base de datos del producto y acepta la operación. Si cancela termina el caso de uso.	1.1 Obtiene el contenido de la base de datos de Gestión y lo guarda en la base de datos del producto.
Cursos Alternos	
Requerimientos Funcionales	R14, R15
Poscondiciones	El sistema ha incorporado el contenido en la base de datos de contenidos del producto.

**Tabla 3-9. Descripción del CU Gestionar Ficheros del Repositorio.**

<b>Caso de Uso #8</b>	Gestionar Ficheros del Repositorio.
Actores	Desarrollador(inicia), TortoiseSVN
Resumen	Los recursos y el producto plantilla de la colección están almacenados en un repositorio de control de versiones previamente definido. El caso de uso se inicia cuando el actor desea bajar o subir ficheros del repositorio central del control de versiones TortoiseSVN
Propósito	Almacenar y recuperar ficheros y directorios de un repositorio.
CU asociados	
Precondiciones	
Descripción	
Interfaz	Ver ANEXO III Fig. 6-52, 6-53 y 6-54
<b>Flujo Normal de Eventos</b>	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra las opciones siguientes: - Bajar Ficheros (ver sección).

### CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	- Subir Ficheros (ver sección).
<b>Sección “Bajar Ficheros”</b>	
1. Selecciona la opción: Bajar Ficheros.	1.1 Muestra los controles para entrar la dirección local hacia donde bajará los ficheros y la ubicación dentro del repositorio.
2. Entra los datos requeridos y acepta la operación.	2.1 Pasa los parámetros necesarios al TortoiseSVN para bajar los ficheros.
Cursos Alternos	
<b>Sección “Subir Ficheros”</b>	
1. Selecciona la opción: Subir Ficheros.	1.1 Muestra los controles para entrar la dirección local donde están los ficheros y la ubicación hacia donde los subirá dentro del repositorio.
2. Entra los datos requeridos y acepta la operación.	2.1 Pasa los parámetros necesarios al TortoiseSVN para subir los ficheros.
Cursos Alternos	
Poscondiciones	Los ficheros son almacenados o recuperados del sistema de control de versiones.
Requerimientos no Funcionales	R19, R20, R21, R22

**Tabla 3-10. Descripción del CU Mostrar Información de Reportes.**

<b>Caso de Uso #9</b>	Mostrar Información de Reportes.
Actores	Personal de Proyecto
Resumen	El caso de uso se inicia cuando el Personal de Proyecto solicita ver información de cuál es el estado de las tareas, cuáles productos se han terminado y cuáles no, en qué estado de desarrollo se encuentra un producto de acuerdo a las incidencias que se generaron durante la incorporación de recursos.
Propósito	Mostrar información.
CU asociados	
Precondiciones	El actor ha sido autenticado correctamente en el sistema.
<b>Descripción</b>	
Interfaz	Ver ANEXO III Fig. 6-49 y 6-50
<b>Flujo Normal de Eventos</b>	

### CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

---

Acción del Actor		Respuesta del Sistema
		1. El sistema muestra los posibles reportes que le pueden ser mostrados al personal de proyecto.
1. El actor selecciona el reporte que desea ver.		1.1 El sistema le muestra la información correspondiente al reporte seleccionado por el actor.
Cursos Alternos		
Requerimientos Funcionales	R23, R24	
Poscondiciones		

#### 3.7 Conclusiones.

En este capítulo se hizo una descripción de la propuesta de solución a través de la modelación del dominio: la identificación de los objetos del dominio del sistema y sus relaciones; el planteamiento de los requisitos funcionales y no funcionales de la aplicación que se va a desarrollar y su modelación en términos de casos de uso de sistema. Con el desarrollo de este flujo de trabajo y los artefactos obtenidos a partir de este, se puede pasar al flujo de diseño para comenzar la construcción de la solución propuesta.

# CAPÍTULO 4

## CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

### 4.1 Introducción.

El presente capítulo expone la construcción de la solución propuesta, a través de los flujos de trabajo de diseño e implementación. Primeramente se presenta el modelo de diseño, donde son expuestas las realizaciones de los casos de uso definidos en el capítulo anterior, mediante diagramas de clases del diseño y diagramas de interacción. Aparecen además descritos los principios de diseño que se siguen, referentes a estándares de interfaz, tratamiento de excepciones y estándares de codificación. El diagrama de clases persistentes, así como el modelo de datos obtenido a partir de éste y el diagrama de despliegue donde se representan los nodos en los que se distribuye la aplicación, son otros artefactos del diseño que se presentan en este capítulo. Se muestra también el modelo de implementación con los diagramas de componentes definidos y una descripción de cada uno de estos.

### 4.2 Modelo de Diseño.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar, constituyendo una entrada principal en la actividad de implementación.

En este modelo, los casos de uso son realizados por las clases del diseño y sus objetos, lo cual se denota por la realización de casos de uso del diseño que describe cómo se realizan estos en particular. A continuación se muestran los diagramas de clases del diseño y diagramas de interacción para la realización de los casos de usos descritos en el capítulo anterior.

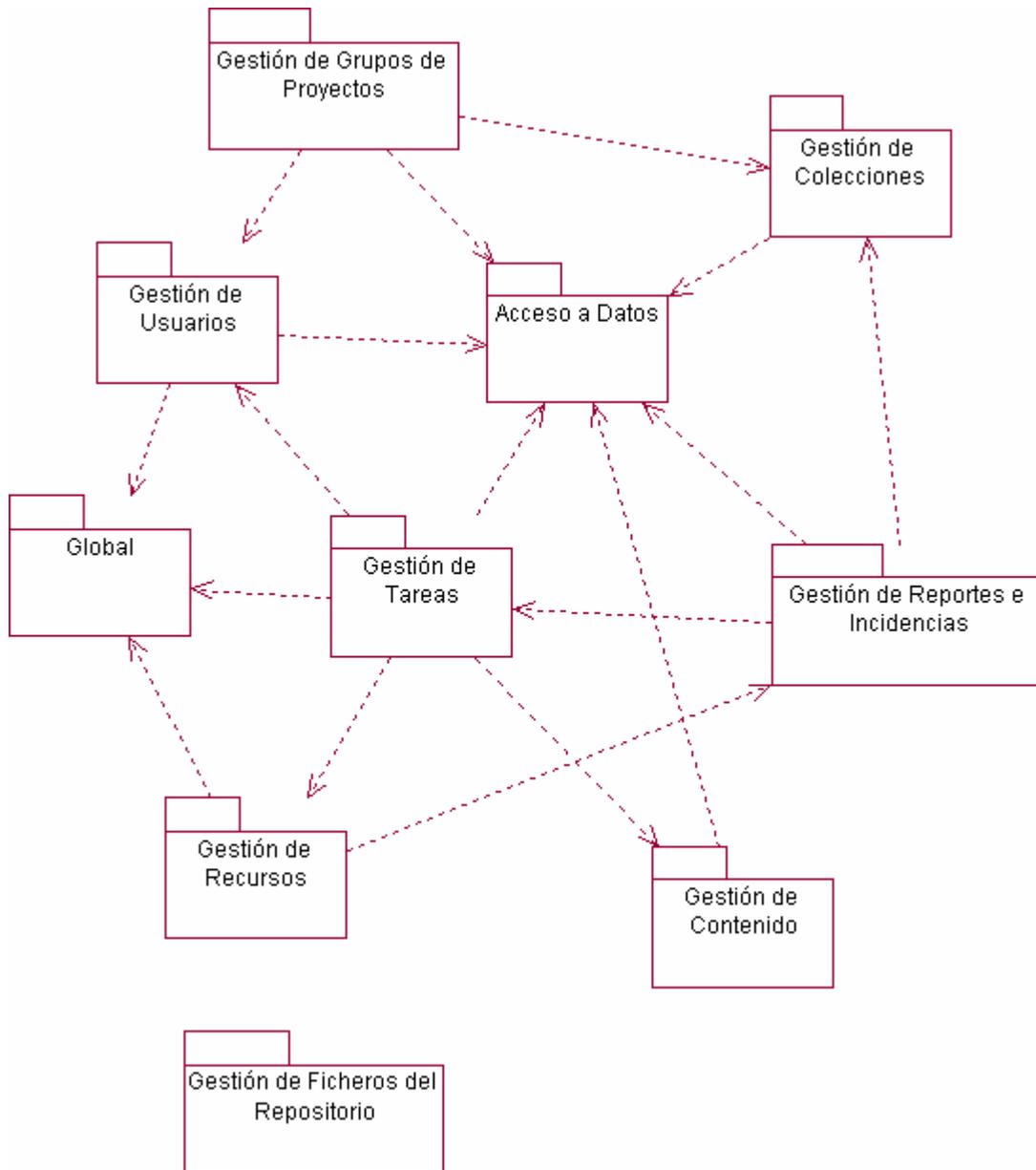


Fig. 4-1. Diagrama de paquetes del sistema

4.2.1 Paquete <Acceso a Datos>

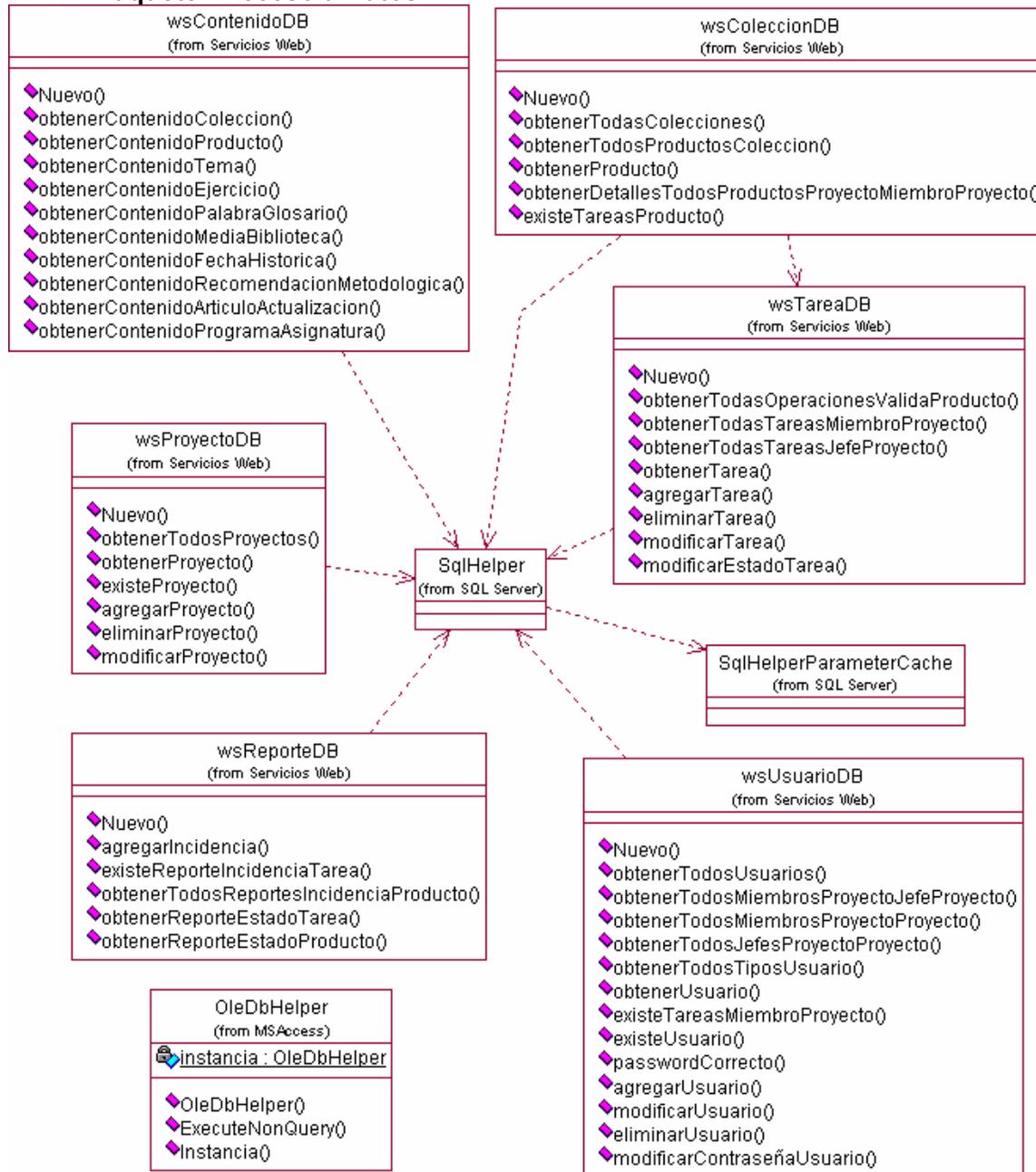


Fig. 4-2. Diagrama de clases del paquete Acceso a Datos

4.2.2 Paquete <Gestión de Colecciones>

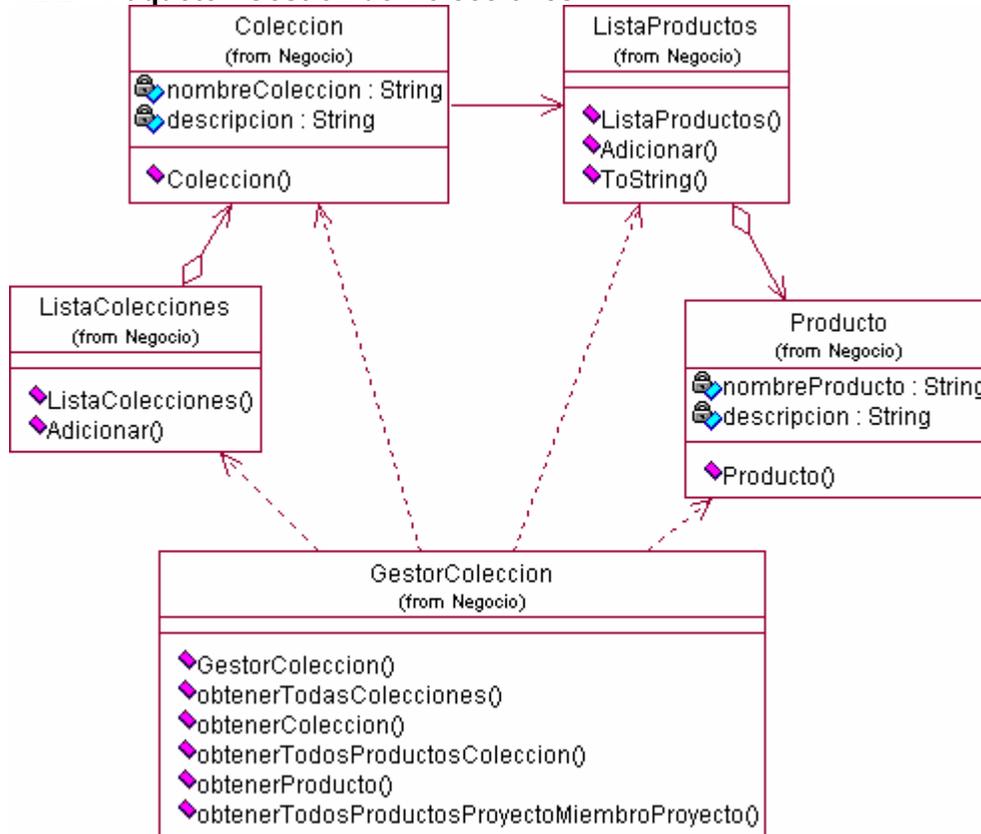


Fig. 4-3. Diagrama de clases del paquete Gestión de Colecciones

4.2.3 Paquete <Gestión de Ficheros del Repositorio>

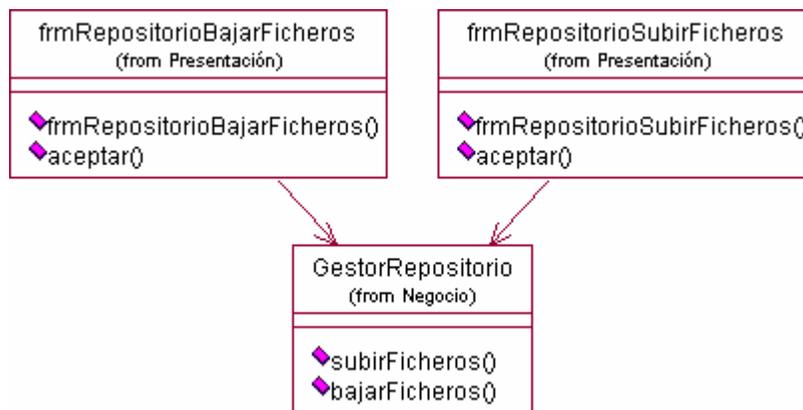


Fig. 4-4. Diagrama de clases del paquete Gestión de Ficheros del Repositorio

4.2.4 Paquete <Gestión de Grupos de Proyecto>

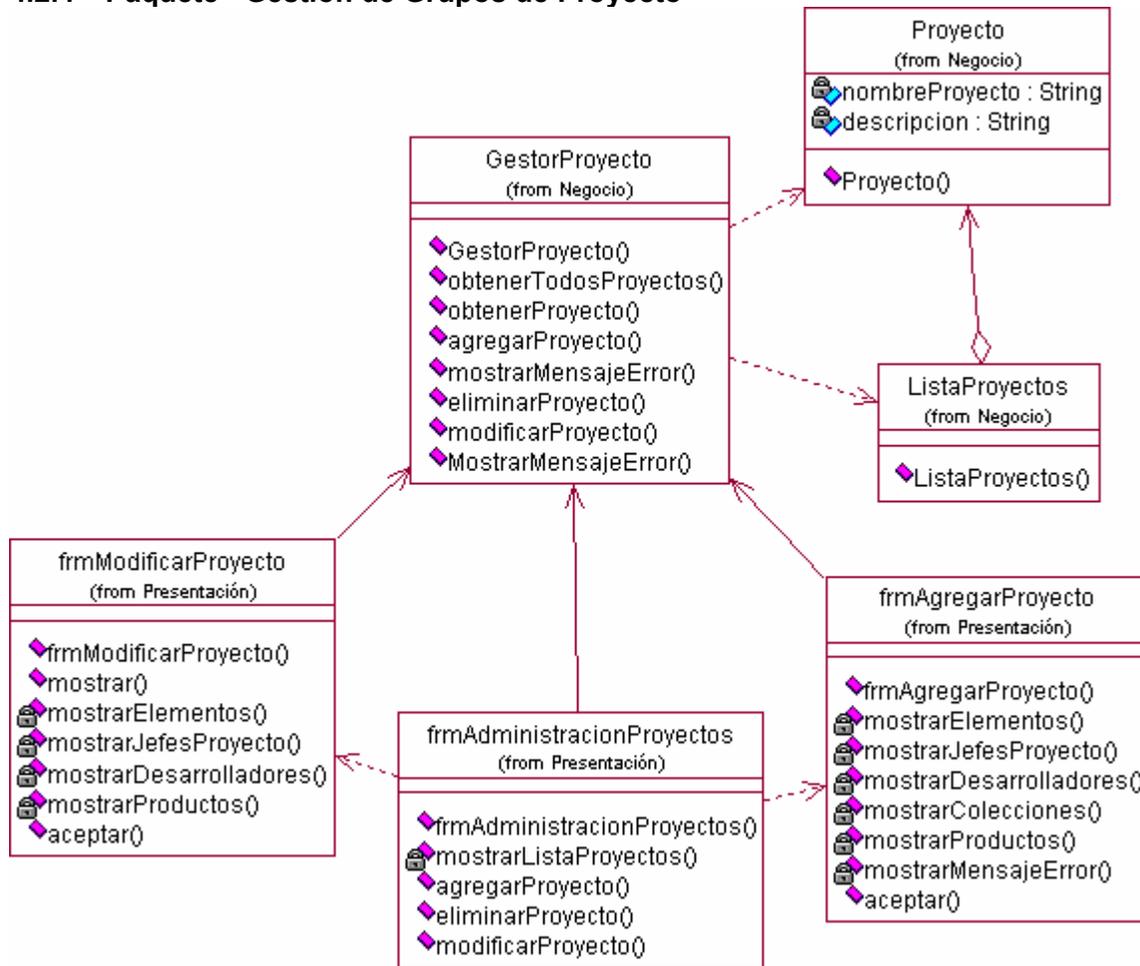


Fig. 4-5. Diagrama de clases del paquete Gestión de Grupos de Proyecto

#### 4.2.5 Paquete <Gestión de Contenido>

Para una mejor comprensión del diagrama se hizo una agrupación por paquetes, definiendo un paquete por cada módulo donde se incorpora contenido.

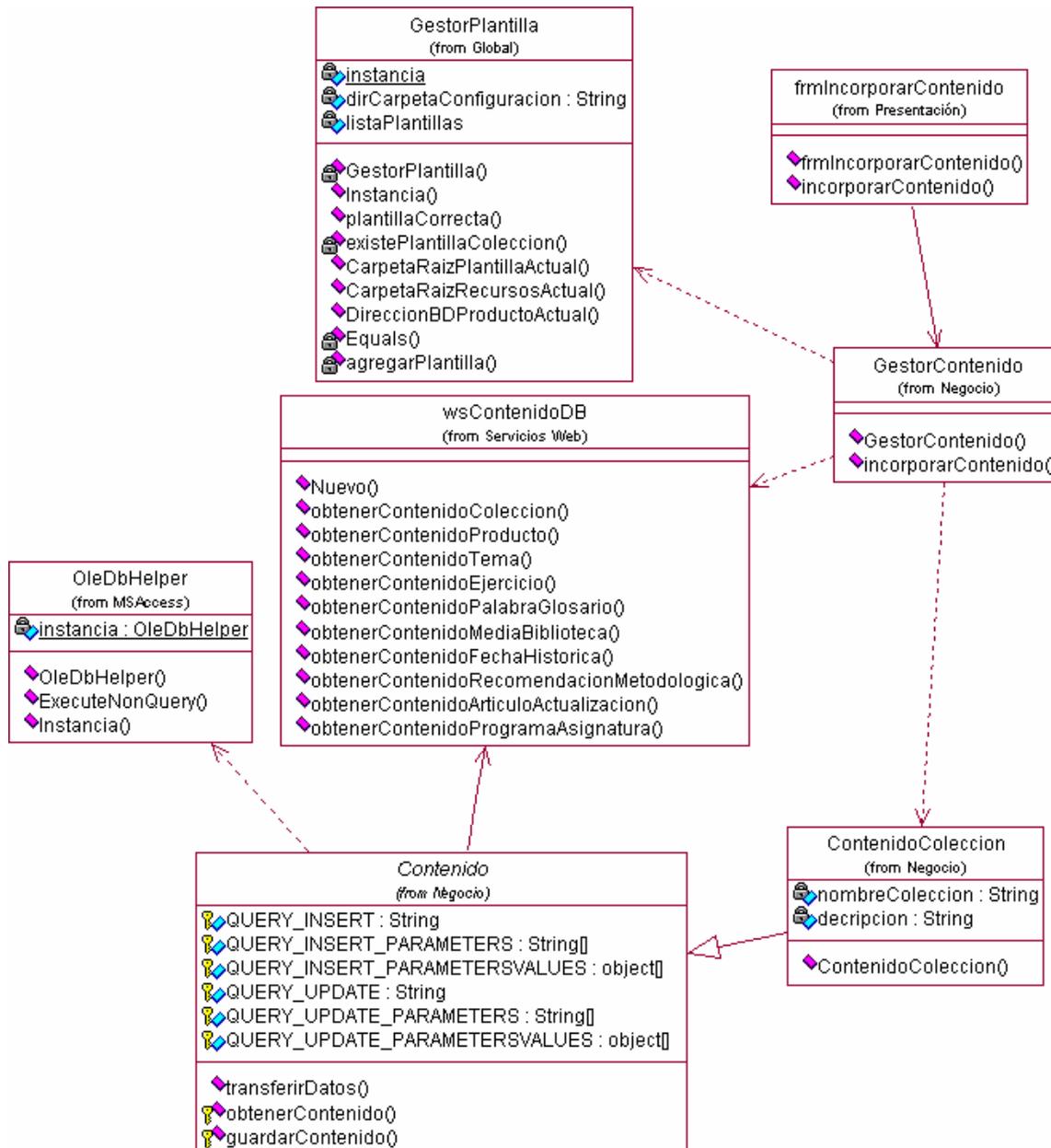


Fig. 4-6. Diagrama de clases del paquete Gestión de Contenido Sección 1

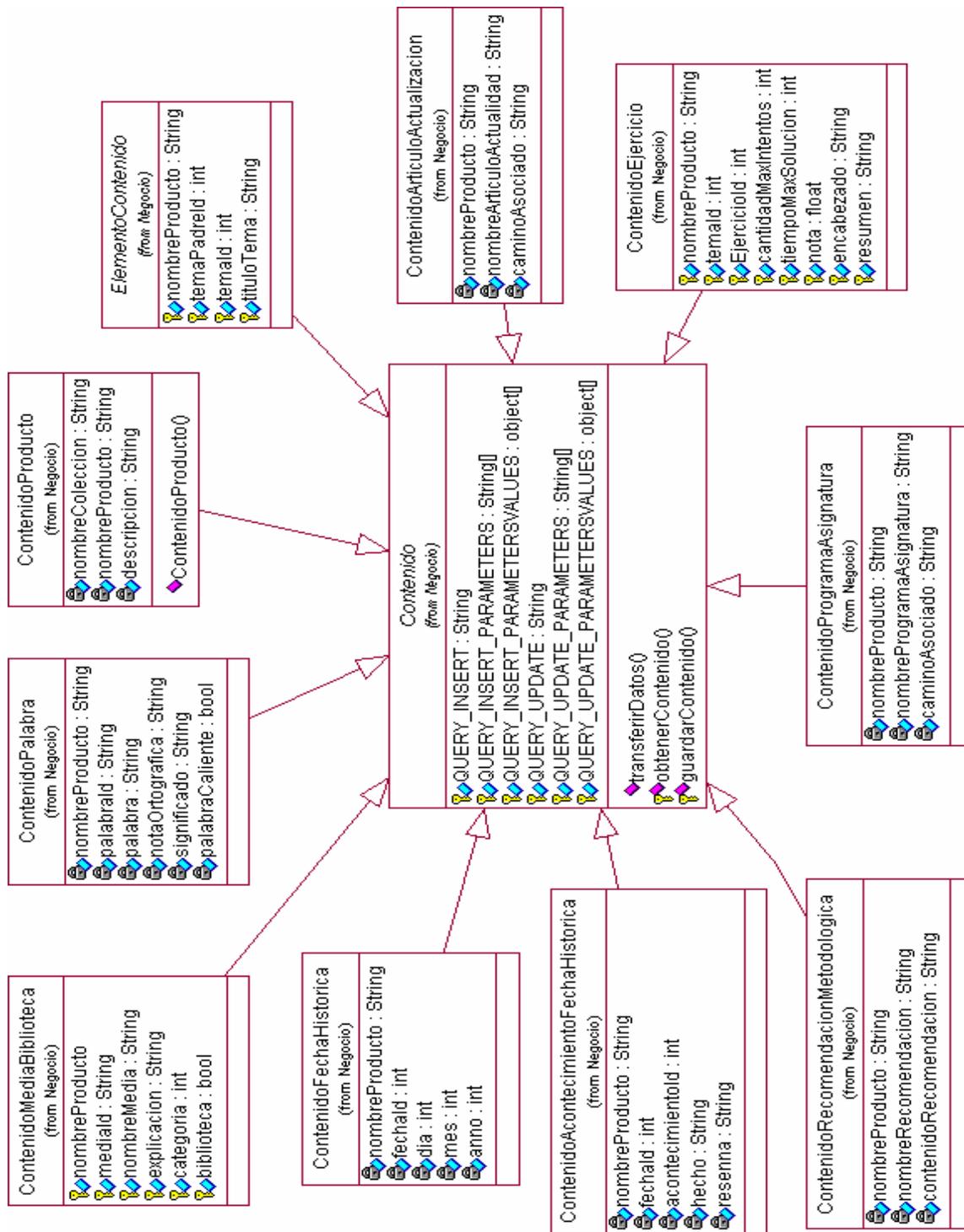


Fig. 4-7. Diagrama de clases del paquete Gestión de Contenido Sección 2

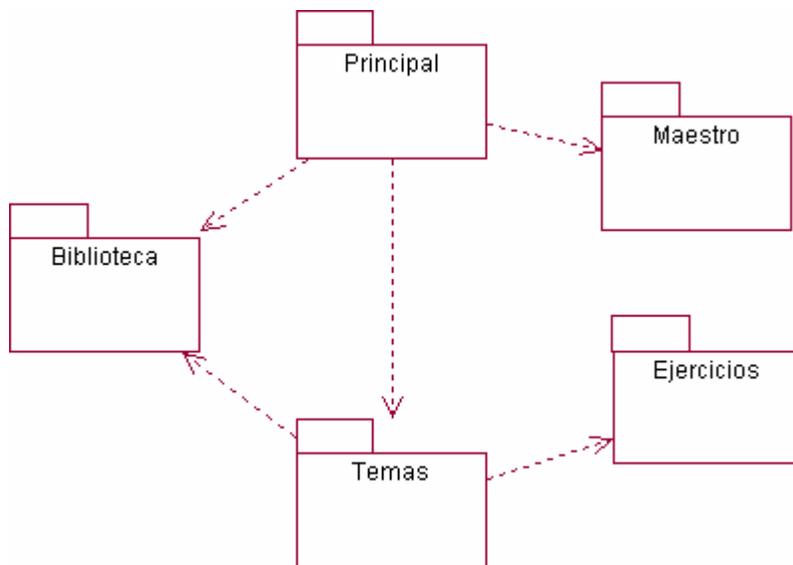


Fig. 4-8. Diagrama de paquetes del paquete Gestión de Contenido

#### 4.2.5.1 Paquete Gestión de Contenidos <Temas>

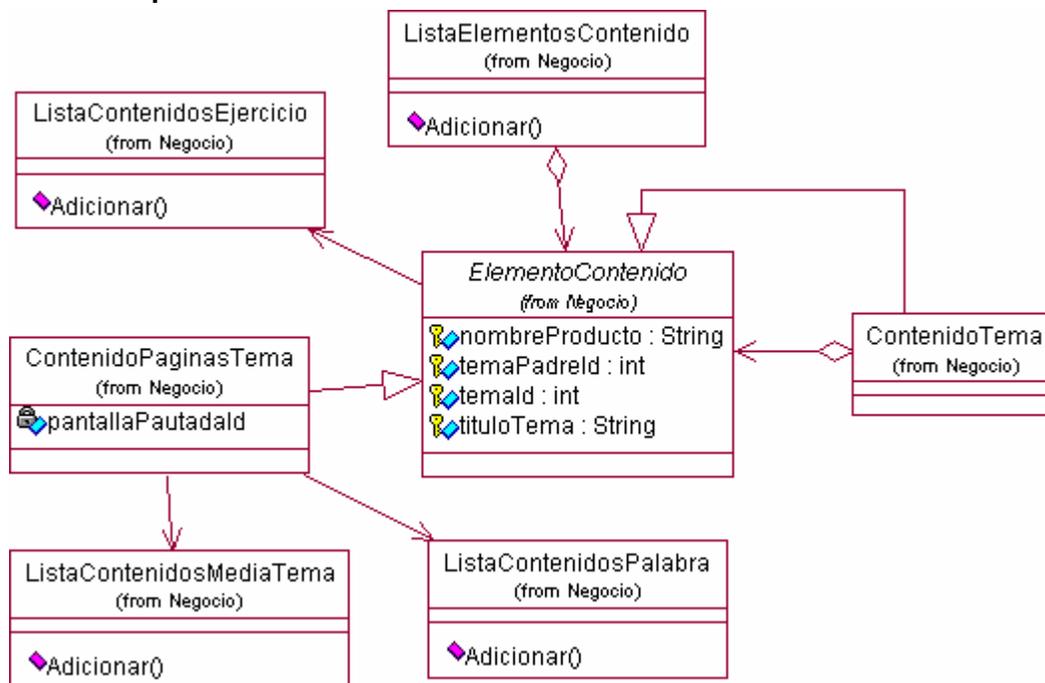


Fig. 4-9. Diagrama de clases del paquete Gestión de Contenido Temas

4.2.5.2 Paquete Gestión de Contenidos <Principal>

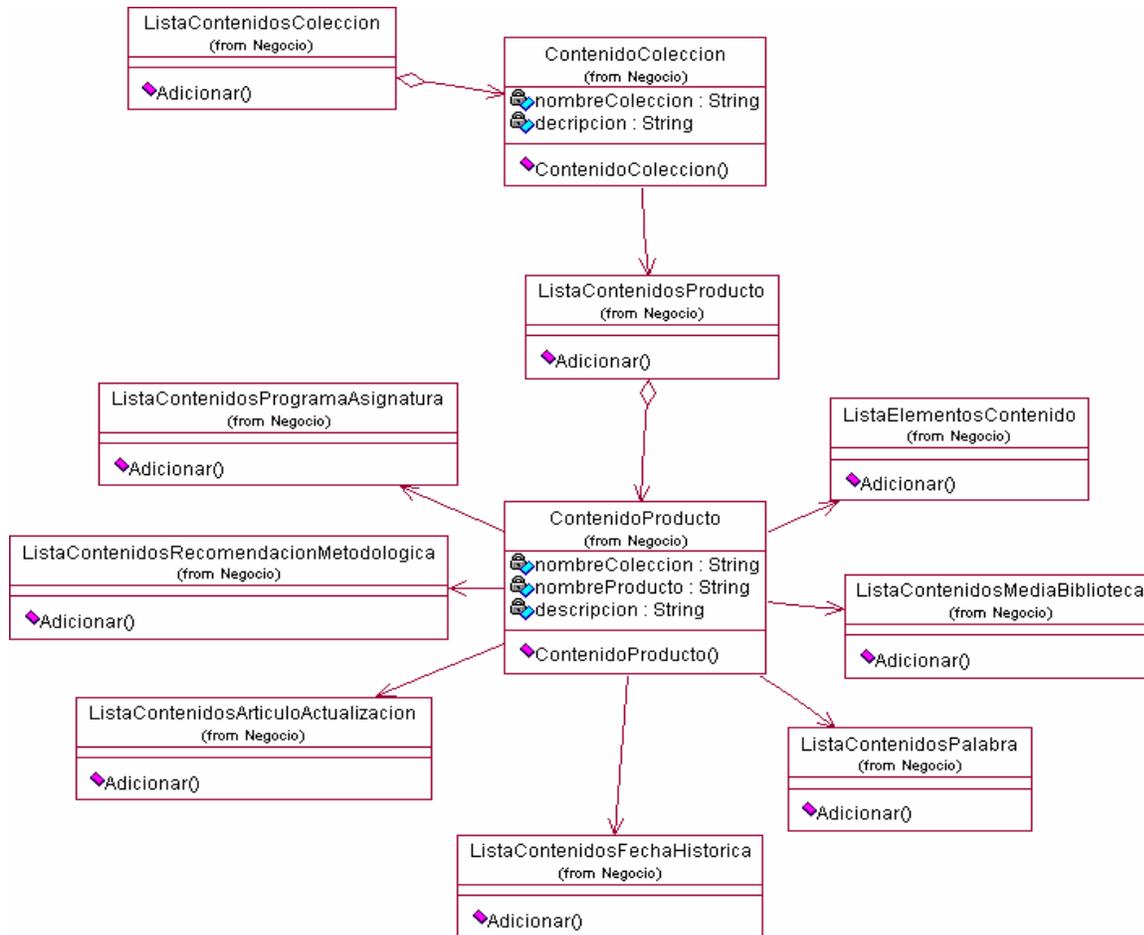


Fig. 4-10. Diagrama de clases del paquete Gestión de Contenido Principal

4.2.5.3 Paquete Gestión de Contenidos <Maestro>

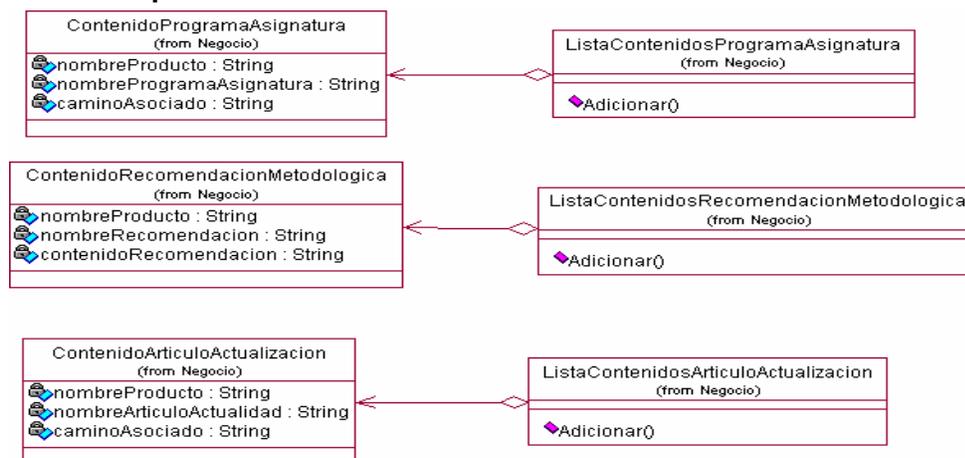


Fig. 4-11. Diagrama de clases del paquete Gestión de Contenido Maestro

4.2.5.4 Paquete Gestión de Contenidos <Ejercicios>

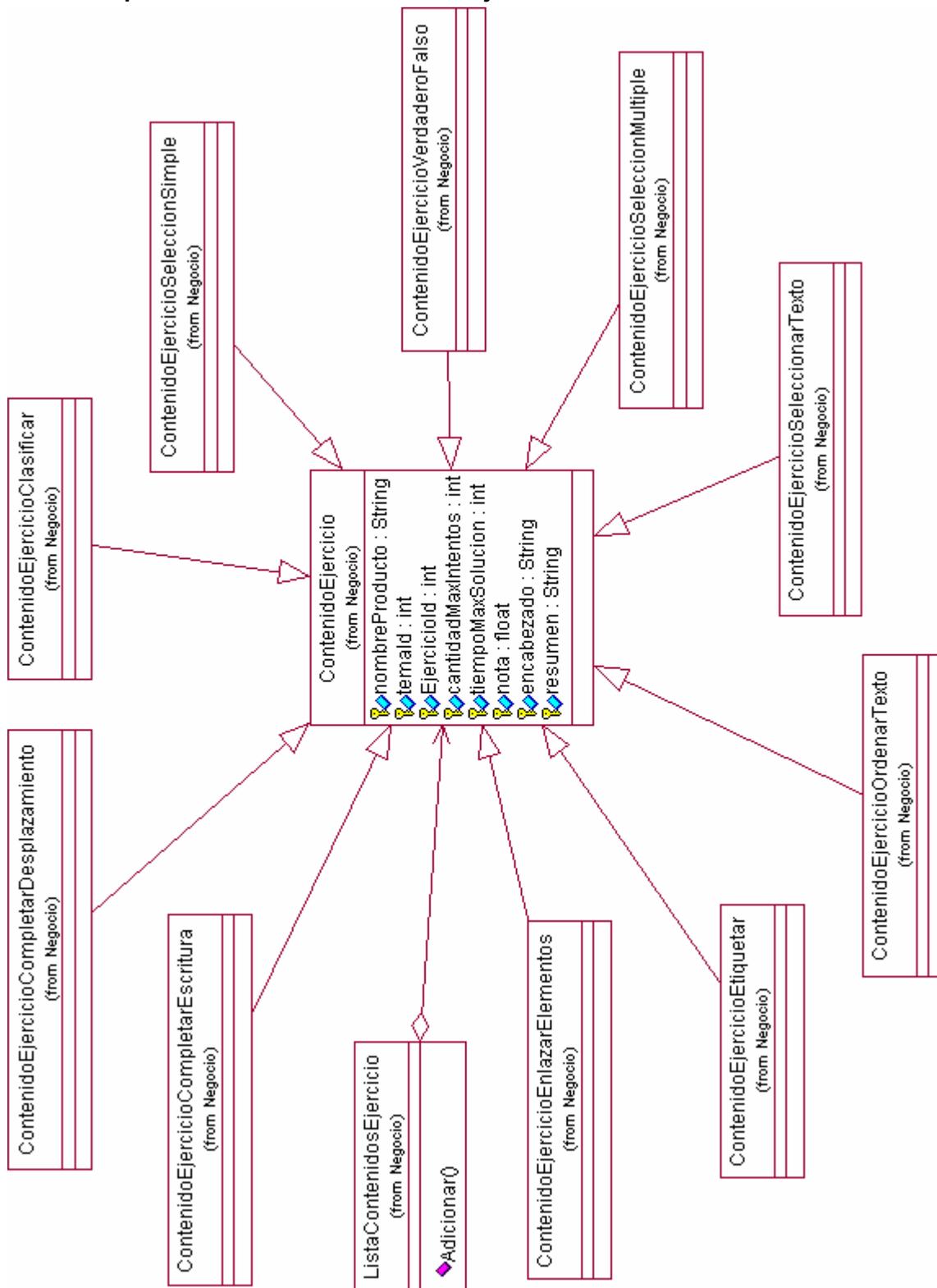


Fig. 4-12. Diagrama de clases del paquete Gestión de Contenido Ejercicios

4.2.5.5 Paquete Gestión de Contenidos <Biblioteca>

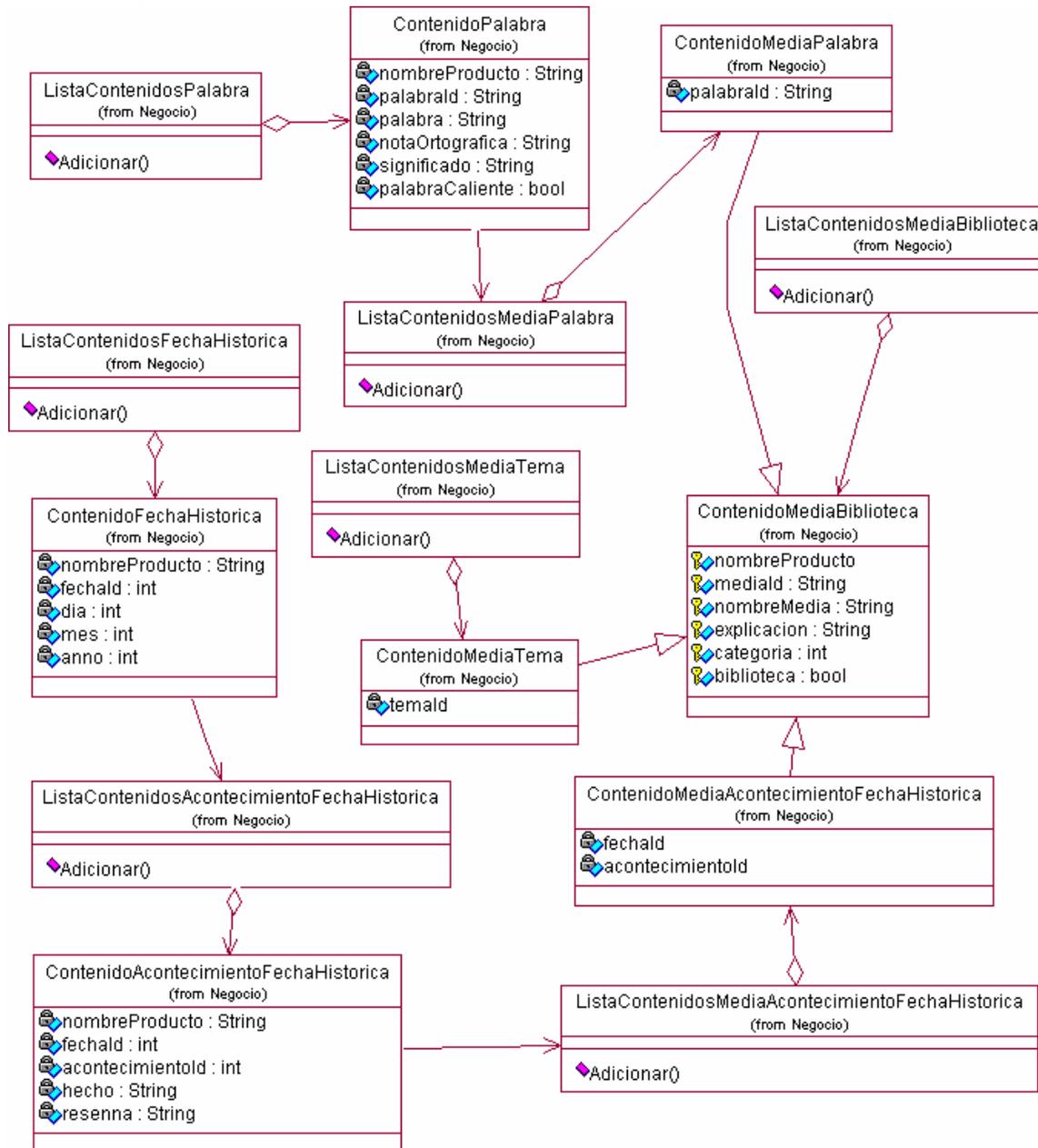


Fig. 4-13. Diagrama de clases del paquete Gestión de Contenido Biblioteca

4.2.6 Paquete <Gestión de Recursos>

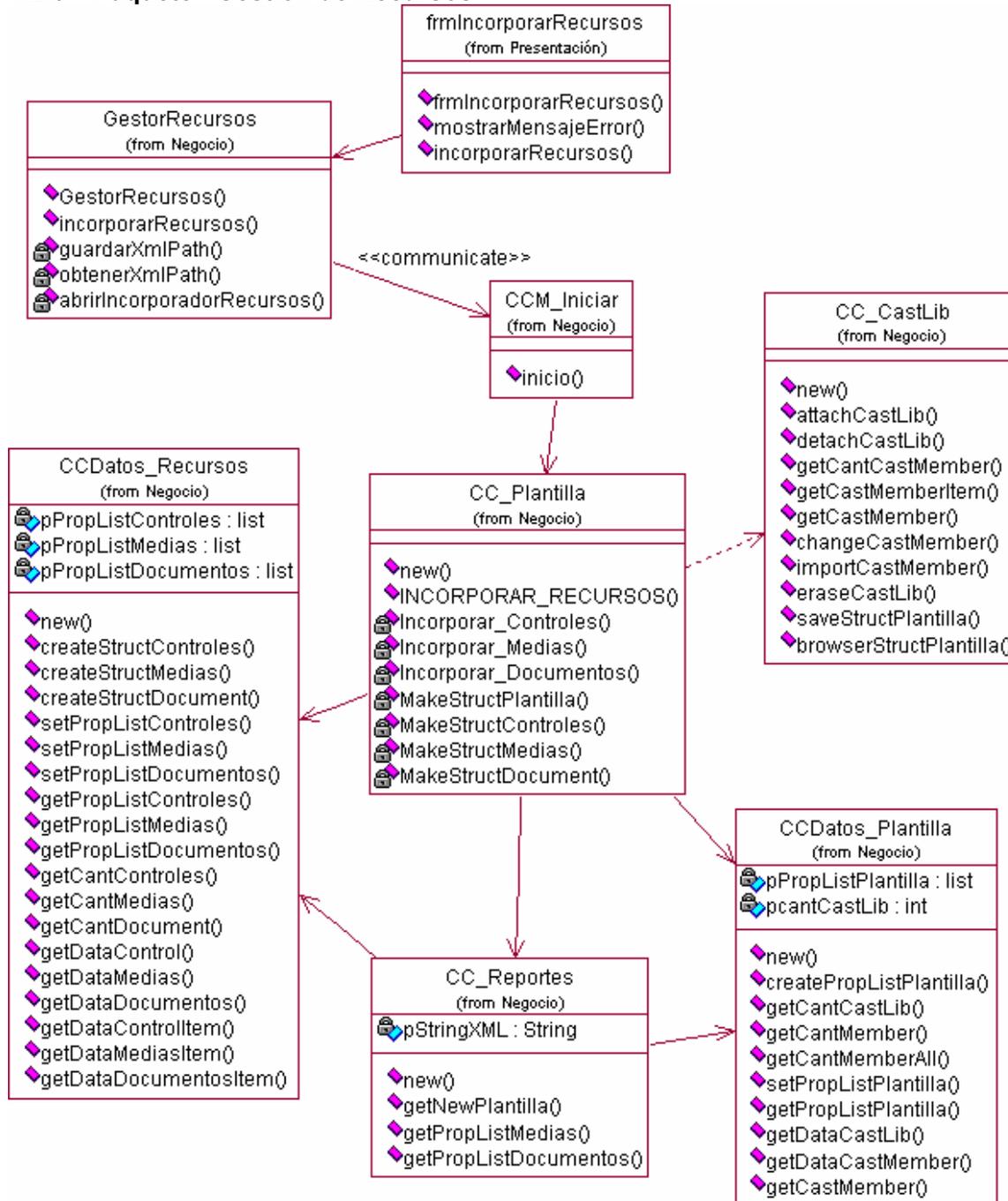


Fig. 4-14. Diagrama de clases del paquete Gestión de Recursos

4.2.7 Paquete <Gestión de Reportes e Incidencias>

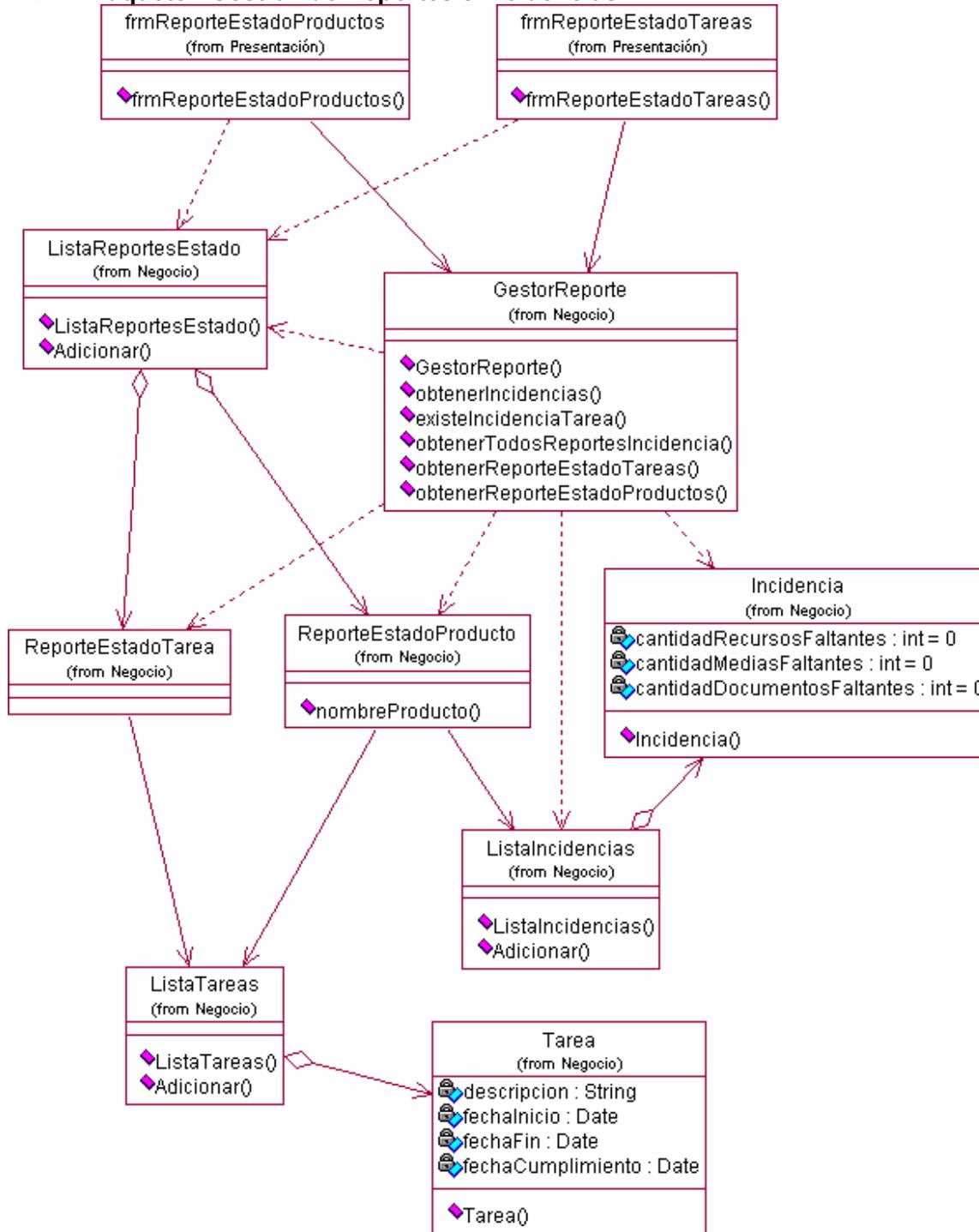


Fig. 4-15. Diagrama de clases del paquete Gestión de Reportes e Incidencias

4.2.8 Paquete <Gestión de Tareas>

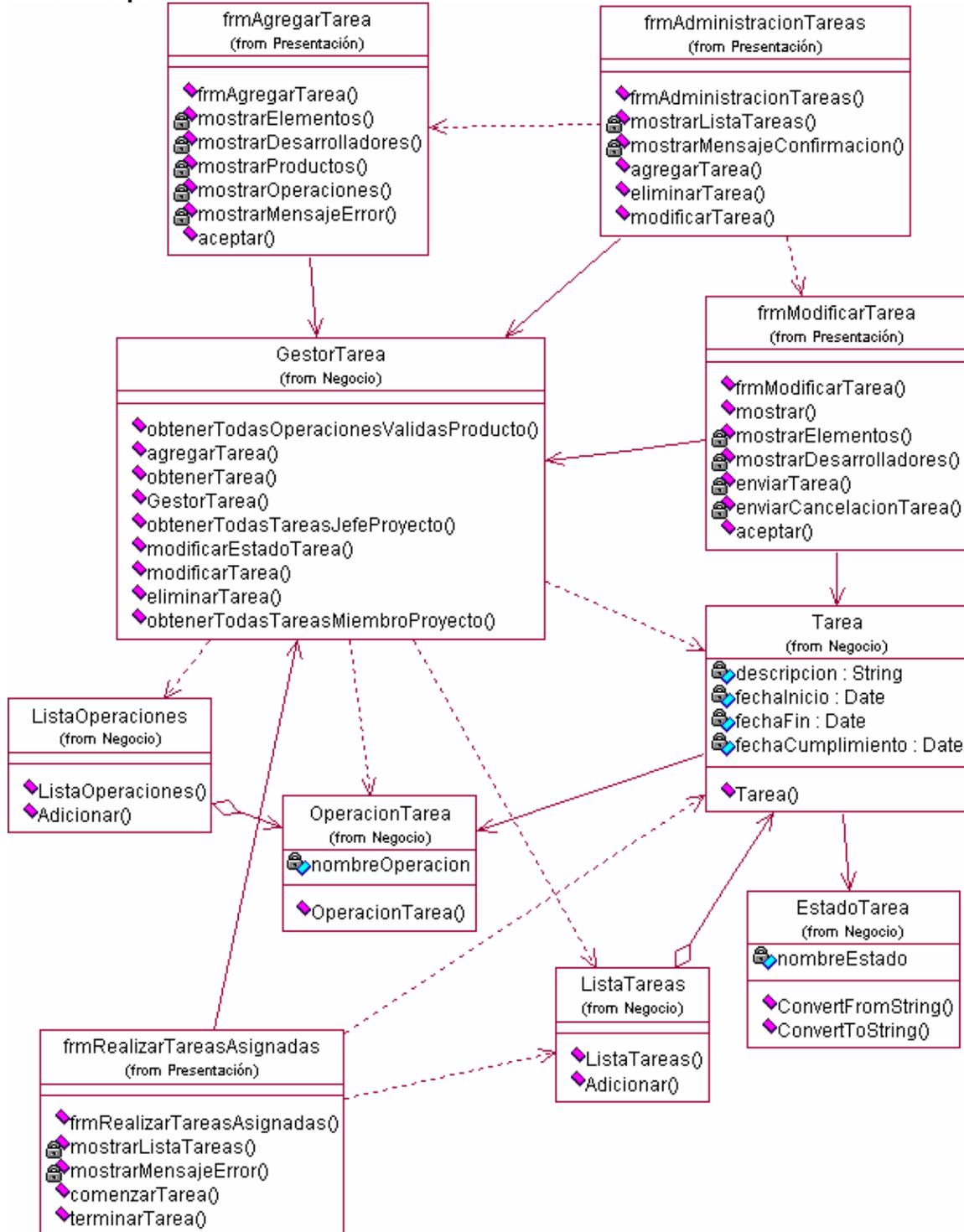


Fig. 4-16. Diagrama de clases del paquete Gestión de Tareas

4.2.9 Paquete <Gestión de Usuarios>

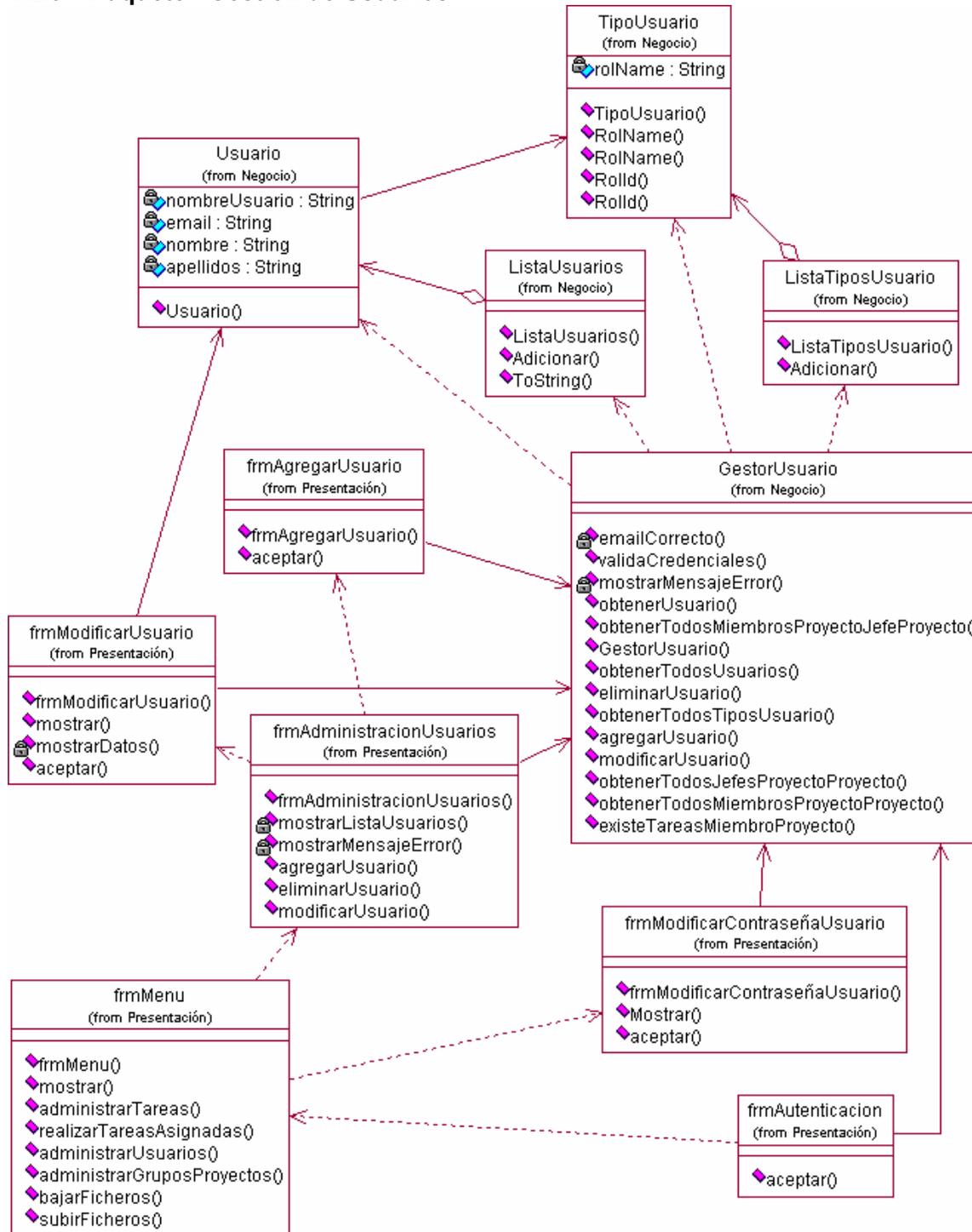


Fig. 4-17. Diagrama de clases del paquete Gestión de Usuarios

4.2.10 Paquete <Global>

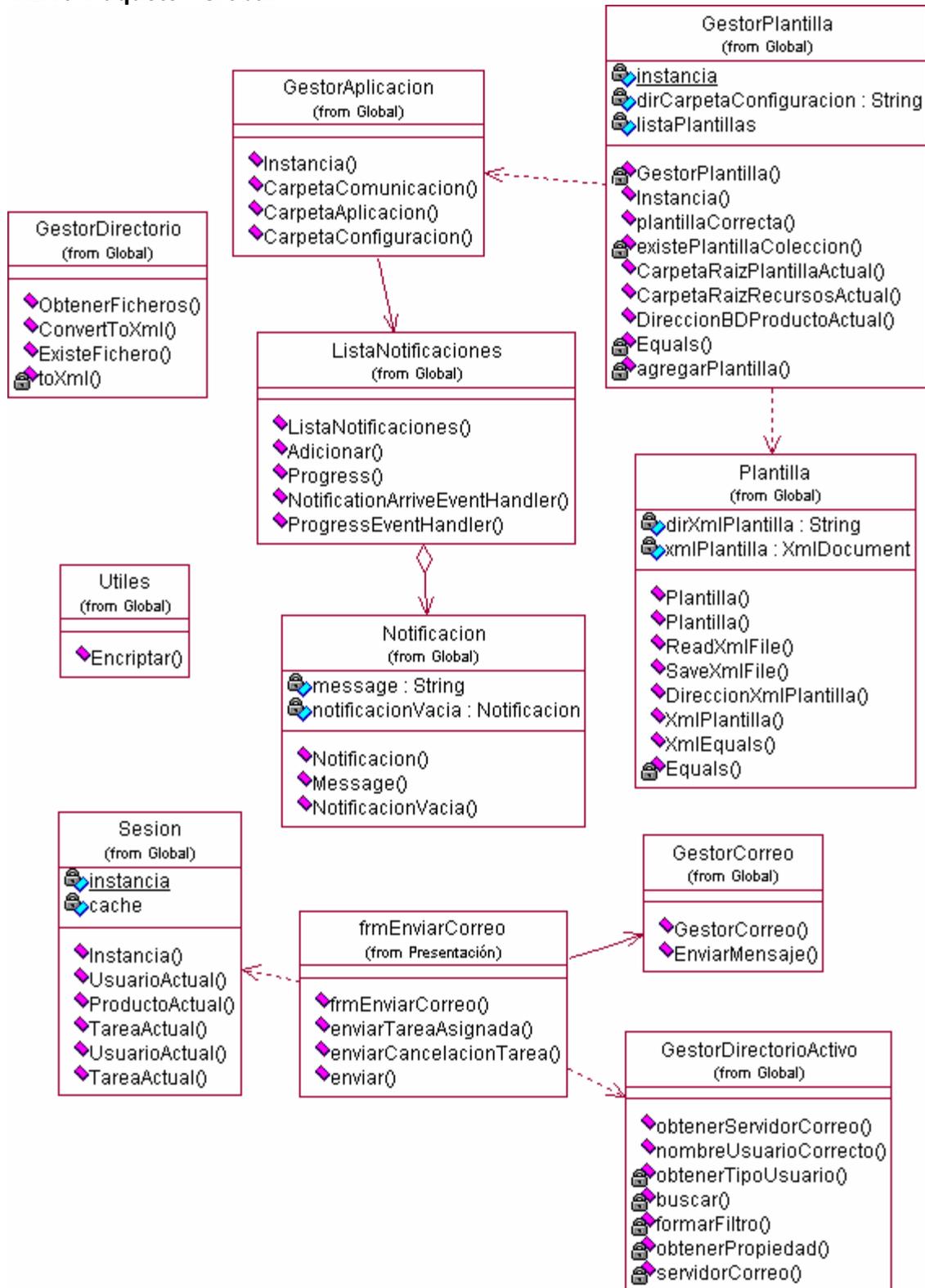


Fig. 4-18. Diagrama de clases del paquete Global

4.2.11 Descripción de las principales clases del diseño.

**Tabla 4-1. Descripción de la clase OleDbHelper**

<b>Nombre:</b> OleDbHelper	
<b>Tipo de clase:</b> Acceso a datos	
<b>Atributo</b>	<b>Tipo</b>
instancia	OleDbHelper
cmd	OleDbCommand
<b>Para cada responsabilidad:</b>	
Nombre:	<i>Instancia()</i>
Descripción:	Devuelve una única instancia de la clase.
Nombre:	<i>ExecuteNonQuery(query, nombreParametros, valoresParametros)</i>
Descripción:	Ejecuta una consulta en la base de datos.

**Tabla 4-2. Descripción de la clase SqlHelper**

<b>Nombre:</b> SqlHelper
<b>Tipo de clase:</b> Acceso a datos
<b>Descripción:</b> Clase encargada de realizar todas las consultas a la base de datos del sistema.

**Tabla 4-3. Descripción de la clase GestorColeccion**

<b>Nombre:</b> GestorColeccion	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
coleccionesDB	colecciondb (Servicio Web)
<b>Para cada responsabilidad:</b>	
Nombre:	<i>obtenerTodasColecciones()</i>
Descripción:	Devuelve una lista de todas las colecciones existentes.
Nombre:	<i>obtenerColeccion(nombreColeccion)</i>
Descripción:	Devuelve los datos de una colección.
Nombre:	<i>obtenerTodosProductosColeccion(nombreColeccion, quitarProductosConProyecto)</i>
Descripción:	Devuelve todos los productos existentes en una colección excepto, si se especifica, aquellos que ya están asignados a un proyecto.
Nombre:	<i>obtenerProducto(nombreColeccion, nombreProducto)</i>
Descripción:	Devuelve los datos de un producto de una colección.

## CAPÍTULO 4. CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

Nombre:	<i>obtenerTodosProductosProyectoMiembroProyecto(nombreMiembroProyecto)</i>
Descripción:	Devuelve todos los productos asignados al proyecto al que pertenece un desarrollador o jefe de proyecto dado.

**Tabla 4-4. Descripción de la clase GestorProyecto**

<b>Nombre:</b> GestorProyecto	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
proyectosDB	proyectodb (Servicio Web)
<b>Para cada responsabilidad:</b>	
Nombre:	<i>obtenerTodosProyectos()</i>
Descripción:	Devuelve una lista de todos los proyectos existentes.
Nombre:	<i>obtenerProyecto(nombreProyecto)</i>
Descripción:	Devuelve los datos de un proyecto.
Nombre:	<i>agregarProyecto(nombreProyecto, descripción, nombreJefeProyecto, miembros, nombreColeccion, productos)</i>
Descripción:	Crea un nuevo proyecto.
Nombre:	<i>eliminarProyecto(nombreProyecto)</i>
Descripción:	Elimina un proyecto.
Nombre:	<i>modificarProyecto(nombreProyecto, descripcion, nombreJefeProyecto, miembros, nombreColeccion, productos)</i>
Descripción:	Modifica los datos de un proyecto.

**Tabla 4-5. Descripción de la clase GestorUsuario**

<b>Nombre:</b> GestorUsuario	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
usuariosDB	usuariodb (Servicio Web)
<b>Para cada responsabilidad:</b>	
Nombre:	<i>emailCorrecto(email)</i>
Descripción:	Verifica si el correo electrónico es o no correcto.
Nombre:	<i>verificaCredenciales(nombreUsuario, password)</i>
Descripción:	Verifica si las credenciales del usuario que va a usar el sistema son o no correctas.

## CAPÍTULO 4. CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

Nombre:	<i>obtenerTodosUsuarios()</i>
Descripción:	Crea un nuevo proyecto.
Nombre:	<i>obtenerTodosMiembrosProyectoJefeProyecto(nombreUsuario)</i>
Descripción:	Devuelve los datos de los miembros del proyecto que dirige un Jefe de Proyecto dado.
Nombre:	<i>obtenerTodosJefesProyectoProyecto(nombreProyecto)</i>
Descripción:	Devuelve los datos del Jefe de Proyecto del proyecto dado. Si el valor es nulo devuelve los datos de todos los Jefes de Proyectos que no estén dirigiendo un proyecto.
Nombre:	<i>obtenerTodosMiembrosProyectoProyecto(nombreProyecto)</i>
Descripción:	Devuelve los datos de los desarrolladores que pertenecen al proyecto dado. Si el valor es nulo devuelve los datos de todos los desarrolladores que no estén en un proyecto.
Nombre:	<i>obtenerUsuario(nombreUsuario)</i>
Descripción:	Devuelve todos los datos de un usuario dado.
Nombre:	<i>agregarUsuario(nombreUsuario, password, rolld, email, nombre, apellidos)</i>
Descripción:	Crea un usuario nuevo.
Nombre:	<i>eliminarUsuario(nombreUsuario)</i>
Descripción:	Elimina un usuario dado.
Nombre:	<i>modificarUsuario(nombreUsuario, nuevoNombreUsuario, password, rolld, email, nombre, apellidos)</i>
Descripción:	Modifica los datos de un usuario.
Nombre:	<i>obtenerTodosTiposUsuario()</i>
Descripción:	Devuelve todos los tipos de usuario existentes.
Nombre:	<i>existeTareasMiembroProyecto(nombreMiembroProyecto)</i>
Descripción:	Verifica si un desarrollador tiene o no tareas asignadas.

**Tabla 4-6. Descripción de la clase GestorTarea**

<b>Nombre:</b> GestorTarea	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
tareasDB	tareadb (Servicio Web)
<b>Para cada responsabilidad:</b>	

## CAPÍTULO 4. CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

Nombre:	<i>obtenerTodasOperacionesValidasProducto (nombreProducto)</i>
Descripción:	Devuelve los datos de todas las operaciones válidas que se pueden hacer sobre un producto.
Nombre:	<i>obtenerTodasTareasJefeProyecto (nombreJefeProyecto, seleccionarEnHistorico)</i>
Descripción:	Devuelve los datos de todas las tareas que ha asignado un jefe de proyecto y selecciona también, si se especifica, aquellas que ya eliminó antes en estado "Terminada"
Nombre:	<i>obtenerTodasTareasMiembroProyecto (nombreUsuario)</i>
Descripción:	Devuelve los datos de todas las tareas que han sido asignadas a un desarrollador.
Nombre:	<i>obtenerTarea (tareald)</i>
Descripción:	Devuelve los datos de una tarea.
Nombre:	<i>agregarTarea (operacionId, descripcion, fechaInicio, fechaFin, nombreUsuario, nombreProducto)</i>
Descripción:	Crea una nueva tarea. Esta tarea es asignada a un usuario. Devuelve el identificador de la tarea.
Nombre:	<i>eliminarTarea (tareald, ponerlaEnHistorico)</i>
Descripción:	Elimina una tarea y la pone en el histórico de tareas si se especifica.
Nombre:	<i>modificarEstadoTarea (tareald, estadold)</i>
Descripción:	Modifica el estado de una tarea.
Nombre:	<i>modificarTarea (descripcion, fechaFin, nombreUsuario)</i>
Descripción:	Modifica los datos de una tarea.

**Tabla 4-7. Descripción de la clase GestorReporte**

<b>Nombre:</b> GestorReporte	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
reportesDB	reportedb (Servicio Web)
dirCarpetaComunicacion	string
<b>Para cada responsabilidad:</b>	
Nombre:	<i>obtenerIncidencias ()</i>
Descripción:	Obtiene las incidencias que hubo durante la realización de una

	tarea de incorporación de recursos y guarda los datos.
Nombre:	<i>existenIncidenciaTarea (tareaId)</i>
Descripción:	Verifica si existieron o no incidencias durante la realización de una tarea de incorporación de recursos.
Nombre:	<i>obtenerTodosReportesIncidencia (nombreProducto)</i>
Descripción:	Devuelve los datos de todas las incidencias que ha tenido un producto durante la realización de las tareas de incorporación de recursos.
Nombre:	<i>obtenerReporteEstadoTareas (nombreUsuario)</i>
Descripción:	Devuelve el reporte del estado de todas las tareas que han sido asignadas a desarrolladores del proyecto al que pertenece.
Nombre:	<i>obtenerReporteEstadoProductos (nombreUsuario)</i>
Descripción:	Devuelve el reporte del estado de todos los productos que atienden en el proyecto al que pertenece.

**Tabla 4-8. Descripción de la clase GestorRecursos**

<b>Nombre:</b> GestorRecursos	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
dirCarpetaComunicacion	string
dirCarpetaAplicacion	string
<b>Para cada responsabilidad:</b>	
Nombre:	<i>incorporarRecursos (versionComenzada)</i>
Descripción:	Controla el proceso de incorporación de recursos. En este método se hacen llamadas a los otros métodos de esta clase.
Nombre:	<i>guardarXmlPath (versionComenzada)</i>
Descripción:	Guarda un fichero XML con los datos necesarios para incorporar recursos en la plantilla.
Nombre:	<i>obtenerXmlPath (versionComenzada)</i>
Descripción:	Obtiene el contenido que tendrá el XML que se guarda en el método <i>guardarXmlPath</i>
Nombre:	<i>abrirIncorporadorRecursos ()</i>
Descripción	Ejecuta la aplicación encargada de incorporar los recursos a la plantilla. Esta aplicación usa el XML que se guardó en el método <i>guardarXmlPath</i>

**Tabla 4-9. Descripción de la clase GestorContenido**

<b>Nombre:</b> GestorContenido	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
contenidosDB	contenidodb (Servicio Web)
<b>Para cada responsabilidad:</b>	
Nombre:	<i>incorporarContenido ()</i>
Descripción:	Controla el proceso de incorporación de contenido.

**Tabla 4-10. Descripción de la clase Contenido**

<b>Nombre:</b> Contenido	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
QUERY_INSERT	string
QUERY_INSERT_PARAMETERS	string
QUERY_INSERT_PARAMETERSVALUES	object[]
QUERY_UPDATE	string
QUERY_UPDATE_PARAMETERS	string
QUERY_UPDATE_PARAMETERSVALUES	object[]
<b>Para cada responsabilidad:</b>	
Nombre:	<i>transferirDatos()</i>
Descripción:	Método encargado de llevar el control de la incorporación de contenido. Cumple con el patrón Método de Plantilla.
Nombre:	<i>obtenerContenido()</i>
Descripción:	Método abstracto que implementarán las clases que hereden de esta clase para obtener el contenido de la base de datos de gestión.
Nombre:	<i>guardarContenido()</i>
Descripción:	Método abstracto que implementarán las clases que hereden de esta clase para guardar el contenido en la base de datos del producto.

**Tabla 4-11. Descripción de la clase GestorRepositorio**

<b>Nombre:</b> GestorRepositorio	
<b>Tipo de clase:</b> Controladora	
<b>Para cada responsabilidad:</b>	

## CAPÍTULO 4. CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

Nombre:	<i>subirFicheros ()</i>
Descripción:	Método encargado de subir ficheros al repositorio.
Nombre:	<i>bajarFicheros()</i>
Descripción:	Método encargado de bajar ficheros del repositorio.

**Tabla 4-12. Descripción de la clase GestorPlantilla**

<b>Nombre:</b> GestorPlantilla	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
dirCarpetaConfiguracion	string
listaPlantillas	Hashtable
<b>Para cada responsabilidad:</b>	
Nombre:	<i>Instancia ()</i>
Descripción:	Devuelve una única instancia de la clase.
Nombre:	<i>plantillaCorrecta (nombreColeccion, xmlPlantilla)</i>
Descripción:	Verifica si la plantilla que se va a usar corresponde o no con la plantilla definida para la colección.
Nombre:	<i>existePlantillaColeccion (nombreColeccion)</i>
Descripción:	Verifica si existe o no la plantilla para la colección.
Nombre:	<i>CarpetaRaizPlantillaActual()</i>
Descripción	Método que devuelve la dirección de la carpeta raíz de la plantilla que se está usando actualmente para desarrollar un producto.
Nombre:	<i>CarpetaRaizRecursosActual()</i>
Descripción	Método que devuelve la dirección de la carpeta raíz donde se encuentran los recursos que se incorporarán a la plantilla para desarrollar un producto.
Nombre:	<i>DireccionBDProductoActual()</i>
Descripción	Método que devuelve la dirección de la base de datos del producto a la que se incorporará el contenido.
Nombre:	<i>agregarPlantilla(nombreColeccion, xmlPlantilla)</i>
Descripción	Método que guarda el XML con la estructura de la plantilla de una colección

**4.3 Diagramas de secuencia.**

**4.3.1 Diagramas de secuencia del CU “Realizar Tareas Asignadas”**

[Ver Anexo II Fig. 6-2, 6-3 y 6-4]

**4.3.2 Diagrama de secuencia del CU “Incorporar Contenido”**

[Ver Anexo II Fig. 6-5]

**4.3.3 Diagrama de secuencia del CU “Incorporar Recursos”**

[Ver Anexo II Fig. 6-6]

**4.3.4 Diagramas de secuencia del CU “Administrar Tareas”**

[Ver Anexo II Fig. 6-7, 6-8, 6-9, 6-10, 6-11 y 6-12]

**4.3.5 Diagramas de secuencia del CU “Administrar Usuarios”**

[Ver Anexo II Fig. 6-13, 6-14, 6-15, 6-16, 6-17 y 6-18]

**4.3.6 Diagramas de secuencia del CU “Mostrar Información de Reportes”**

[Ver Anexo II Fig. 6-19 y 6-20]

**4.3.7 Diagrama de secuencia del CU “Autenticar Usuario”**

[Ver Anexo II Fig. 6-21]

**4.3.8 Diagramas de secuencia del CU “Gestionar Ficheros del Repositorio”**

[Ver Anexo II Fig. 6-22 y 6-23]

**4.3.9 Diagramas de secuencia del CU “Administrar Grupos de Proyecto”**

[Ver Anexo II Fig. 6-24, 6-25, 6-26, 6-27, 6-28 y 6-29]

4.4 Diseño de la base de datos.

4.4.1 Diagrama de clases persistentes.

A continuación se muestra el diagrama de clases persistentes del sistema, donde aparecen todas las entidades que se manejan en él.

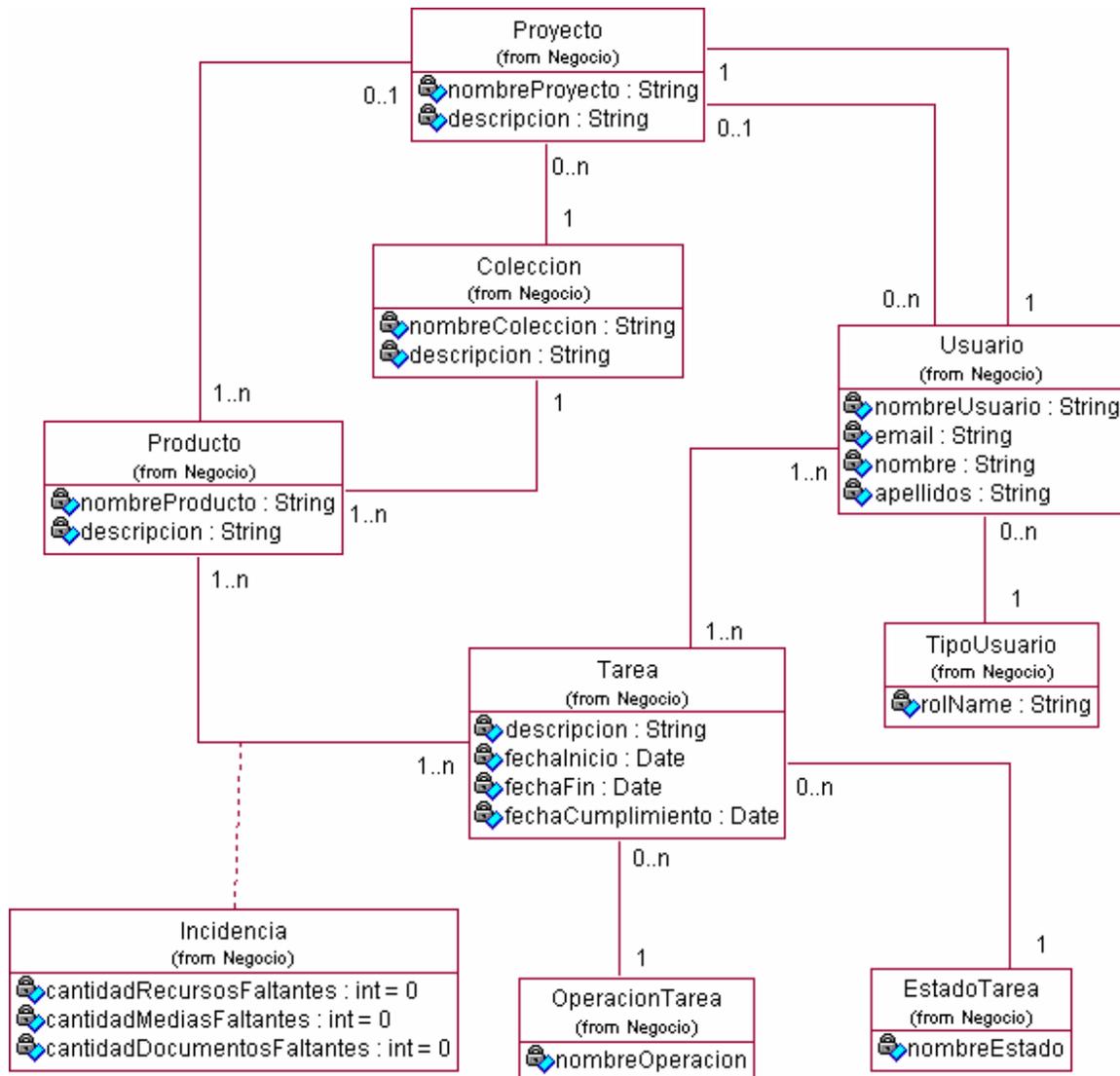


Fig. 4-19. Diagrama de clases persistentes

4.4.2 Modelo de datos.

El modelo de datos que se obtuvo a partir del diagrama de clases persistentes se muestra a continuación.

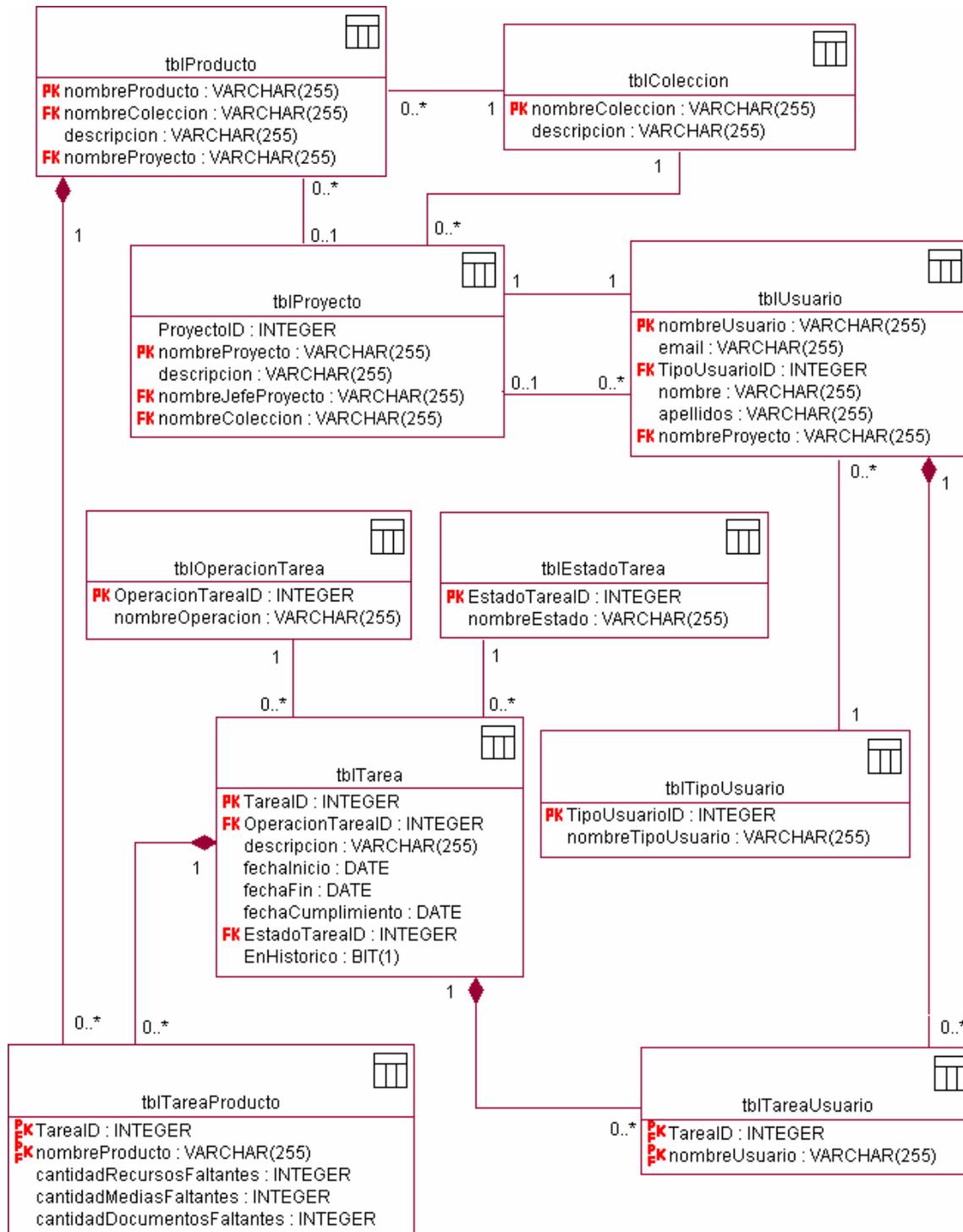


Fig. 4-20. Modelo de datos

4.4.3 Descripción de las tablas de la bases de datos.

**Tabla 4-13. Descripción de la tabla tblProyecto**

<b>Nombre: tblProyecto</b>		
<b>Descripción:</b> En esta tabla se almacenan los datos de los proyectos existentes.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
ProyectoID	int	Identificador autonumérico.
nombreProyecto	varchar	Nombre único que identifica a un proyecto.
descripcion	varchar	Descripción del proyecto.
nombreJefeProyecto	varchar	Nombre que identifica al Jefe de Proyecto.
nombreColeccion	varchar	Nombre que identifica a la colección en la que se está trabajando en el proyecto.

**Tabla 4-14. Descripción de la tabla tblColeccion**

<b>Nombre: tblColeccion</b>		
<b>Descripción:</b> En esta tabla se almacenan los datos de las colecciones existentes.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
nombreColeccion	varchar	Nombre único que identifica a una colección.
descripcion	varchar	Descripción de la colección.

**Tabla 4-15. Descripción de la tabla tblProducto**

<b>Nombre: tblProducto</b>		
<b>Descripción:</b> En esta tabla se almacenan los datos de los productos existentes.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
nombreProducto	varchar	Nombre única que identifica a un producto.
nombreColeccion	varchar	Nombre que identifica a la colección a la que pertenece el producto.
descripcion	varchar	Descripción del producto.
nombreProyecto	varchar	Nombre que identifica el proyecto que desarrollará el producto. Este valor puede ser nulo indicando que el producto no está asignado a un proyecto.

**Tabla 4-16. Descripción de la tabla tblUsuario**

<b>Nombre: tblUsuario</b>		
<b>Descripción:</b> En esta tabla se almacenan los datos de los usuarios del sistema existentes.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
nombreUsuario	varchar	Nombre único que identifica a un usuario.
email	varchar	Correo electrónico del usuario.
rol	int	Número que identifica el tipo de usuario.
nombre	varchar	Nombre del usuario.
apellidos	varchar	Apellidos del usuario.
nombreProyecto	varchar	Nombre que identifica al proyecto al que pertenece el usuario. Este valor puede ser nulo indicando que el usuario no pertenece a un proyecto. El administrador del sistema no pertenece a un proyecto.

**Tabla 4-17. Descripción de la tabla tblTipoUsuario**

<b>Nombre: tblTipoUsuario</b>		
<b>Descripción:</b> En esta tabla se almacenan los datos de los tipos de usuario que pueden existir en el sistema.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
TipoUsuarioID	int	Número único que identifica a un tipo de usuario.
nombreTipoUsuario	varchar	Nombre del tipo de usuario. Este valor es el correspondiente a “Administrador del sistema”, “Jefe de Proyecto” o “Desarrollador”.

**Tabla 4-18. Descripción de la tabla tblTarea**

<b>Nombre: tblTarea</b>		
<b>Descripción:</b> En esta tabla se almacenan los datos de las tareas que se le han asignado a los desarrolladores.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
TareaID	int	Número único que identifica a la tarea.
OperacionTareaID	int	Numero que identifica a la operación

		asociada a la tarea.
descripcion	varchar	Descripción de la tarea.
fechaInicio	date	Fecha en que debe comenzar a realizar la tarea.
fechaFin	date	Fecha en que debe terminar la tarea.
fechaCumplimiento	date	Fecha en que cumplió la tarea. Este valor puede ser nulo indicando que la tarea no se ha cumplido todavía.
EstadoTareaID	int	Número que identifica al estado en que se encuentra la tarea.
EnHistorico	bit	Valor que indica que la tarea está guardada en el histórico o no. Una tarea se guarda en el histórico cuando el Jefe de Proyecto la elimina estando en estado Terminada.

**Tabla 4-19. Descripción de la tabla tblOperacionTarea**

<b>Nombre: tblOperacionTarea</b>		
<b>Descripción:</b> En esta tabla se almacenan los datos de las operaciones que se pueden realizar sobre un producto.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
OperacionTareaID	int	Número único que identifica a una operación.
nombreOperacion	varchar	Nombre de la operación. Este valor es el correspondiente a "Incorporar Contenidos" o "Incorporar Recursos".

**Tabla 4-20. Descripción de la tabla tblEstadoTarea**

<b>Nombre: tblEstadoTarea</b>		
<b>Descripción:</b> En esta tabla se almacenan los datos de los estados en que puede estar una tarea.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
EstadoTareaID	int	Número único que identifica a un estado de la tarea.
nombreOperacion	varchar	Nombre del estado. Este valor es el correspondiente a "Por Comenzar", "En

		Desarrollo”, “Terminada”, “Reasignada”.
--	--	---

**Tabla 4-21. Descripción de la tabla tblTareaUsuario**

<b>Nombre: tblTareaUsuario</b>		
<b>Descripción:</b> En esta tabla se almacenan las tareas que han sido asignadas a cada usuario.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
TareaID	int	Número único que identifica a la tarea.
nombreUsuario	varchar	Nombre del usuario al que se le asignó la tarea identificada con el atributo TareaID.

**Tabla 4-22. Descripción de la tabla tblTareaProducto**

<b>Nombre: tblTareaProducto</b>		
<b>Descripción:</b> En esta tabla se almacenan las tareas asociadas a un producto determinado.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
TareaID	int	Número único que identifica a la tarea.
nombreProducto	varchar	Nombre del producto al que se le asoció la tarea identificada con el atributo TareaID.
cantidadRecursosFaltantes	int	Número que indica la cantidad de recursos que faltan por incorporar a un producto. Este número por defecto es cero. Solamente puede ser diferente de cero cuando se trate de una tarea cuya operación es “Incorporar Recursos”
cantidadMediasFaltantes	int	Número que indica la cantidad de medias que faltan por incorporar a un producto. Este número por defecto es cero. Solamente puede ser diferente de cero cuando se trate de una tarea cuya operación es “Incorporar Recursos”

cantidadDocumentosFaltantes	int	Número que indica la cantidad de documentos que faltan por incorporar a un producto. Este número por defecto es cero. Solamente puede ser diferente de cero cuando se trate de una tarea cuya operación es “Incorporar Recursos”
-----------------------------	-----	--

#### 4.5 Principios de diseño.

##### 4.5.1 Estándares en la interfaz de la aplicación.

La interfaz gráfica del usuario es el medio por el cual este interactúa con el sistema, por lo que esta debe ser lo más amigable posible y lograr que se sienta identificado con la misma.

Para el diseño de la interfaz del sistema se tuvo en cuenta que los elementos que se repiten en varias pantallas deben ser ubicados en el mismo lugar permitiéndole al usuario acostumbrarse al ambiente y que no esté desorientado. Cada pantalla sólo contiene la información necesaria para el usuario, son uniformes logrando un balance de los elementos que la componen y aparecen localizadas siempre en el centro del escritorio.

Cada pantalla está representada por un título acorde a su funcionalidad. Todos los botones están adornados con iconos que sirven de referencia a la función de los mismos, entre los que se puede destacar los siguientes:



**Fig. 4-21. Iconos usados en la interfaz de usuario**

##### 4.5.2 Formatos de reportes.

Los reportes se muestran en formato de tabla, con los registros ordenados según sea conveniente en cada caso de uso. En el caso de los reportes de estado de las tareas, los registros se ordenan según su pronto vencimiento de forma descendente.

##### 4.5.3 Tratamiento de excepciones.

El tratamiento de errores posibilita el buen funcionamiento de una aplicación dándole una mejor apariencia ante los clientes. En el sistema se controla la entrada de datos del usuario a

nivel de interfaz. Cuando se produce un error por la entrada incorrecta de un valor suministrado por el usuario se captura y se muestra un mensaje de error al usuario. Estos mensajes aparecen en otra ventana y en algunos casos brindan información detallada de la excepción producida. Para esto se utilizan las librerías de clases de .Net (*System.Exception*). El gestor de base de datos (SQL Server 2000) garantiza la integridad referencial.

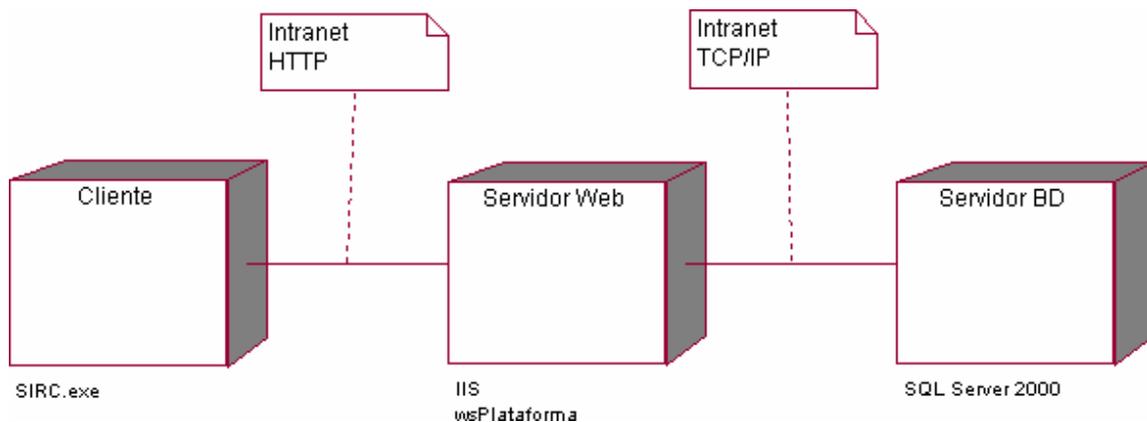
### 4.6 Estándares de codificación.

Para la implementación de la aplicación se siguieron estándares para la generación de código. Fue utilizada sangría para una mejor organización y claridad del código fuente. Se describió cada operación de cada una de las clases haciendo uso de comentarios basados en XML que introdujo Microsoft, donde fue especificado su propósito y una breve descripción de cada uno de los parámetros que recibe.

Se tuvo en cuenta una nomenclatura para el nombre de las clases y los objetos, así por ejemplo el nombre de todas las clases de control comienza con Gestor, en las clases que representan colecciones de objetos comienza con Lista seguido del nombre de la clase que contiene, en las clases de interfaz comienzan con frm. Los nombres de método comienzan con un verbo que indica la acción que va a realizar. Los nombres de controlador de eventos empiezan con un nombre que describe el tipo de evento seguido por el sufijo "EventHandler".

### 4.7 Modelo de despliegue.

El modelo de despliegue es un modelo de objetos, que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. El diagrama de despliegue de la aplicación se muestra a continuación.



**Fig. 4-22. Modelo de despliegue del sistema**

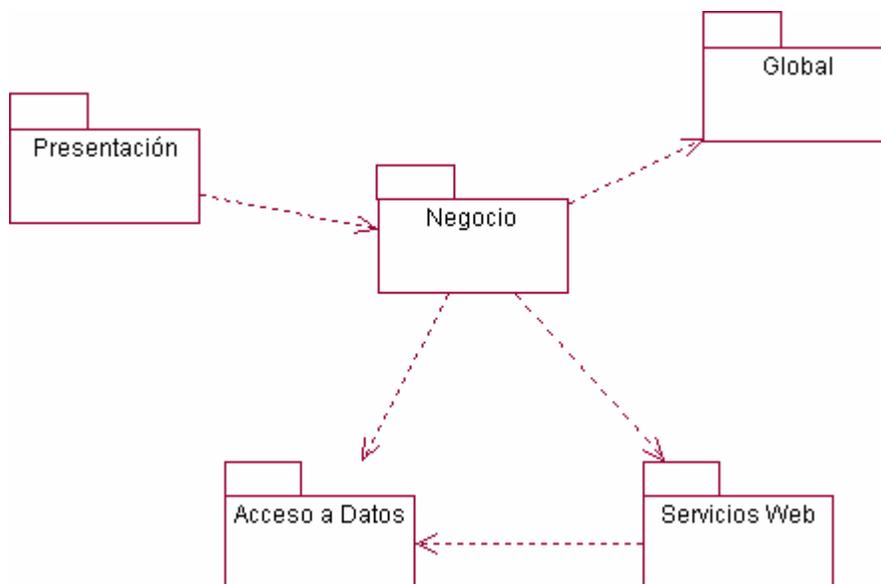
En el modelo de despliegue del sistema se representan tres nodos. Uno de estos es el Cliente, que representa los ordenadores de los usuarios donde se ejecuta la aplicación, desde los cuales podrán acceder, utilizando el protocolo HTTP, a los servicios Web de la plataforma publicados en el Servidor Web. Éste a su vez se comunica con el Servidor de Base de Datos (con SQL Server 2000 como sistema gestor de base de datos), a través del protocolo TCP/IP para realizar consultas y actualizaciones de la información que manipula el sistema.

### 4.8 Modelo de implementación.

El modelo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes. Describe también cómo se organizan y se relacionan unos con otros, definiendo un componente como el empaquetamiento físico de los elementos de un modelo, como es el caso de las clases del modelo de diseño.

A continuación se muestran los paquetes definidos y los componentes que pertenecen a cada uno de estos, así como la relación de dependencia que existe entre componentes. Además, se describe cada uno de estos componentes de acuerdo a su propósito y clases que contiene.

Los componentes que tienen extensión *.asmx* están relacionados cada uno de ellos con un componente de extensión *.cs*. Igualmente los componentes del paquete Presentación están relacionados cada uno de ellos con un componente con extensión *.resx*, los cuales no fueron representados en los diagramas para su mejor visualización.



**Fig. 4-23. Diagrama de paquetes del modelo de implementación**

#### 4.8.1 Diagrama de Componentes <Paquete Acceso a Datos>

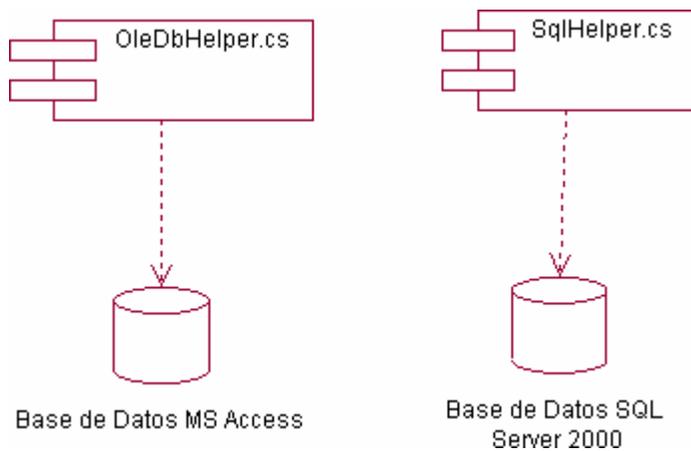


Fig. 4-24. Diagrama de componentes del paquete Acceso a Datos

#### 4.8.2 Diagrama de Componentes <Paquete Global>

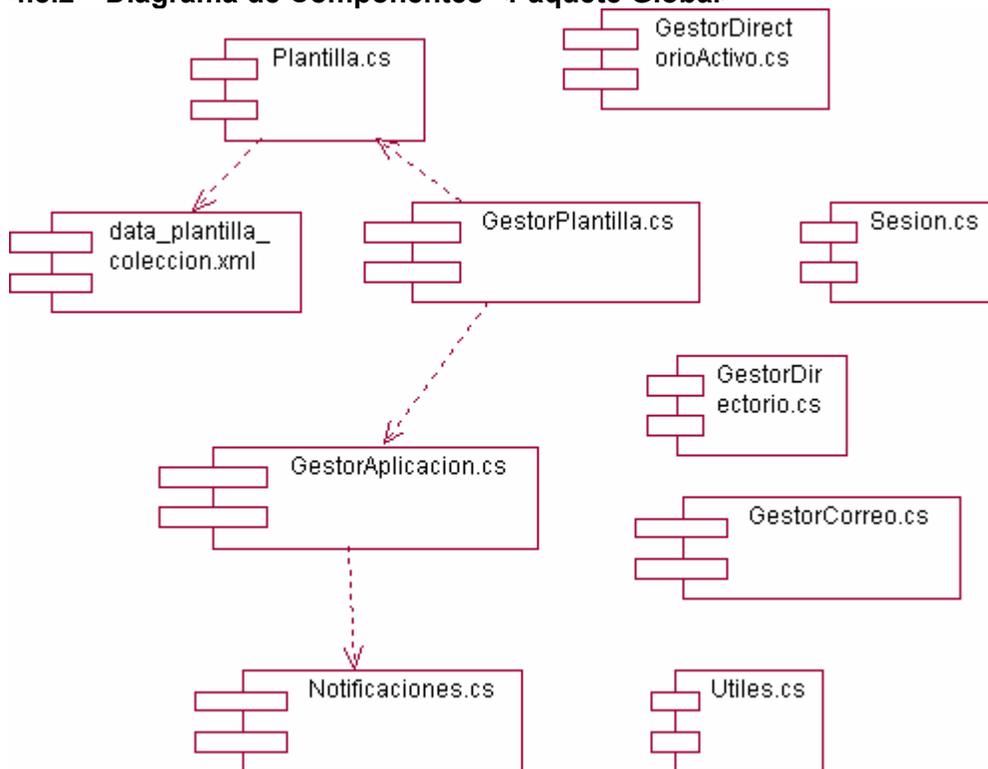
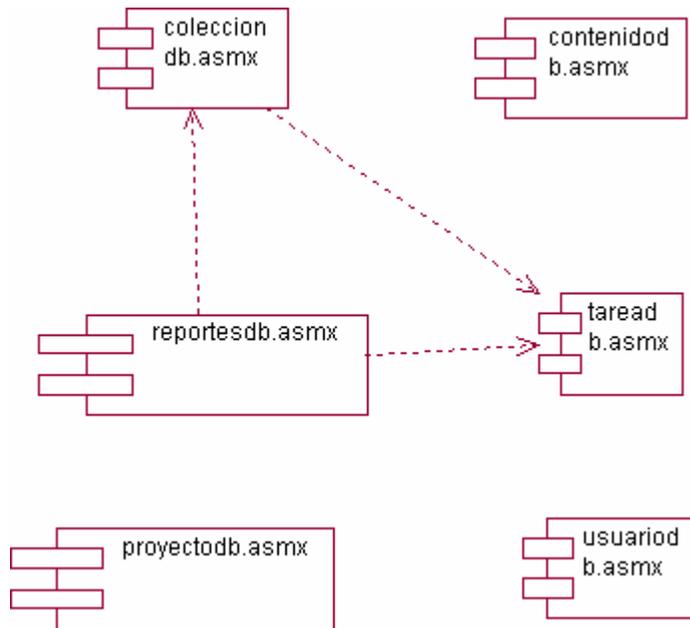


Fig. 4-25. Diagrama de componentes del paquete Acceso a Datos

### 4.8.3 Diagrama de Componentes <Paquete Servicios Web>



**Fig. 4-26. Diagrama de componentes del paquete Servicios Web**

4.8.4 Diagrama de Componentes <Paquete Negocio>

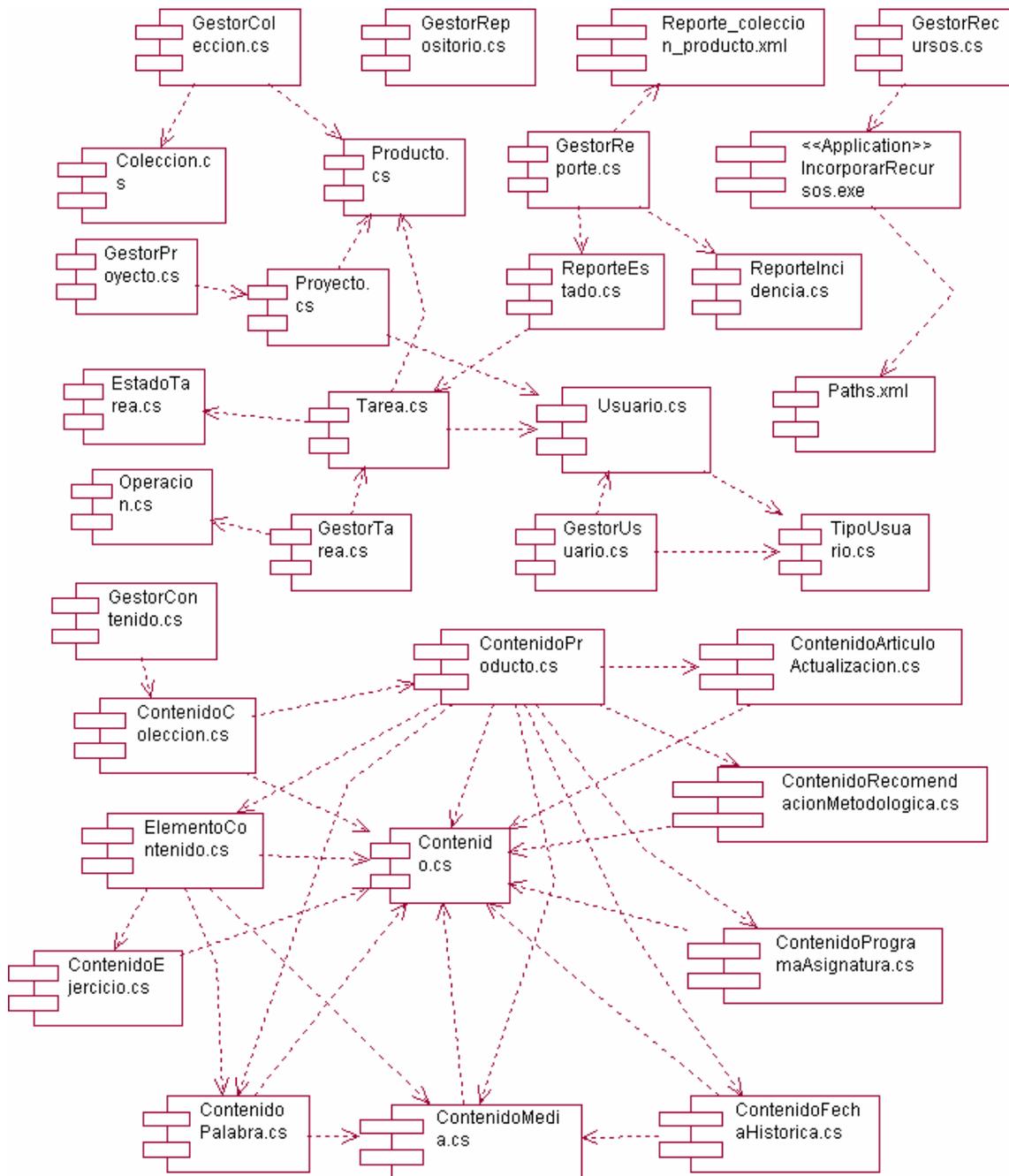


Fig. 4-27. Diagrama de componentes del paquete Negocio

4.8.5 Diagrama de Componentes <Paquete Presentación>

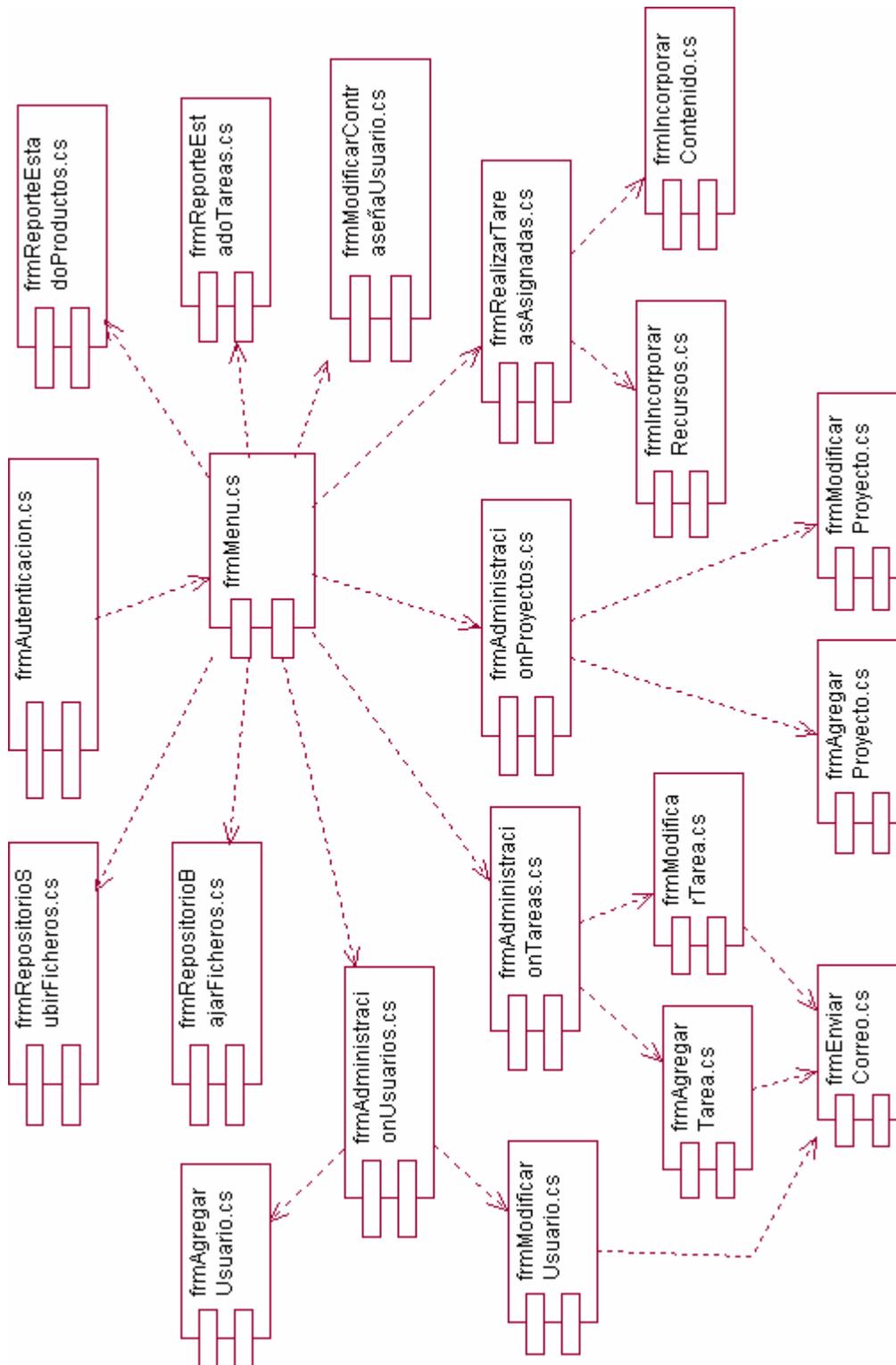


Fig. 4-28. Diagrama de componentes del paquete Presentación

### **4.9 Descripción de los principales componentes.**

[Ver ANEXO IV]

### **4.10 Conclusiones.**

En el presente capítulo se han desarrollado los flujos de trabajo de diseño e implementación que propone la metodología RUP. Se han elaborados los diagramas de clases del diseño, así como diagramas de secuencia para la realización de los caso de uso obtenidos en el capítulo anterior. Fueron descritos los principios de diseño en los que se basa la solución propuesta y fue elaborado el diagrama de clases persistentes, a partir del cual se obtuvo el modelo de datos. También fue realizada la modelación de los nodos en los que será distribuida la aplicación, especificando para cada uno de éstos el protocolo de comunicación y descrita la arquitectura en la que se basa la solución del software. Se obtuvo además el diagrama de componentes, elaborando una descripción que indica su propósito y contenido.

# CAPÍTULO 5

## ESTUDIO DE FACTIBILIDAD

### 5.1 Introducción.

En éste capítulo se realiza la estimación de costo en el estudio preliminar que tiene la realización del proyecto realizado. Basado en la herramienta COCOMO II (*Constructive Const Model*) se valoran una serie de resultados que permiten determinar un estimado de tiempo de duración del proyecto, cantidad de personas que se necesitan para desarrollarlo y el costo aproximado que tendrá, apoyándose en fórmulas de estimación y puntos de función establecidos por la herramienta.

Parte de este estudio de factibilidad lo constituyen los beneficios tangibles e intangibles, así como un análisis de costo beneficios que se encuentran después del cálculo de las estimaciones.

### 5.2 Planificación.

**Tabla 5-1. Entradas externas**

Nombre de la entrada externa (EI)	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación
Autenticación	1	2	Simple
Proyecto	3	6	Medio
Usuario	3	7	Medio
Tarea	3	5	Medio
Incorporar Recursos	1	3	Simple
Incorporar Contenido	1	1	Simple
Correo	1	3	Simple
Repositorio	2	4	Simple

**Tabla 5-2. Salidas externas**

Nombre de la salida externa (EI)	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación
Estado tareas	1	6	Simple
Estado productos	1	10	Simple

**Tabla 5-3. Ficheros lógicos internos**

Nombre del fichero interno (ILF)	Cantidad de records	Cantidad de elementos de datos	Clasificación
sirc_Proyecto	1	5	Simple
sirc_Coleccion	1	2	Simple
sirc_Producto	1	4	Simple

## CAPÍTULO 5. ESTUDIO DE FACTIBILIDAD

sirc_Tarea	1	8	Simple
sirc_EstadoTarea	4	2	Simple
sirc_Operacion	2	2	Simple
sirc_Usuario	1	8	Simple
sirc_TipoUsuario	1	2	Simple
sirc_Incidencia	1	5	Simple

**Tabla 5-4. Consulta externa**

Nombre de la consulta externa (EQ)	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación
Listado de proyectos	3	12	Medio
Listado de tareas	3	5	Simple
Listado de usuarios	6	5	Medio
Productos de una colección	2	2	Simple
Productos asignados a un proyecto	2	2	Simple

**Tabla 5-5. Puntos de Función desajustados**

Elementos	Simple		Medio		Complejo		Subtotal
	No	Peso	No	Peso	No	Peso	
Entrada Externa	5	3	3	4	0	6	27
Salida Externa	2	4	0	5	0	7	8
Fichero interfaz externa	0	5	0	7	0	10	0
Fichero lógico Interno	9	7	0	10	0	15	63
Consulta externa	3	3	2	4	0	6	17
<b>Subtotal (UFP)</b>							<b>115</b>

**Tabla 5-6. Tabla de características**

Características	Valor
Puntos de función desajustados(UFP)	115
Lenguajes	C# (80%)
	Lingo (5%)
	SQL (15%)
Instrucciones fuentes por puntos de función	59
	56
	39
Miles de instrucciones fuentes (Size)	<b>6.42</b>

**Tabla 5-7. Multiplicadores de esfuerzo:**

<b>Multiplicador</b>	<b>Descripción</b>	<b>Valor</b>
Complejidad y Confianza del Producto (RCPX)	La complejidad del producto es alta.	1.33
Reutilización Requerida (RUSE)	Se implementa código reutilizable para su aprovechamiento en el proyecto.	1.0
Inconvenientes de la Plataforma (PDIF)	La plataforma es estable. Requerimientos bajos de almacenamiento y tiempo de ejecución.	0.87
Capacidad Personal (PERS)	La capacidad del personal de desarrollo es alta. La continuidad del personal es alta.	0.83
Experiencia Personal (PREX)	El equipo tiene buen dominio y conocimiento del lenguaje de programación, plataforma y herramientas de desarrollo utilizados. No ha desarrollado aplicaciones similares, casi ninguna experiencia.	1.0
Facilidades (FCIL)	Se utilizan herramientas e instrumentos de programación modernos.	1.0
Planificación (SCED)	Los requerimientos de calendario de desarrollo son altos.	0.87
$\prod_{i=1}^7 EM_i$		<b>0.84</b>

**Tabla 5-8. Factores de Escala**

<b>Factor</b>	<b>Descripción</b>	<b>Valor</b>
PREC	Aspectos novedosos.	3.72
FLEX	Cierta flexibilidad	3.04
RESL	La arquitectura es sólida y los riesgos generalmente se mitigan. Poca incertidumbre, riesgos no son críticos.	1.41
TEAM	Interacciones altamente cooperativas. Considerable experiencia previa operando como equipo, visión y compromisos considerablemente compartidos	1.1

PMAT	Relación con el proceso de madurez del software. Nivel 3.	3.12
$\sum_{j=1}^5 SF_j$		<b>12.39</b>

**Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.**

Las ecuaciones que plantea COCOMO para calcular el esfuerzo, tiempo de desarrollo, cantidad de hombres y costos son las siguientes:

**Tabla 5-9. Cálculos**

Calculo de:	Valor	Justificación
<p><b>Esfuerzo</b></p> $PM = A \times Size^E \times \prod_{i=1}^n EM_i$ $E = B + 0.01 \times \sum_{j=1}^5 SF_j$	<b>16,76 ≈ 17 hombres/mes</b>	<p>A = 2.94</p> <p>Size = 6.42</p> $\prod_{i=1}^7 EM_i = 0.84$ <p>B = 0.91</p> $\sum_{j=1}^5 SF_j = 12.39$
<p><b>Tiempo de desarrollo</b></p> $TDEV = C \times PM^F$ $F = D + 0.2 \times (E - B)$	<b>8.65 ≈ 9 meses</b>	<p>C = 3.67</p> <p>PM = 16.76</p> <p>D = 0.28</p> <p>E = 1.03</p> <p>B = 0.91</p>
<p><b>Cantidad de hombres</b></p> $CH = PM / TDEV$	<b>1.94 ≈ 2 hombres</b>	<p>PM = 16.76</p> <p>TDEV = 8.65</p>
<p><b>Costo</b></p> $Costo = CH \times SP \times PM$	<b>\$14631.48</b>	<p>CH = 1.94</p> <p>SP (Salario medio) = \$450</p> <p>PM = 16.76</p>

**5.3 Beneficios tangibles e intangibles.**

El sistema propuesto traerá como resultado un ahorro del tiempo empleado en el desarrollo de los productos de las colecciones educativas para el MINED. Por otra parte se ahorrarán gran cantidad de recursos humanos y materiales que hasta el momento se usan en dicha labor, teniendo en cuenta que para el desarrollo de un producto de la colección Multisaber se tiene previsto que participe un profesor líder de proyecto y cinco estudiantes

durante un tiempo entre los 3 y 5 meses y en el caso de un producto de la colección ETP la estimación de tiempo tiende a incrementar debido a que presentan mas complejidad.

Mediante el desarrollo de este sistema se hará más eficiente y cómodo el trabajo por parte de cada uno de los participantes en el proceso de montaje del producto final mediante la incorporación automatizada de todos los recursos y contenidos. Por otra parte se logrará prescindir de personal capacitado en lenguajes de programación para el desarrollo de estos productos.

### **5.4 Análisis de costos y beneficios.**

Por cálculos realizados en este capítulo el costo de desarrollo de este sistema es aproximado a los \$14631.48 por concepto de salario contando con 2 hombres para su desarrollo en un tiempo cerca de los 9 meses, pero teniendo en cuenta que los desarrolladores con que se cuenta son estudiantes y no reciben salario y analizando los beneficios, en especial el ahorro del tiempo y el aumento de la eficiencia en al producción resulta factible el desarrollo de esta aplicación.

### **5.5 Conclusiones.**

El sistema propuesto trae consigo ahorro en tiempo, mayor productividad en el desarrollo de las multimedia educativas, disminución del esfuerzo, ahorro de recursos humanos y materiales, por lo que se concluye que es factible su implementación.

## CONCLUSIONES

En el presente trabajo se ha hecho un análisis del desarrollo de productos de las colecciones del MINED y se han determinado las dificultades que se presentan a la hora de realizar el montaje del producto final. Se ha hecho un estudio de diferentes alternativas que pueden ser usadas para resolver dicho problema y se ha propuesto el desarrollo de un sistema que permita la automatización del proceso de incorporación de recursos y contenido a estos productos, para lo cual se ha realizado un estudio de las tecnologías actuales que pueden ser empleadas para la implementación de la misma.

Este trabajo recoge las distintas etapas de desarrollo del sistema propuesto. Además se ha hecho un estudio de la factibilidad de este, analizando sus beneficios tangibles e intangibles y su costo llegando a la conclusión de que resulta factible la implementación del mismo.

Luego de todo este proceso de trabajo se puede concluir que el sistema propuesto da solución a la situación problemática que lo originó y que su explotación significará una mejora considerable en la calidad y eficiencia de los procesos que automatiza.

## RECOMENDACIONES

El sistema propuesto interactúa con una plantilla implementada en Macromedia Director, pero teniendo en cuenta que la política de la UCI es hacer una migración a software libre, se recomienda hacer estudios para lograr un correcto funcionamiento con una plantilla implementada en otra herramienta, como puede ser *Revolution* de la empresa RunRev (<http://www.runrev.com>).

Se recomienda continuar y perfeccionar la integración con el cliente de *Subversion*, de forma tal que se puedan implementar más funcionalidades respecto al control de los recursos que se manipulan en la plataforma SISME.

En vista a la política de migración hacia software libre llevada a cabo por nuestro país y a la que no se encuentra ajeno la UCI, se recomienda en etapas venideras hacer un profundo estudio sobre la migración paulatina de los sistemas que forman la plataforma SISME a software libre.

Se recomienda además agregarle al sistema nuevas funcionalidades que extiendan su alcance al proceso de revisión de la calidad, permitiendo que se pueda registrar los errores detectados y así ofrecer al Jefe de Proyecto nuevos reportes que ofrezcan más información sobre el estado de desarrollo de un producto.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Catálogo de Software Educativo Cubano – InstEd  
<http://www.rimed.cu/catalogo/default.asp?Acc=MO&Nombre=InstEd> (21/01/2005)
- [2] Desarrollador 5 estrellas. Introducción al entorno de desarrollo .NET  
<http://www.microsoft.com/latam/msdn/comunidad/dce/> (21/01/2006)
- [3] .NET Framework, uno para todos y todos para uno. Por Jorge Serrano Pérez  
<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art65.asp> (21/01/2006)
- [4] Características de Visual Studio .NET  
<http://www.microsoft.com/latam/vstudio/producto/caracteristicas.asp> (19/01/2006)
- [5] Ayuda del Visual Studio .NET 2003
- [6] .NET. Wikipedia. 2005.  
[http://es.wikipedia.org/wiki/.NET\\_de\\_Microsoft](http://es.wikipedia.org/wiki/.NET_de_Microsoft) (17/01/2006)
- [7] Aspectos básicos de XML Web Services  
<http://www.microsoft.com/spanish/msdn/articulos/archivo/280202/voices/webservbasics.asp>  
(21/01/2006)
- [8] Aplicaciones basadas en Web Internet/Intranet.  
<http://www.janium.com/phpws/index.php>. (18/01/2006)
- [9] Jacobson, Ivar, Booch, Grady, Rumbaugh, James. *El proceso Unificado de Desarrollo de Software*. Primera edición en español. Addison Wesley. 2000
- [10] ¿Qué es UML?  
<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/c12.html> (25/01/2006)
- [11] C Sharp  
[http://es.wikipedia.org/wiki/C\\_Sharp](http://es.wikipedia.org/wiki/C_Sharp) (17/01/2006)
- [12] Características futuras del lenguaje de programación C#  
<http://www.microsoft.com/spanish/msdn/articulos/archivo/050503/voices/vbconC.asp>  
(17/01/2006)
- [13] SQL  
<http://es.wikipedia.org/wiki/SQL> (25/01/2006)
- [14] Microsoft SQL Server  
[http://es.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://es.wikipedia.org/wiki/Microsoft_SQL_Server) (24/01/2006)

## **GLOSARIO DE TÉRMINOS**

**Cast externos:** Ventanas de cast que se guardan fuera de la película de Macromedia Director. Tiene una extensión .cst y puede ser usada y compartida por varias películas.

**Cast members:** Concepto usado en Macromedia Director. En español miembros de reparto. Son elementos de la película, tales como sonidos, textos, gráficos, animaciones, videos, películas de Macromedia Flash, etc. que se guardan en las ventanas de cast.

**Colección ETP:** Colección del MINED para la Educación Técnica y Profesional denominada “Aprender técnica” que está en proceso de producción. La misma está formada por 58 software educativos que se producirán en diferentes etapas de desarrollo. Ellos responden enteramente a las diferentes especialidades y perfiles de esta educación.

**Colección Multisaber:** Colección del MINED para la Educación Primaria que ha tenido gran aceptación y cuenta con varios avales de las escuelas primarias por la cobertura que presenta en casi todos los temas abordados en este nivel.

**Estructura interna del producto plantilla:** Información guardada en un fichero XML, que describe cuántos cast externos tiene el Producto Plantilla y los members que contiene cada uno de ellos.

**Excepción:** Es un evento que ocurre durante la ejecución del programa que interrumpe el flujo normal de las sentencias. Son una forma clara para controlar los errores sin confundir el código con muchas instrucciones de control del error.

GPE: Grupo de Plataformas Educativas.

**HTTP:** Conjunto de reglas para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la World Wide Web.

**Incidencia:** Es la notificación de un error que hubo en la incorporación de recursos. Permite saber qué recursos faltan por incorporar al producto plantilla.

**Producto:** Aplicación multimedia perteneciente a una de las colecciones definidas. Todos los productos de una colección cumplen las mismas pautas de diseño y funcionamiento.

**Repositorio:** Es prácticamente lo mismo que un servidor de ficheros ordinario, salvo que recuerda todos los cambios que se hayan hecho a sus ficheros y directorios. Esto permite que pueda recuperar versiones antiguas de sus ficheros y examinar la historia de cuándo y cómo cambiaron sus datos.

**SISME:** Sistemas de Multimedia Educativas. Plataforma de desarrollo de productos de las colecciones de software educativos para las diferentes enseñanzas.

**TCP/IP:** Se refieren a dos protocolos de red: *Transmission Control Protocol* (Protocolo de Control de Transmisión) e *Internet Protocol* (Protocolo de Internet). Los diferentes protocolos de la suite TCP/IP trabajan conjuntamente para proporcionar el transporte de datos dentro de Internet (o Intranet). En otras palabras, hacen posible que poder acceder a los distintos servicios de la Red.

**TortoiseSVN:** Es un cliente gratuito de código abierto para el sistema de control de versiones Subversion. Esto es, TortoiseSVN maneja ficheros y directorios a lo largo del tiempo. Los ficheros se almacenan en un repositorio central. Está desarrollado bajo la Licencia Pública General GNU (GPL).

**Ventana de Cast:** Es una de las ventanas que aparecen en el área de trabajo de Macromedia Director. Contiene los miembros de reparto de la película. Se puede usar la ventana de Cast para crear o importar nuevos miembros de reparto y ver los existentes, así como controlar las propiedades de cada uno de ellos. Una película de Macromedia Director puede contener uno o varias ventanas de cast.

**XML:** *Extensible Markup Language* (Lenguaje extensible de etiquetas) Es un meta-lenguaje que permite definir lenguajes de marcado adecuado a usos determinados. Se propone como lenguaje de bajo nivel (a nivel de aplicación, no de programación) para intercambio de información estructurada entre diferentes plataformas.

**.NET Passport:** Microsoft® .NET Passport es un servicio basado en Web, diseñado para que el registro a los sitios Web sea rápido y sencillo. .NET Passport permite a los sitios participantes autenticar a un usuario con un solo conjunto de credenciales de registro, eliminando la necesidad de recordar numerosas contraseñas y nombres de registro.

## ANEXO I. MODELO DE CASOS DE USO DEL SISTEMA

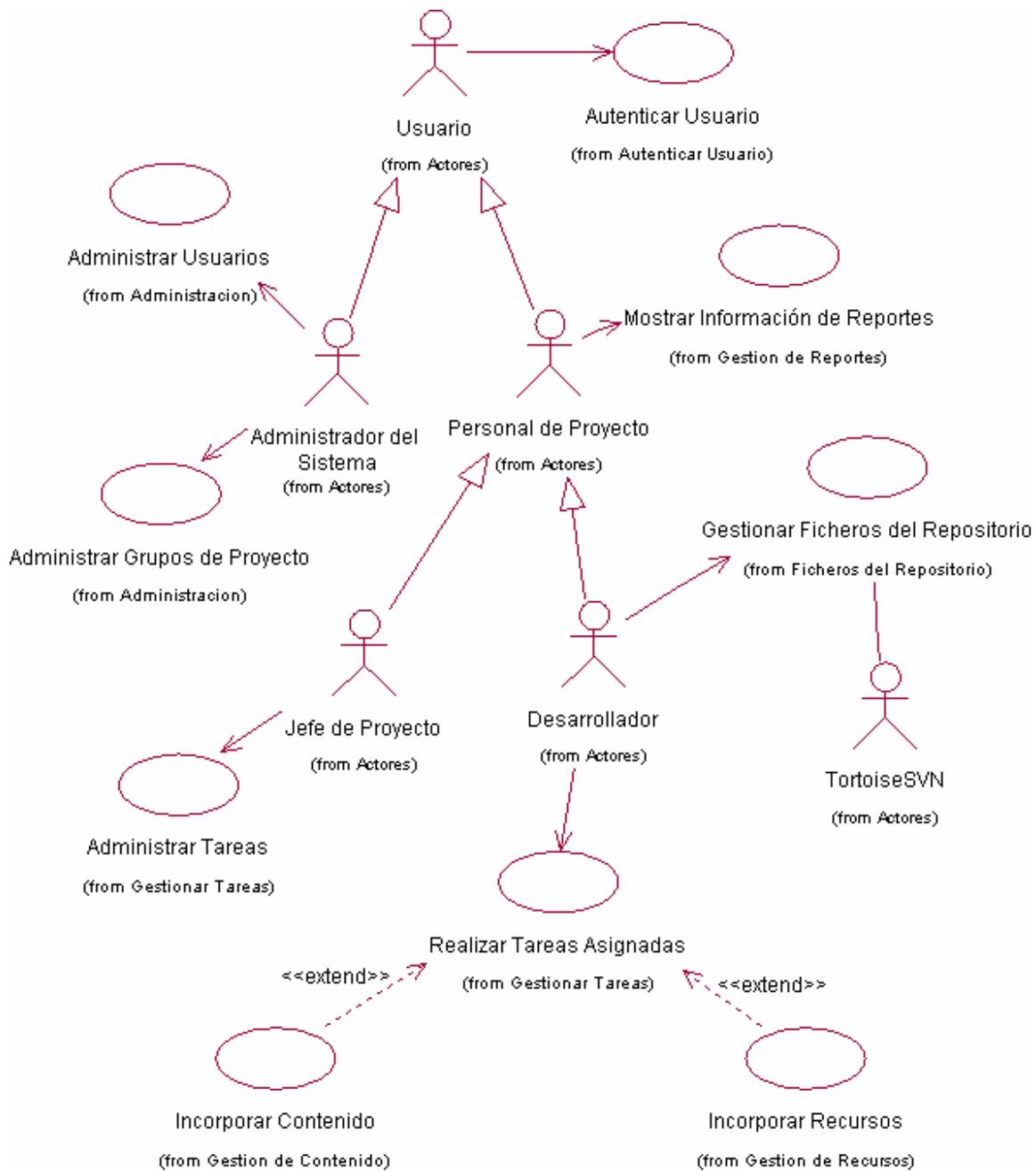


Fig. 6-1. Modelo de casos de uso del sistema

ANEXO II. DIAGRAMAS DE SECUENCIA

Diagramas de secuencia del CU "Realizar Tareas Asignadas"

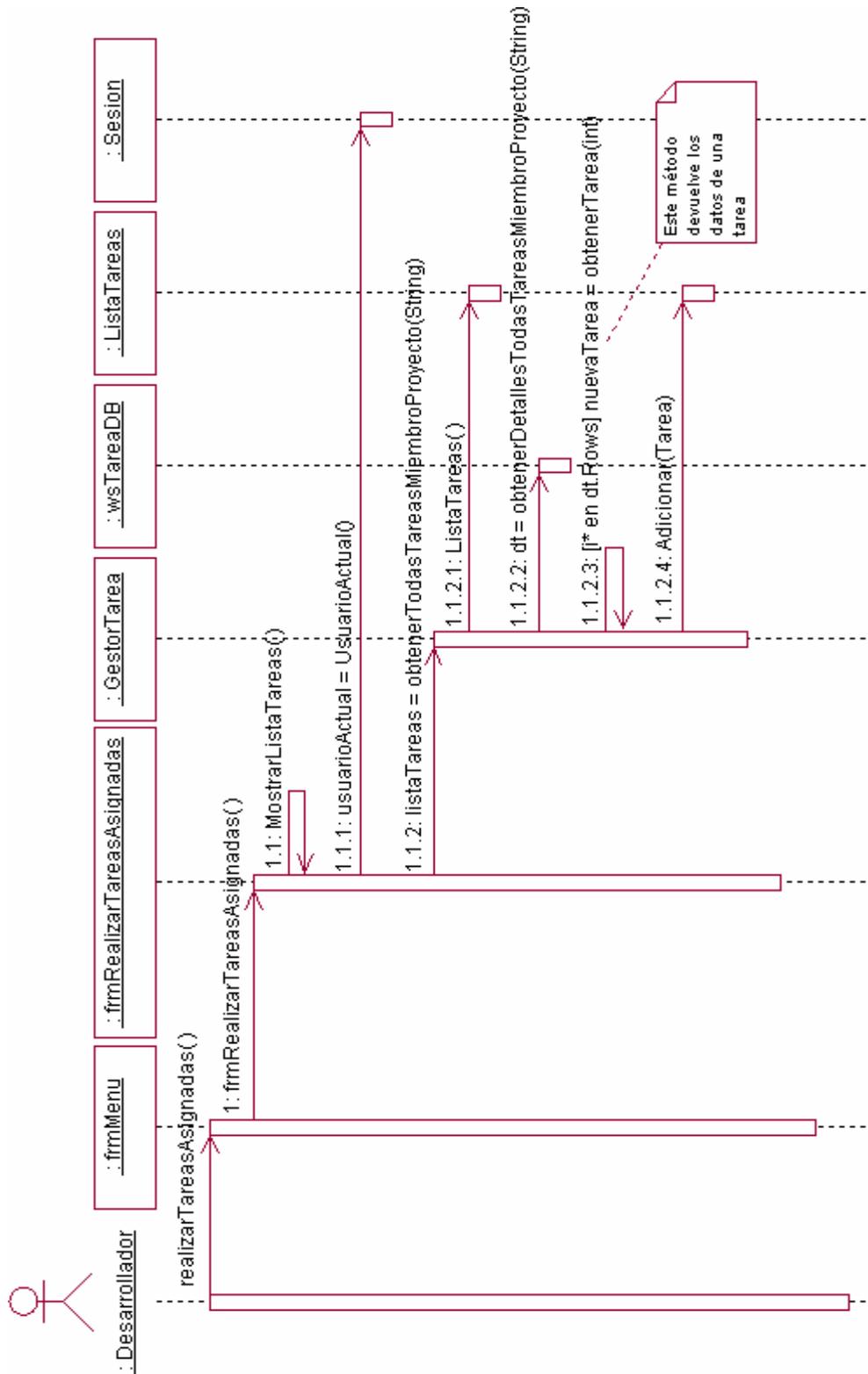


Fig. 6-2. Realizar Tareas Asignadas (Sección "Principal")

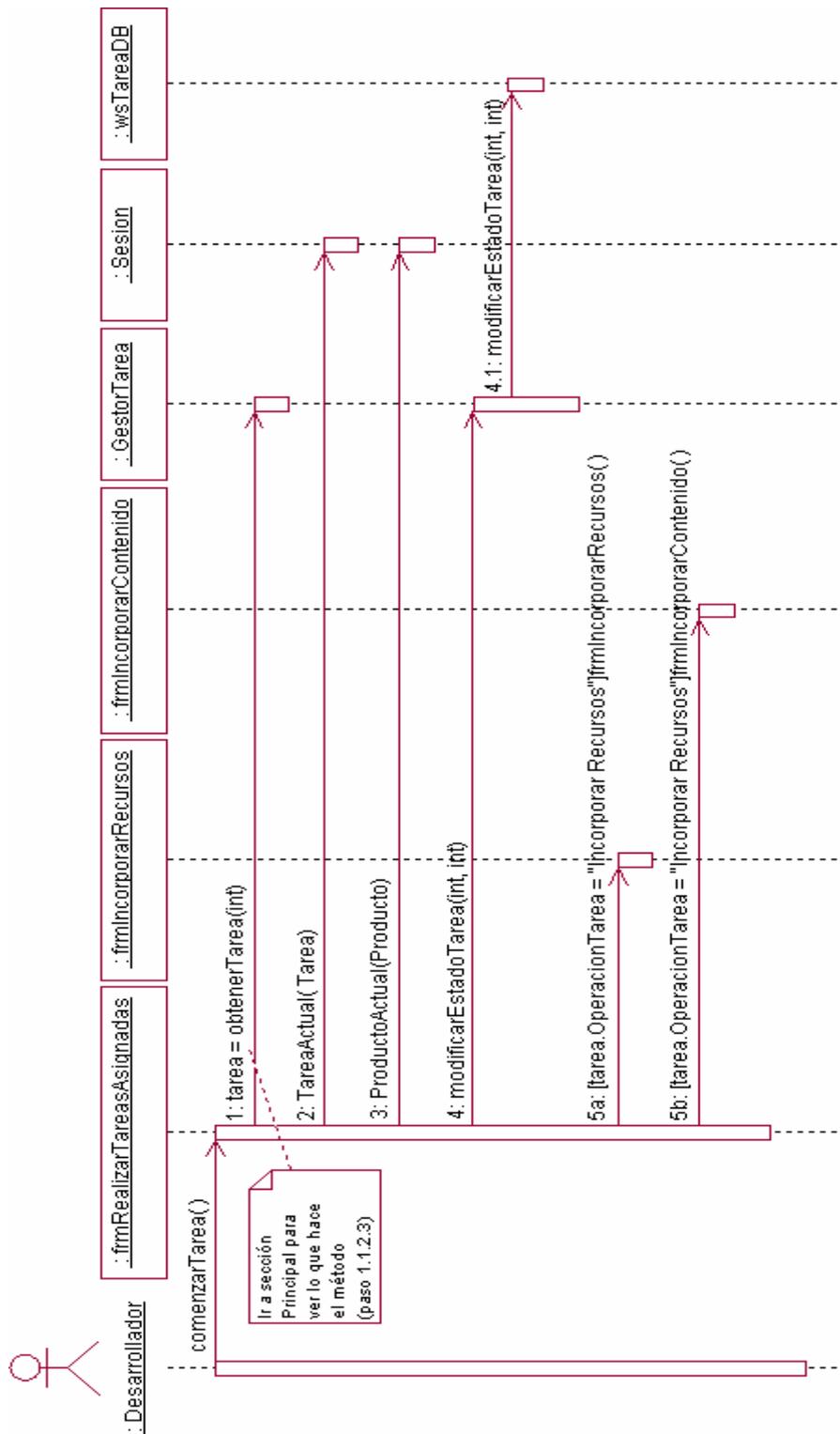


Fig. 6-3. Realizar Tareas Asignadas (Sección "Realizar Tarea")

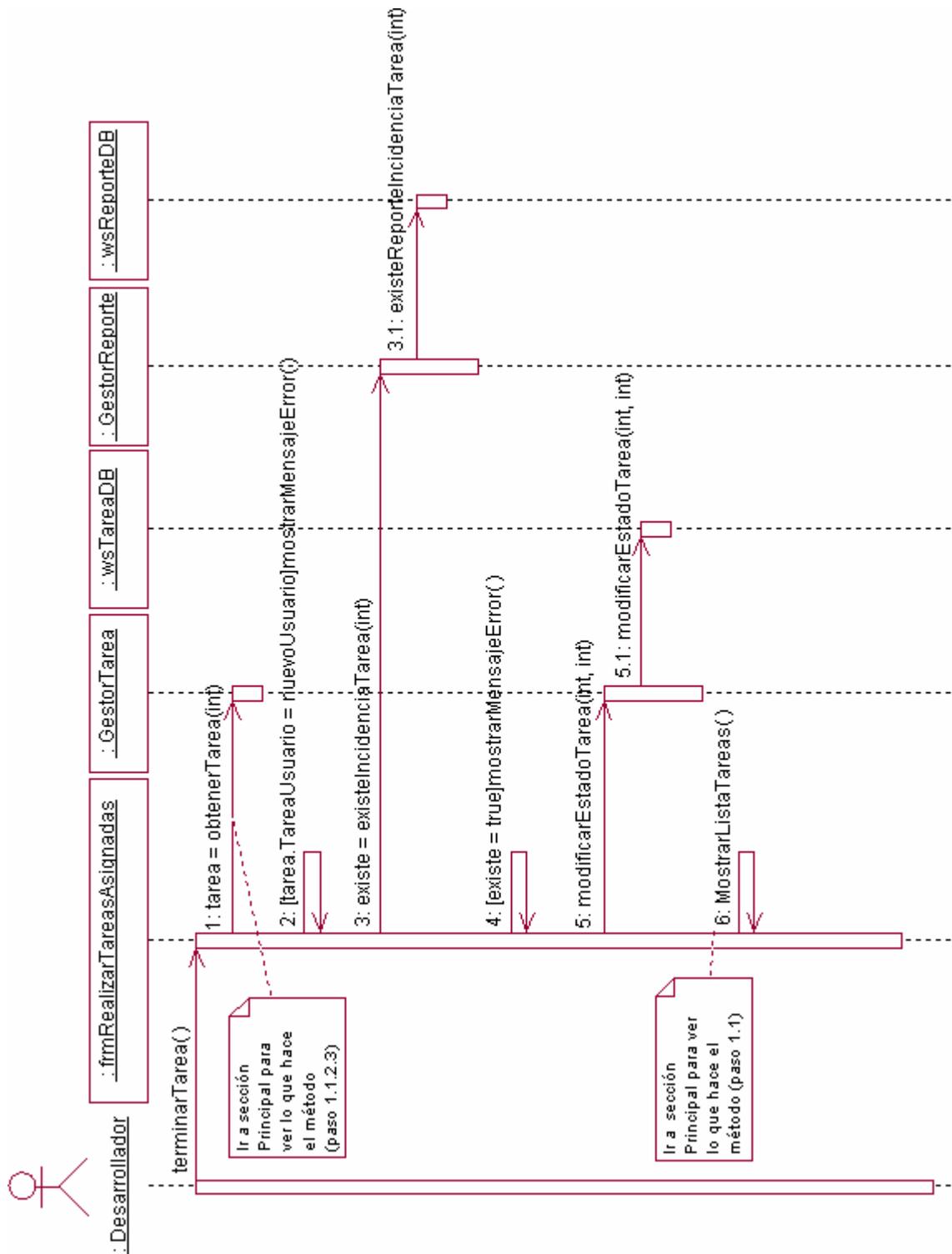


Fig. 6-4. Realizar Tareas Asignadas (Sección "Terminar Tarea")

Diagrama de secuencia del CU “Incorporar Contenido”

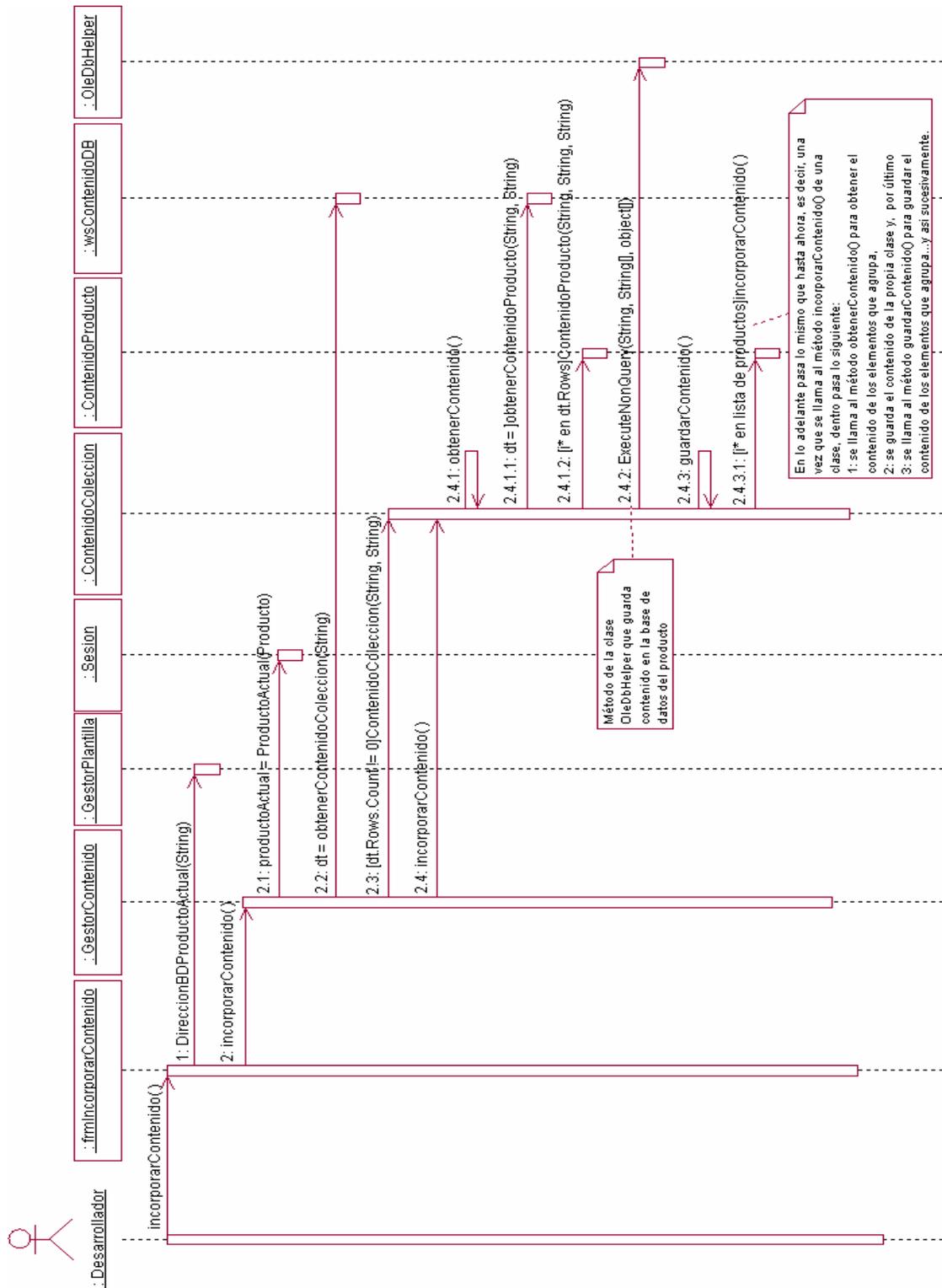


Fig. 6-5. Incorporar Contenido

Diagrama de secuencia del CU “Incorporar Recursos”

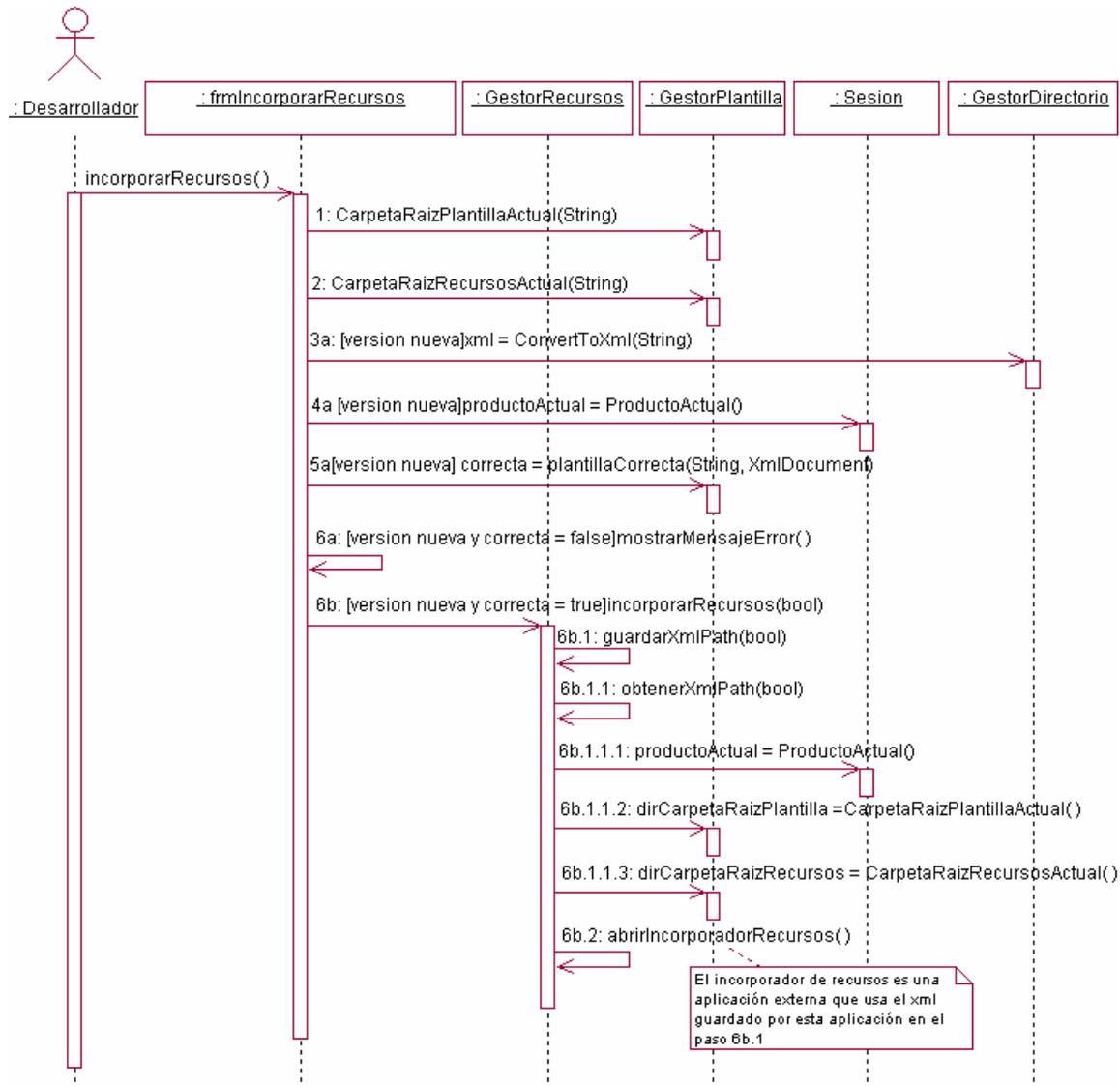


Fig. 6-6. Incorporar Recursos

Anexo # Diagramas de secuencia del CU "Administrar Tareas"

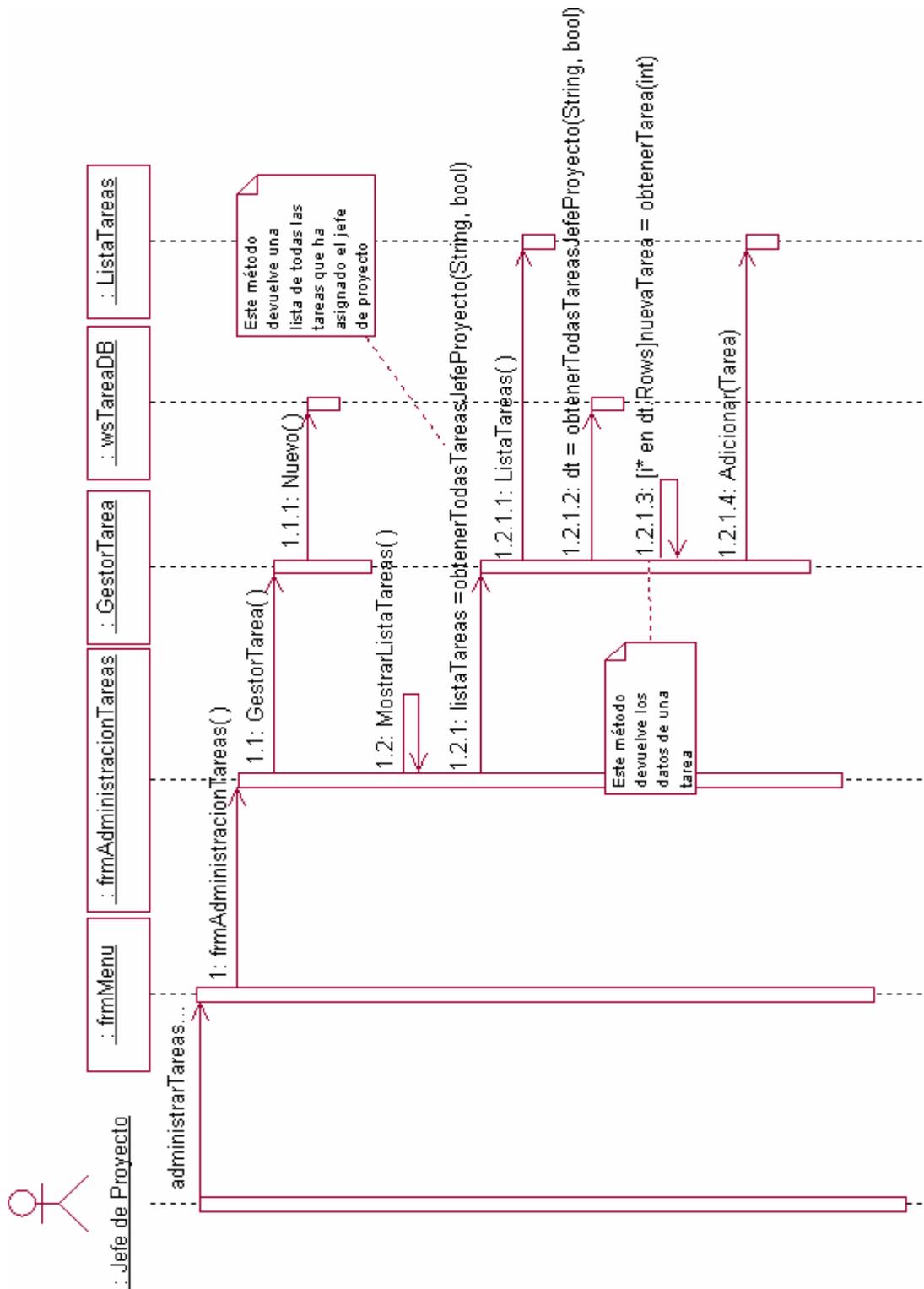


Fig. 6-7. Administrar Tareas (Sección "Principal")

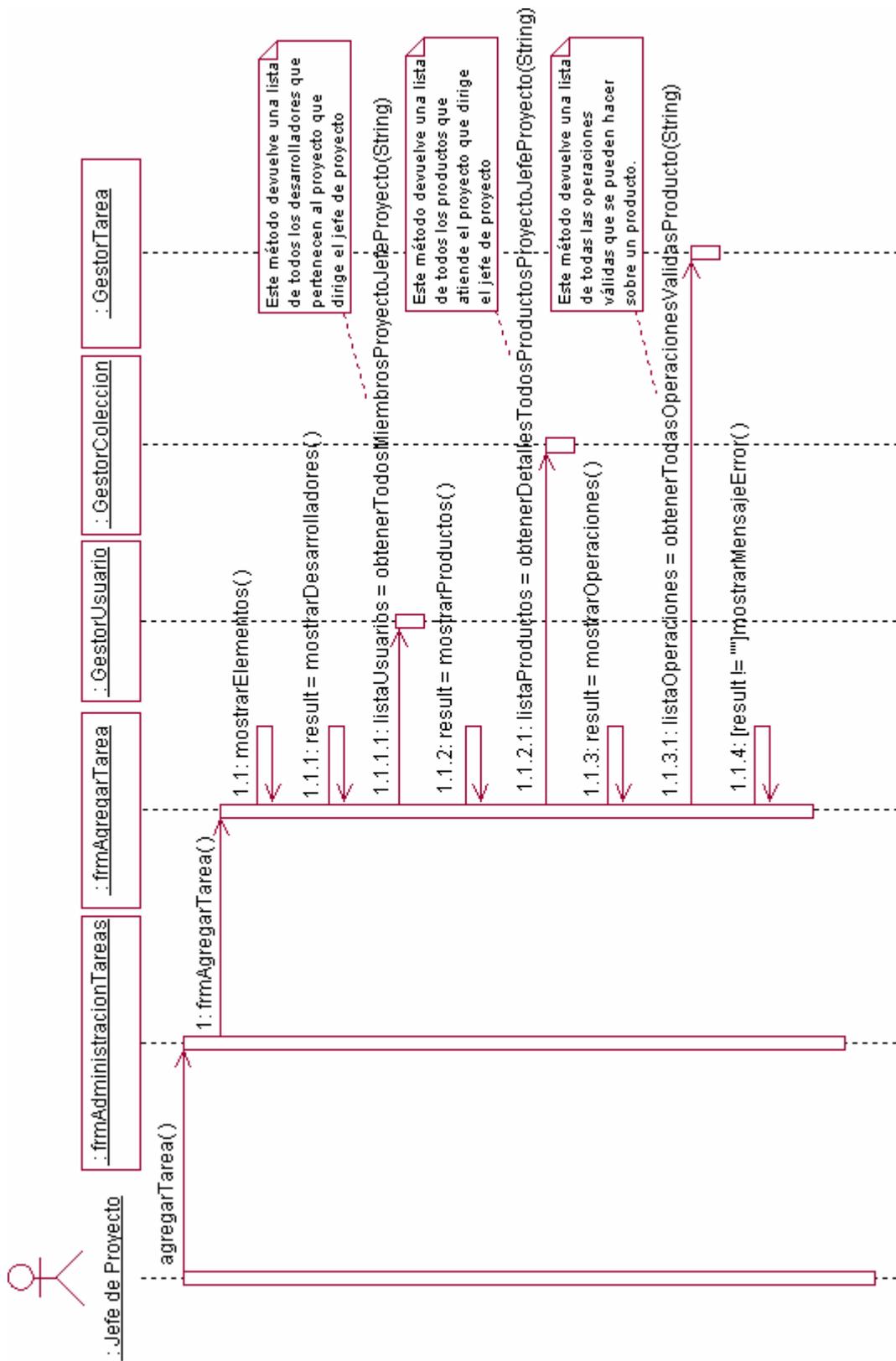


Fig. 6-8. Administrar Tareas (Sección "Agregar Tarea" Escenario Mostrar Pantalla)

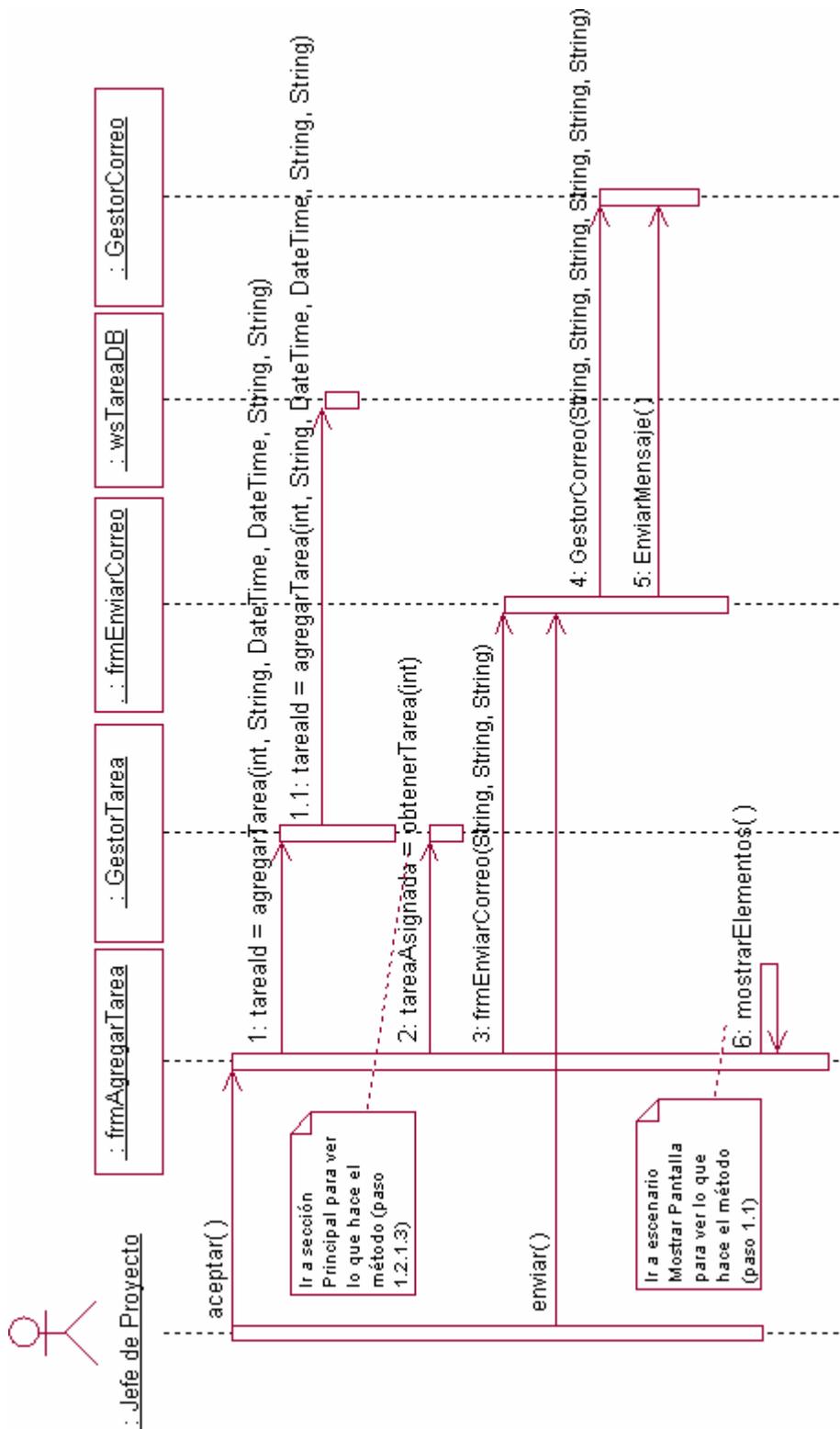


Fig. 6-9. Administrar Tareas (Sección "Agregar Tarea" Escenario Agregar)

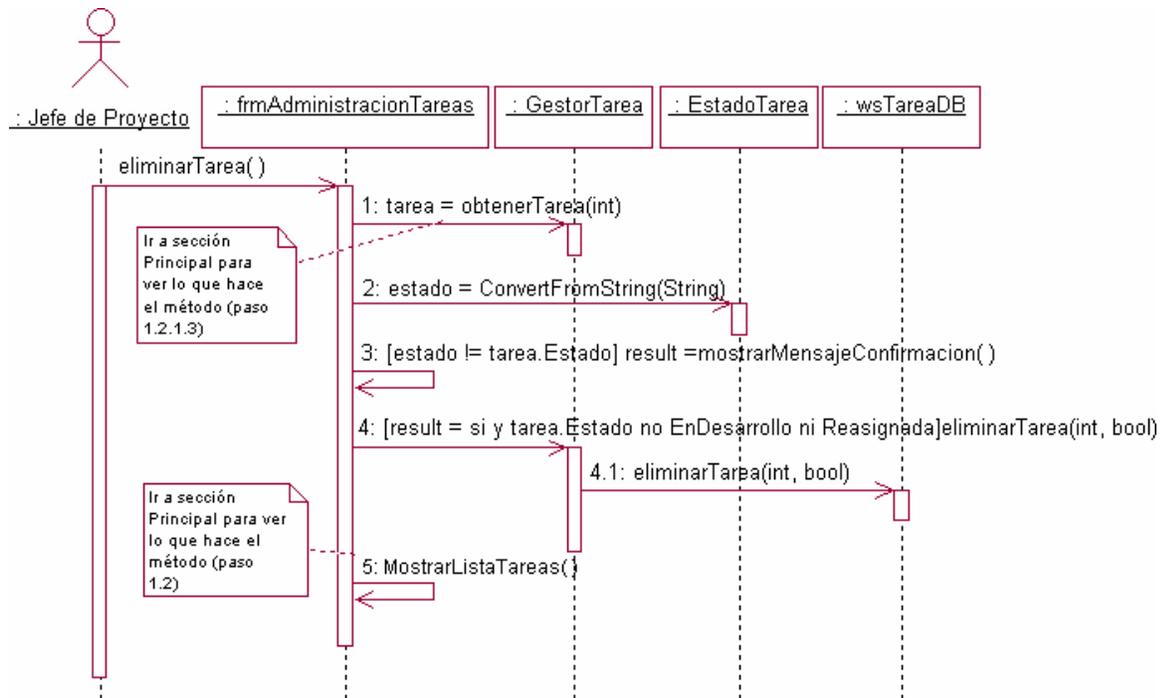


Fig. 6-10. Administrar Tareas (Sección "Eliminar Tarea")

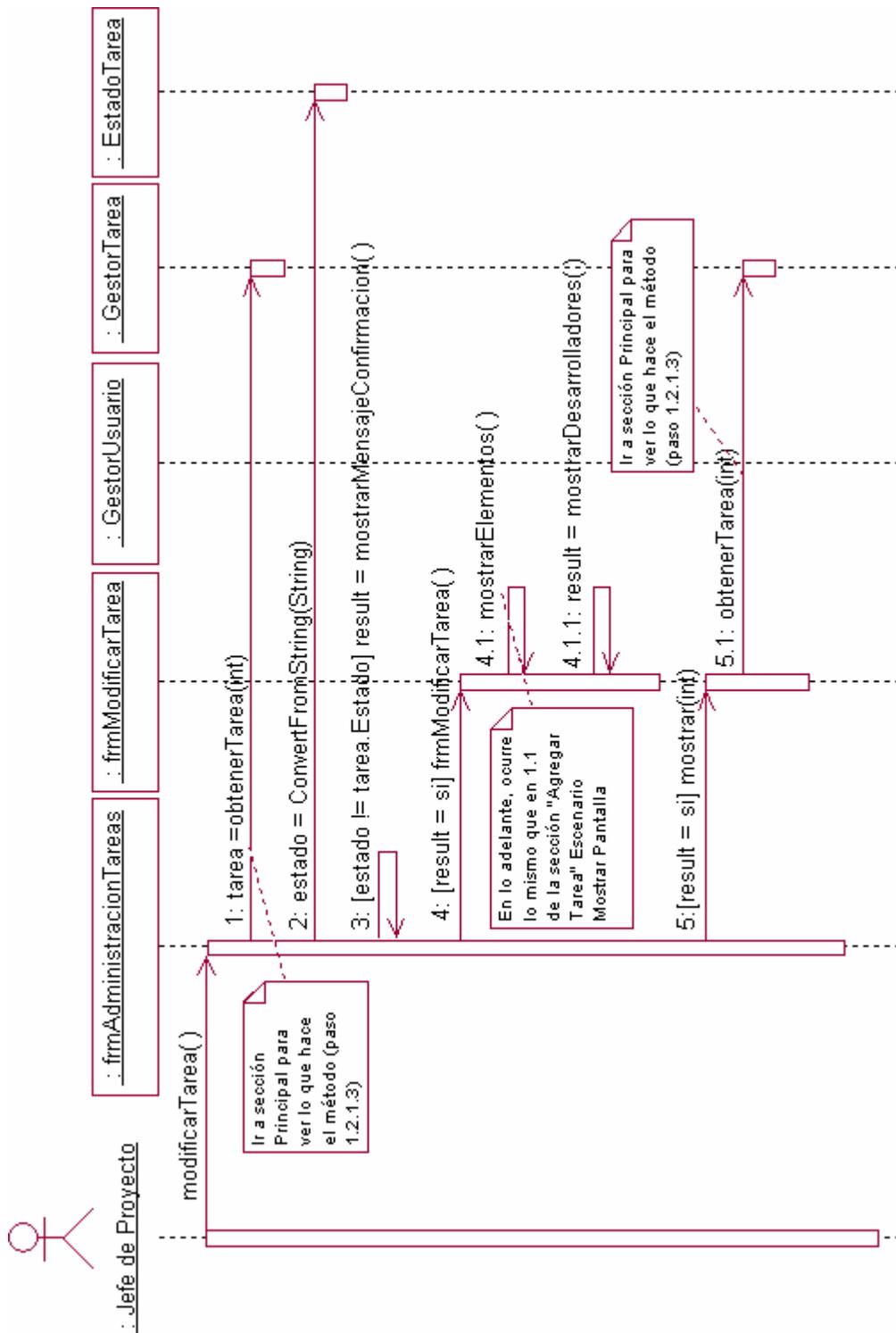


Fig. 6-11. Administrar Tareas (Sección "Modificar Tarea" Escenario Mostrar Pantalla)

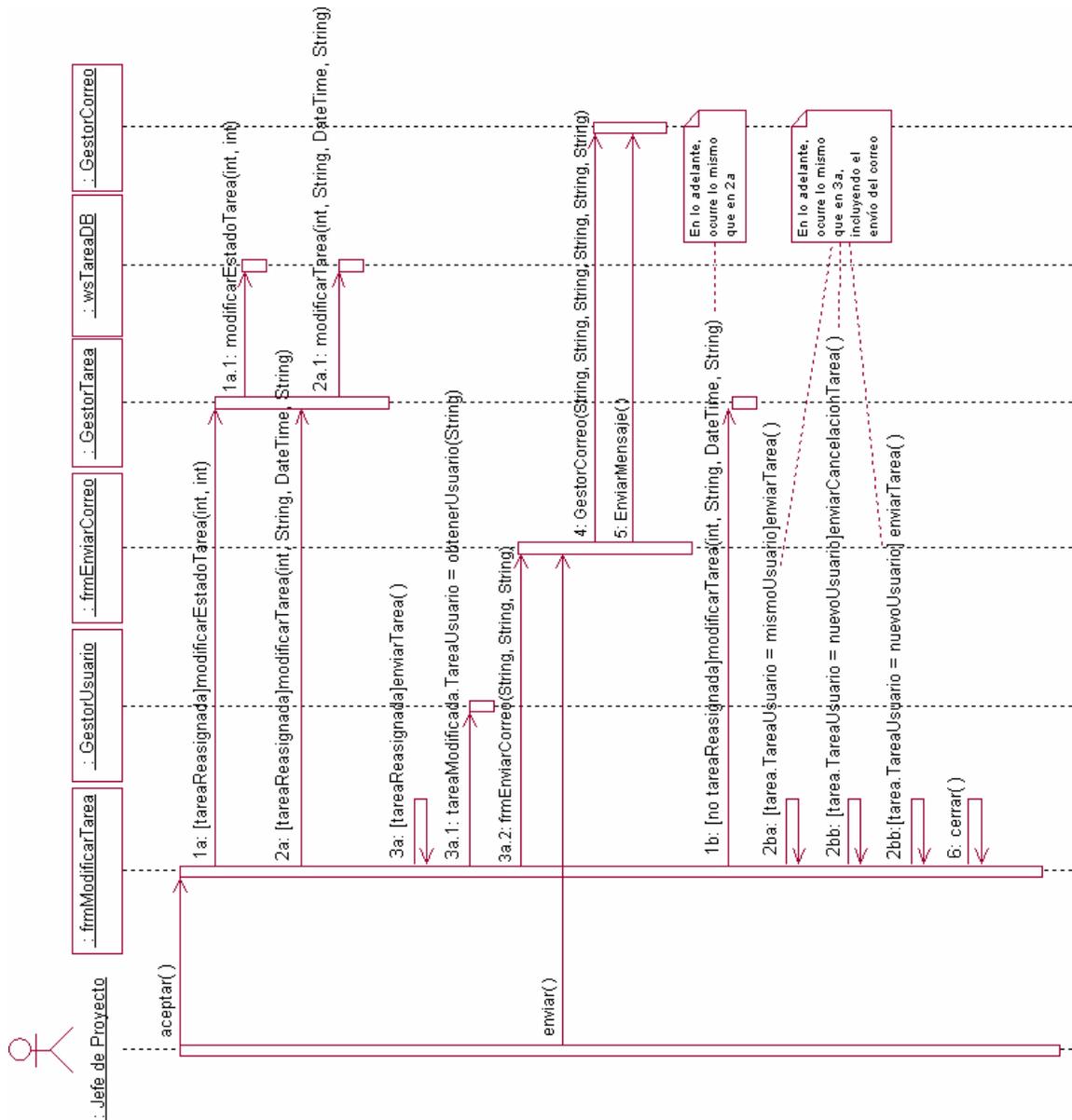


Fig. 6-12. Administrar Tareas (Sección "Modificar Tarea" Escenario Modificar)

Diagramas de secuencia del CU "Administrar Usuarios"

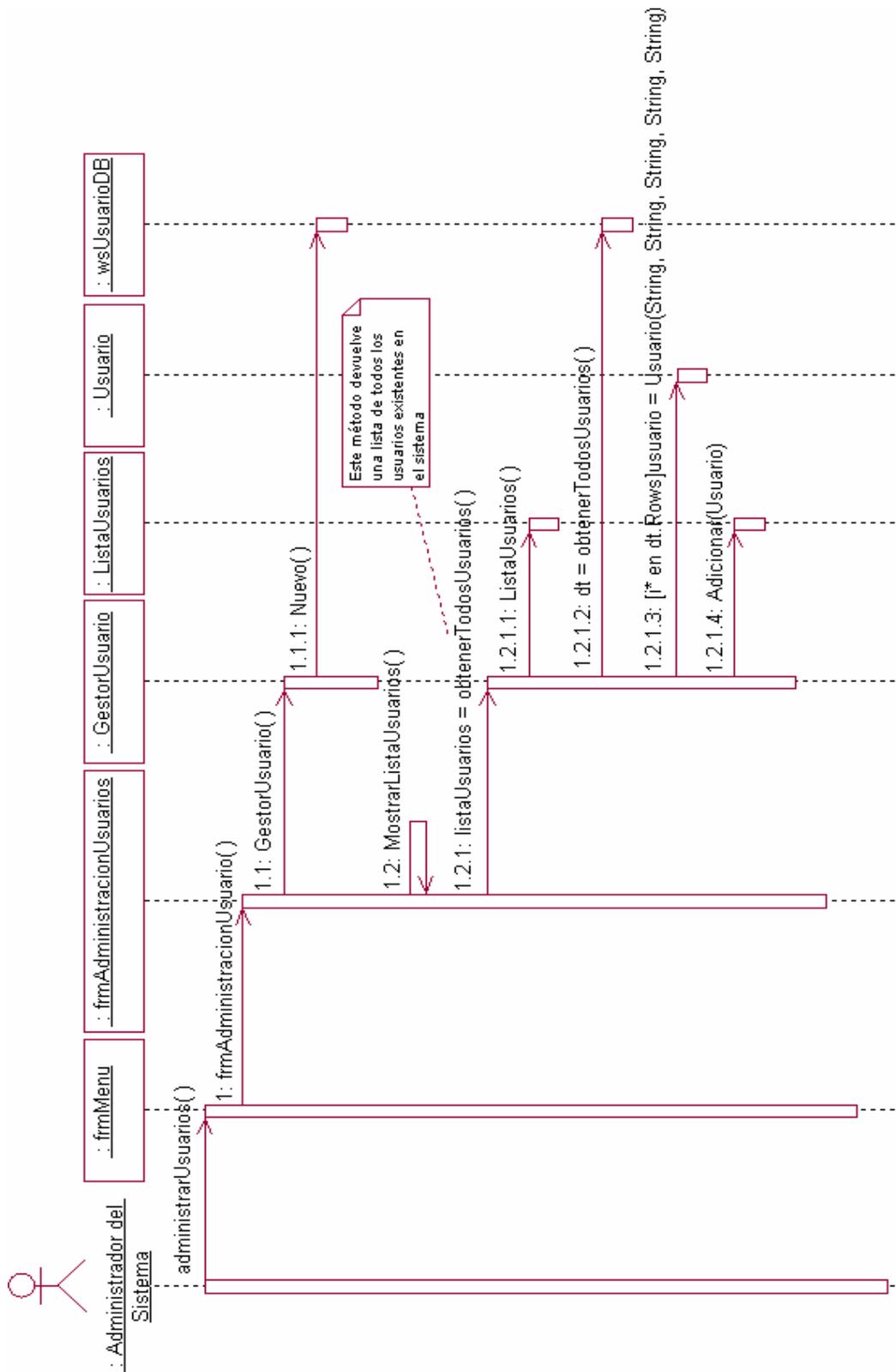


Fig. 6-13. Administrar Usuarios (Sección "Principal")

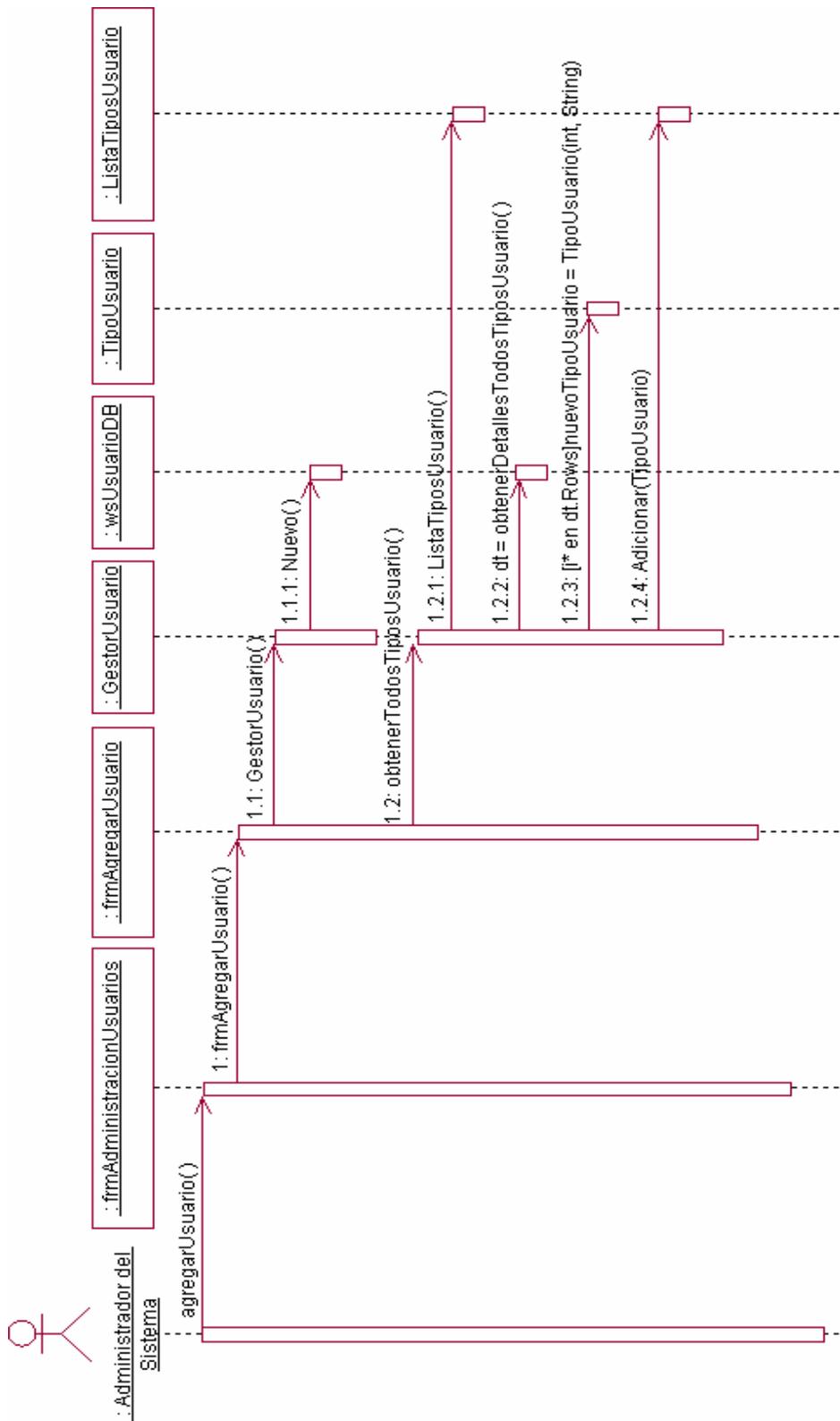


Fig. 6-14. Administrar Usuarios (Sección "Agregar Usuario" Escenario Mostrar Pantalla)

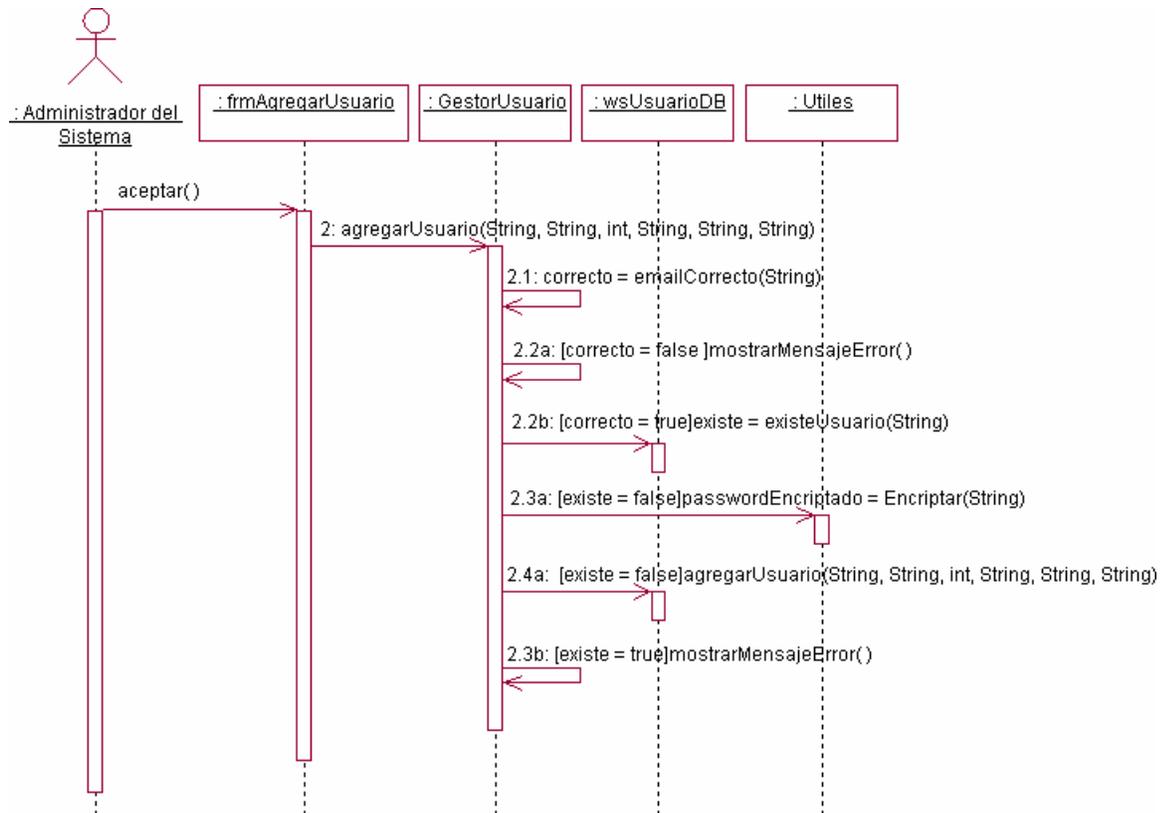


Fig. 6-15. Administrar Usuarios (Sección "Agregar Usuario" Escenario Agregar)

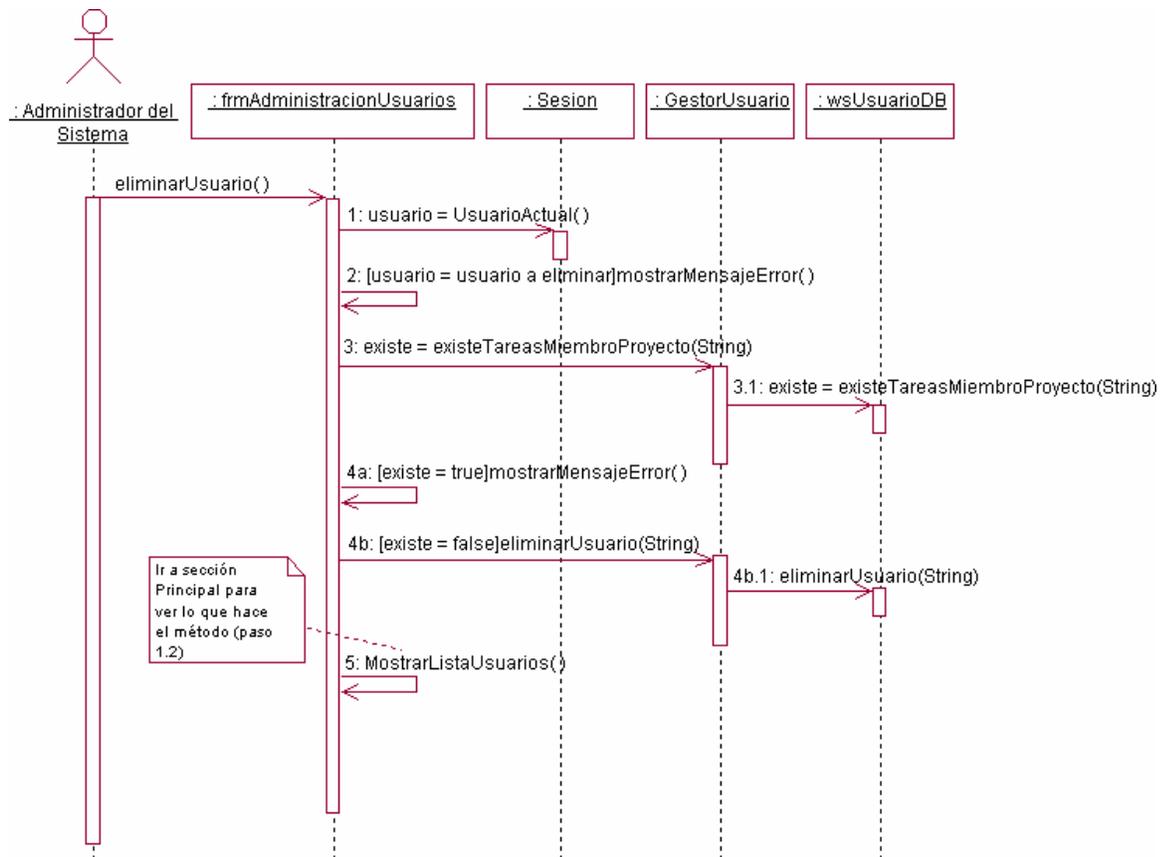


Fig. 6-16. Administrar Usuarios (Sección "Eliminar Usuario")

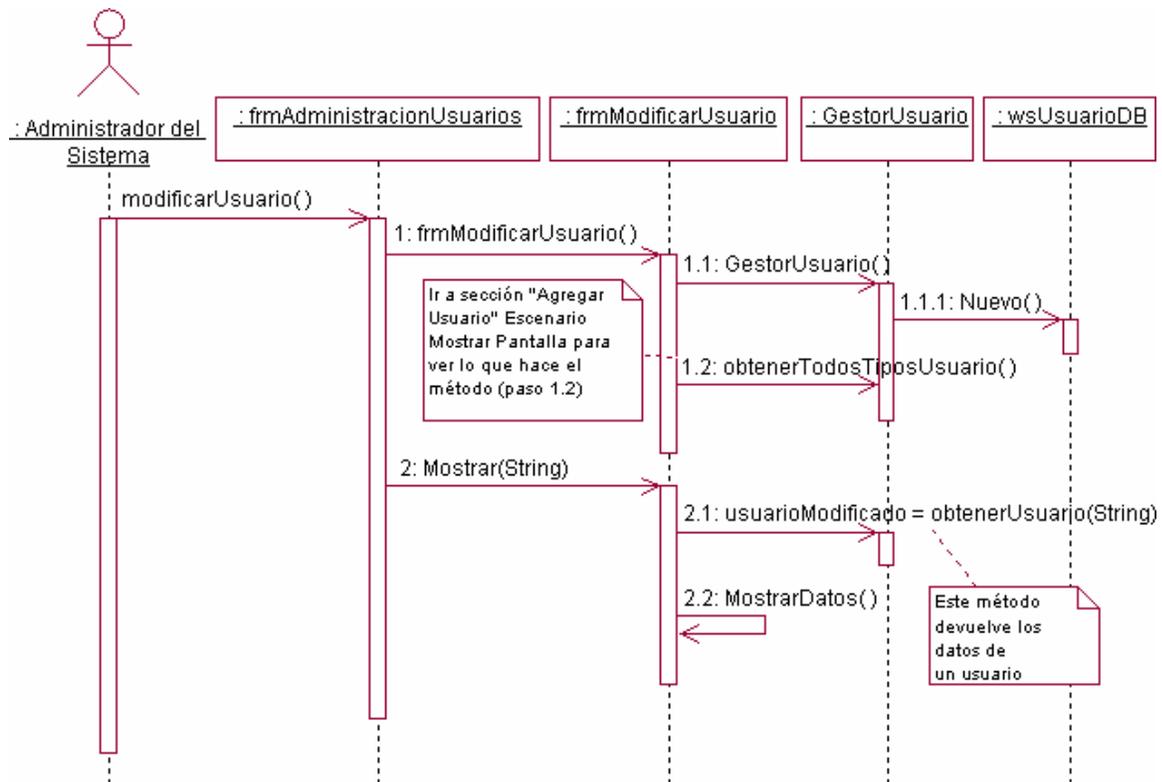


Fig. 6-17. Administrar Usuarios (Sección "Modificar Usuario" Escenario Mostrar Pantalla)

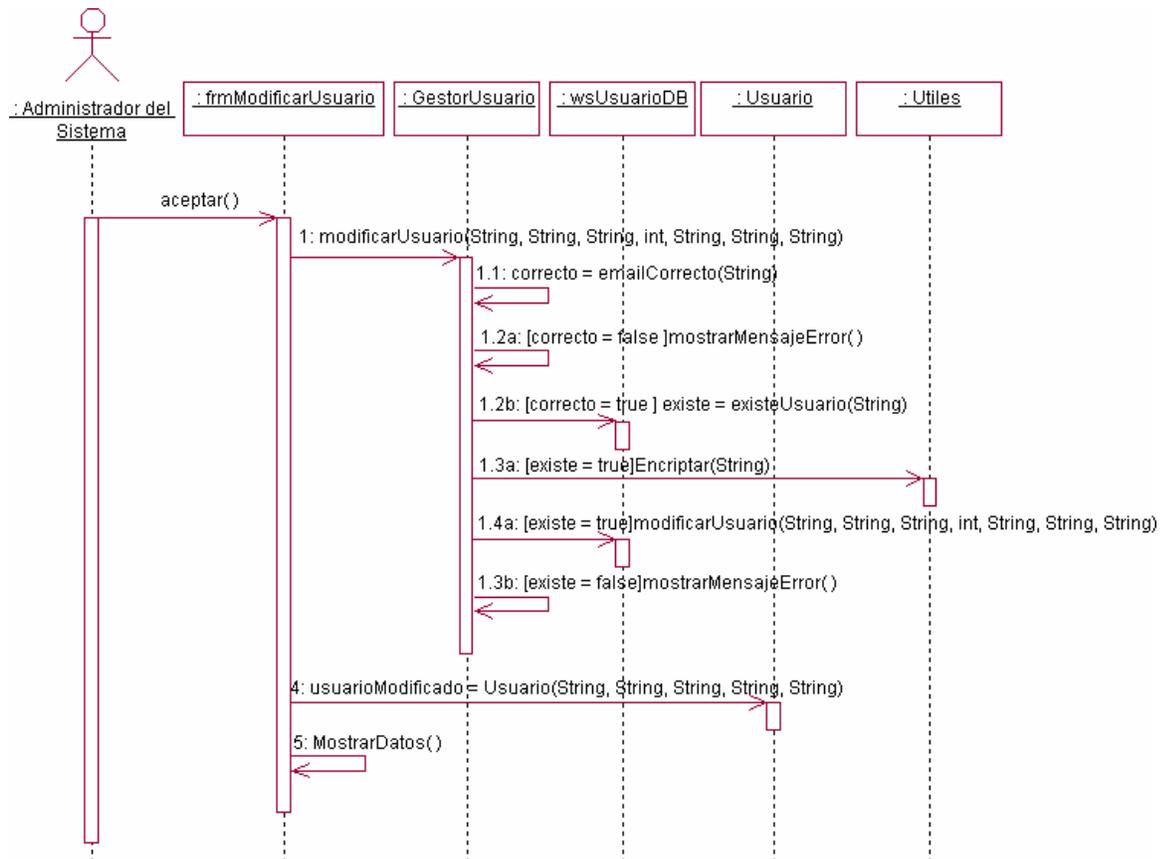


Fig. 6-18. Administrar Usuarios (Sección "Modificar Usuario" Escenario Modificar)

Diagramas de secuencia del CU "Mostrar Información de Reportes"

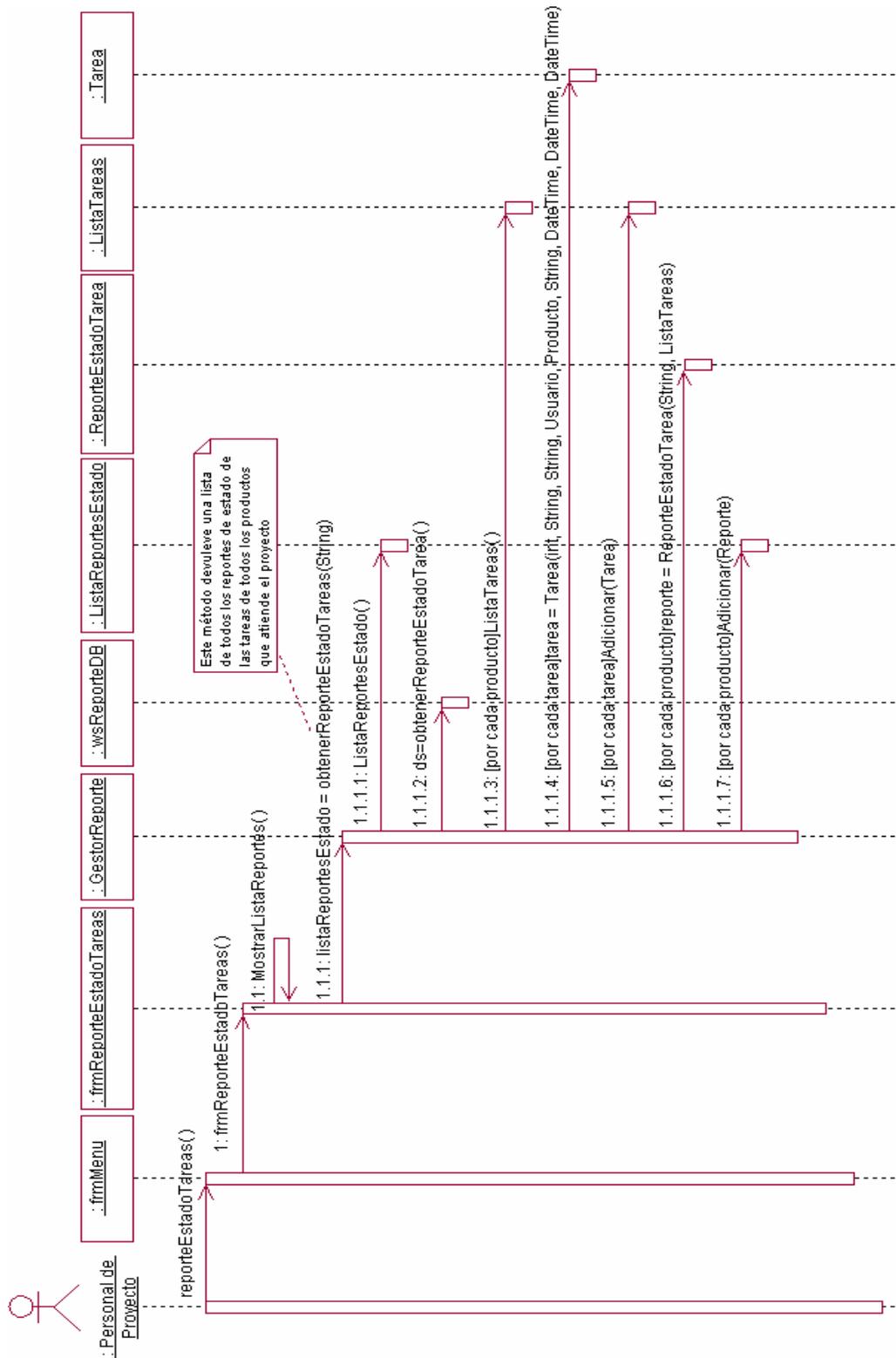


Fig. 6-19. Mostrar Información de Reportes (Sección "Reporte del estado de las tareas")

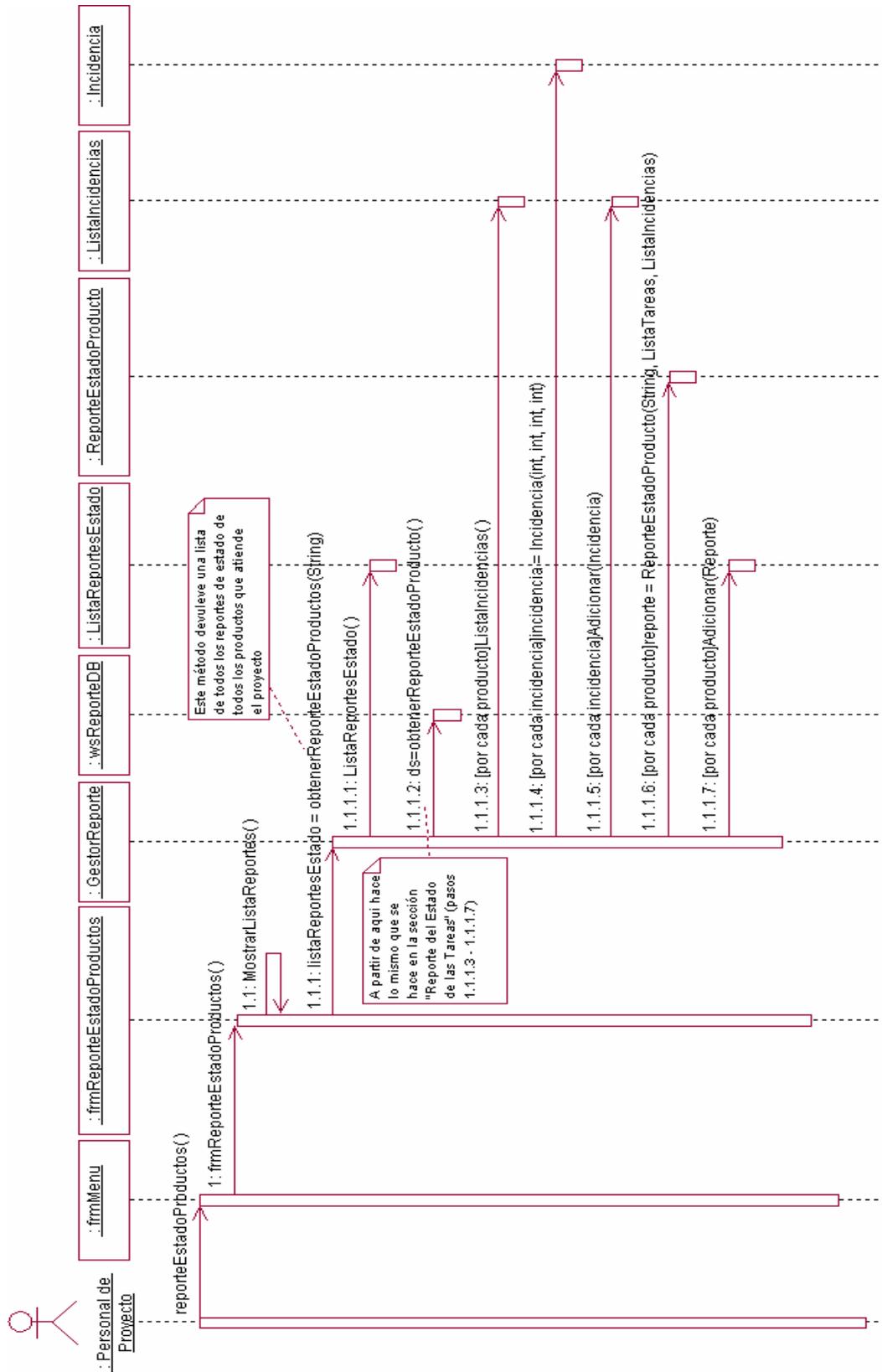


Fig. 6-20. Mostrar Información de Reportes (Sección "Reporte del estado de los productos")

Diagrama de secuencia del CU “Autenticar Usuario”

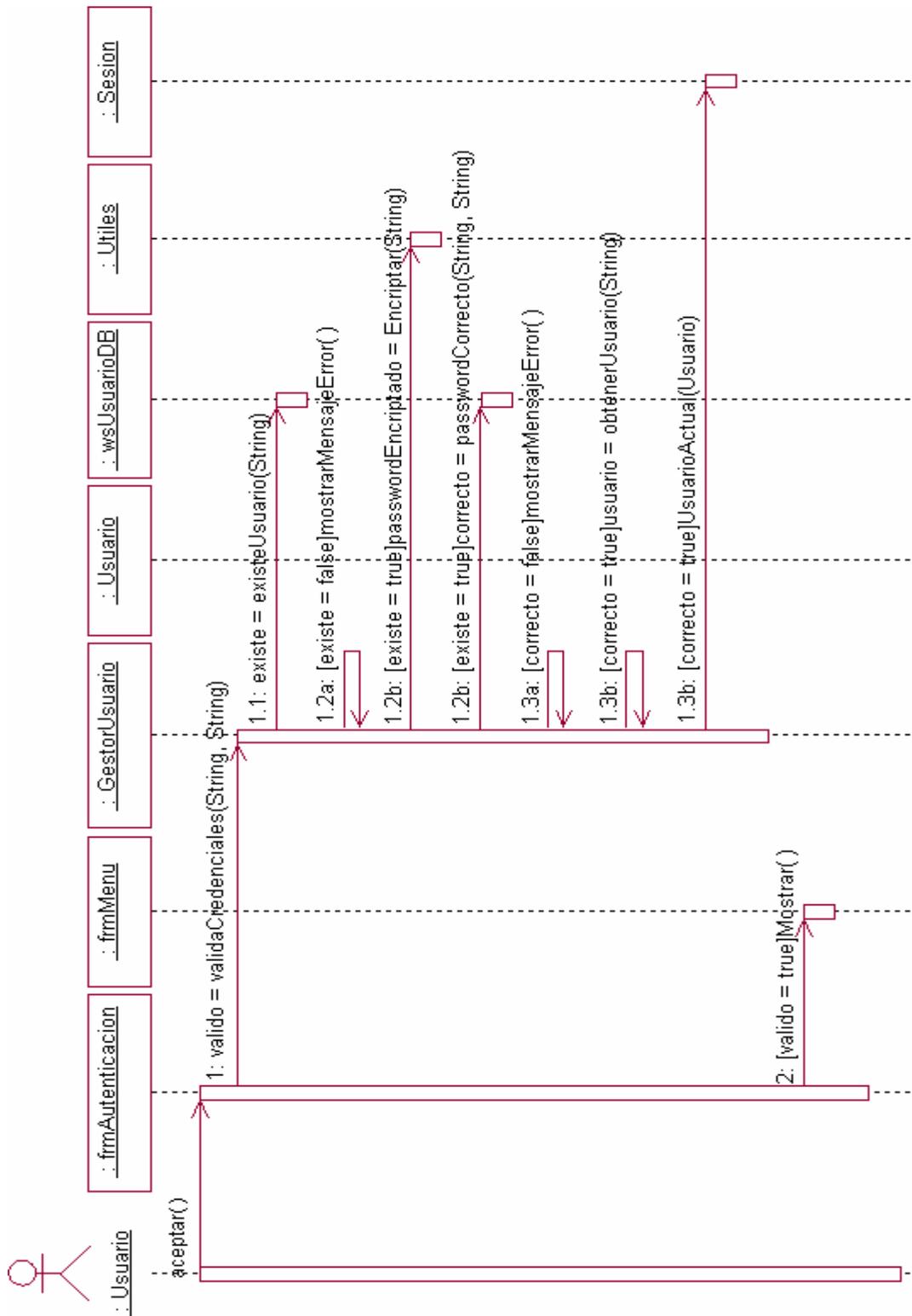


Fig.6-21. Autenticar Usuario

## Diagramas de secuencia del CU "Gestionar Ficheros del Repositorio"

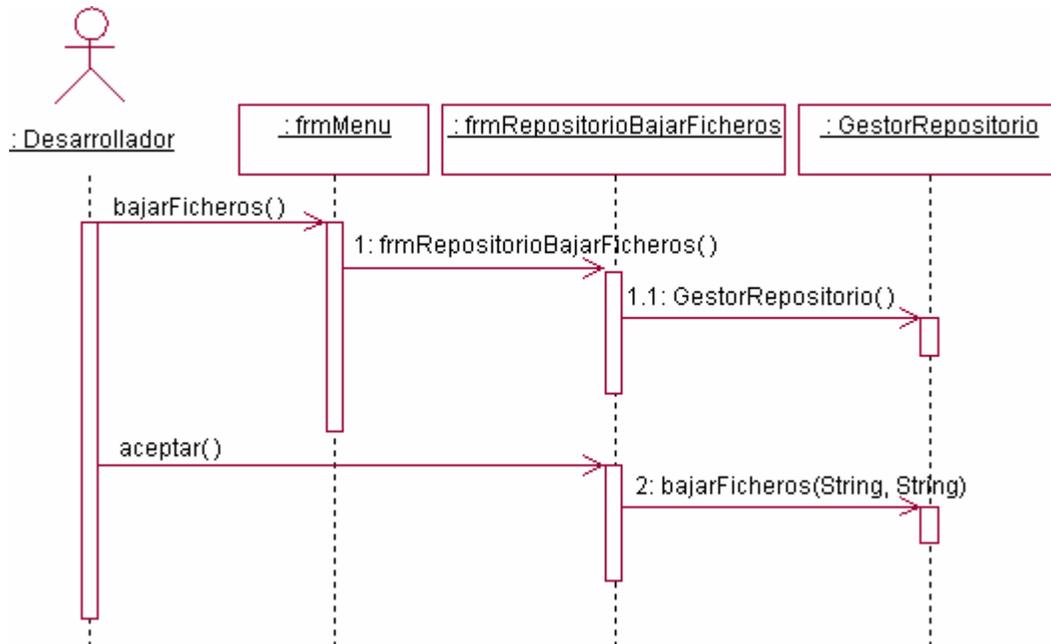


Fig.6-22. Gestionar Ficheros del Repositorio (Sección "Bajar Ficheros")

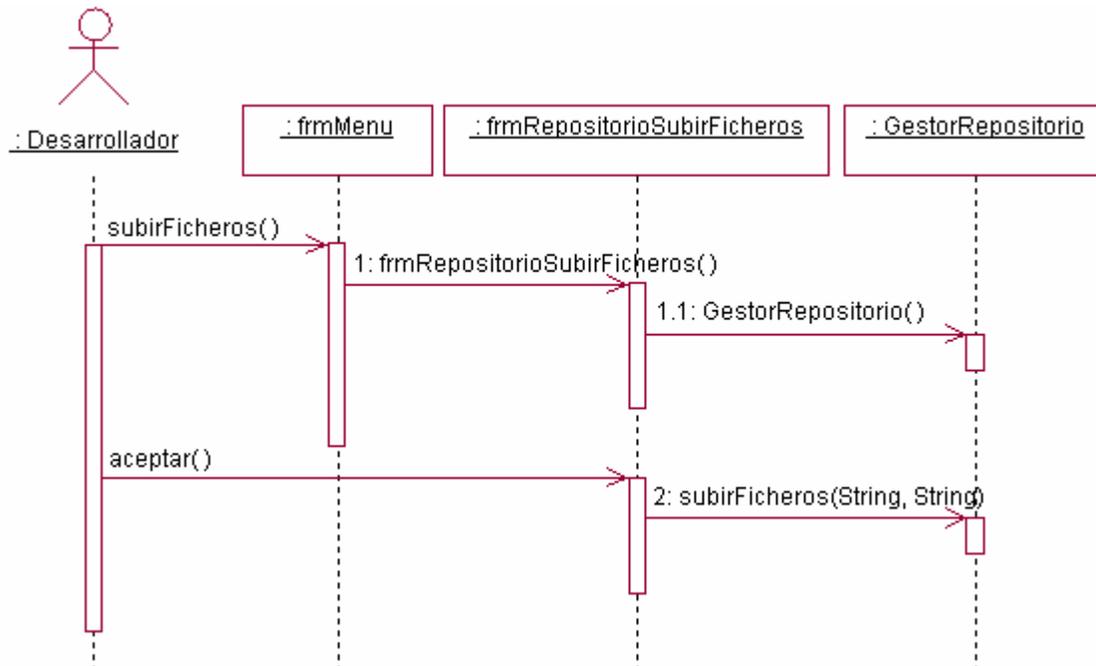


Fig. 6-23. Gestionar Ficheros del Repositorio (Sección "Subir Ficheros")

Diagramas de secuencia del CU "Administrar Grupos de Proyecto"

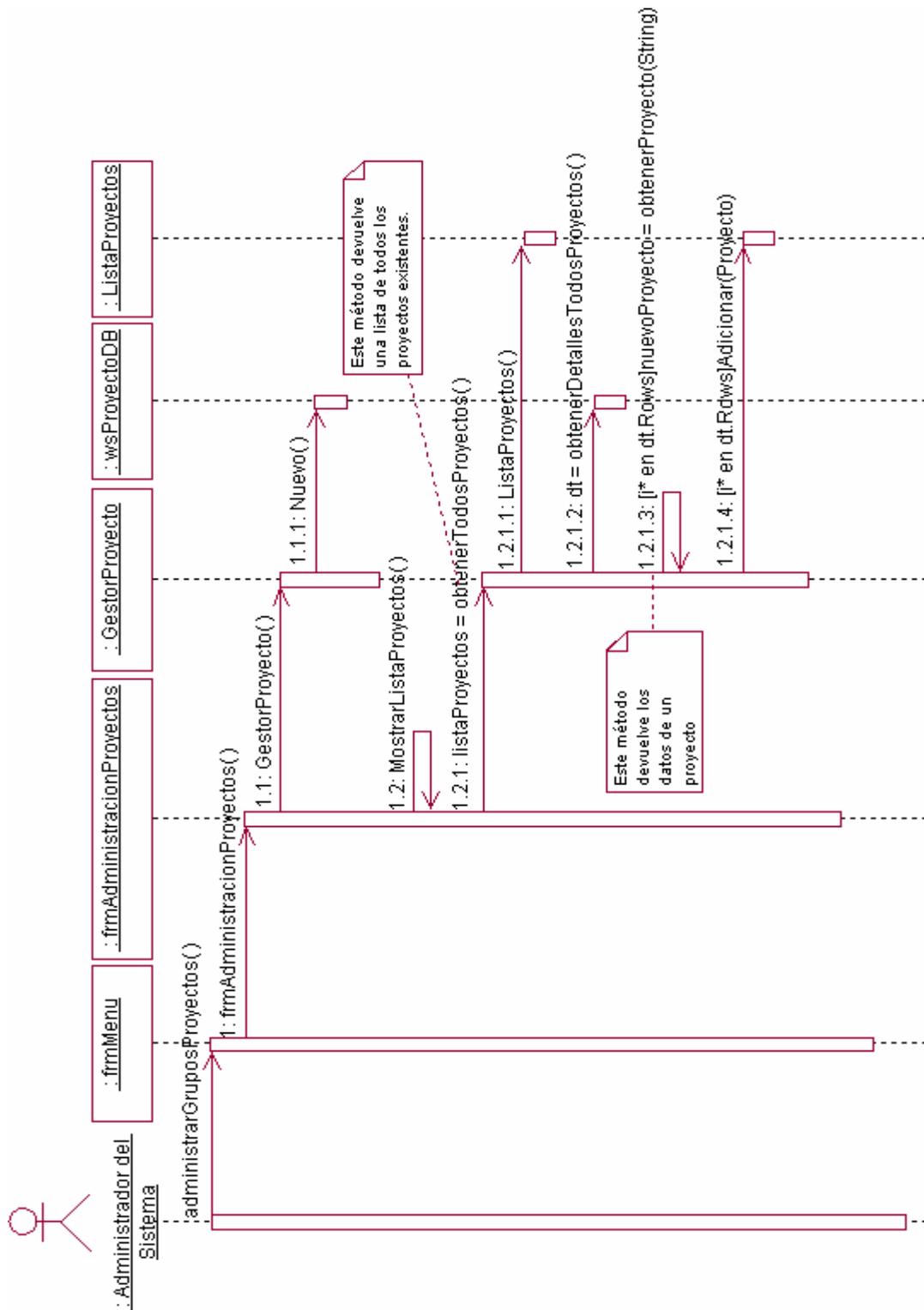


Fig. 6-24. Administrar Grupos de Proyecto (Sección "Principal")

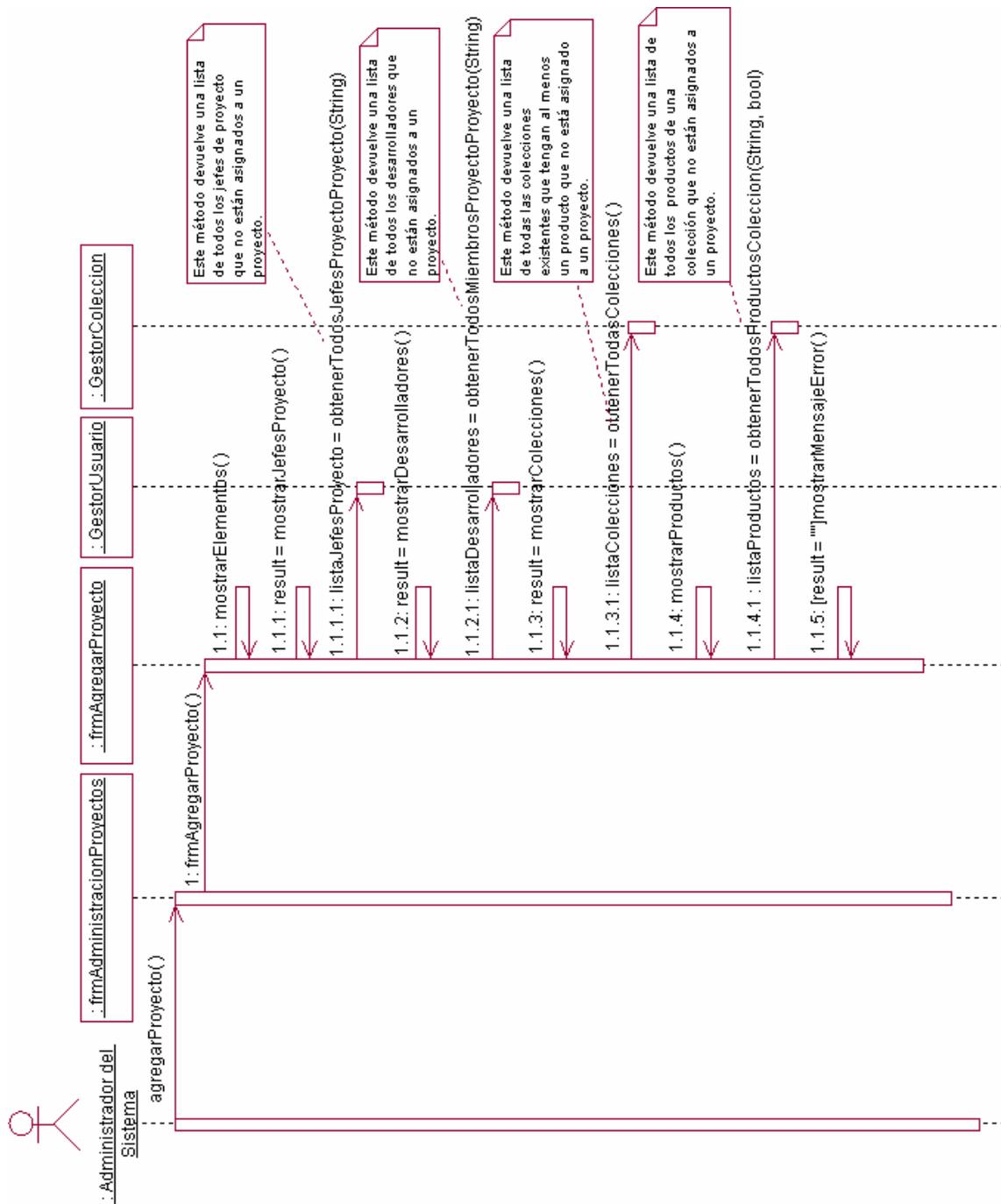


Fig. 6-25. Administrar Grupos de Proyecto (Sección "Agregar Proyecto" Escenario Mostrar Pantalla)

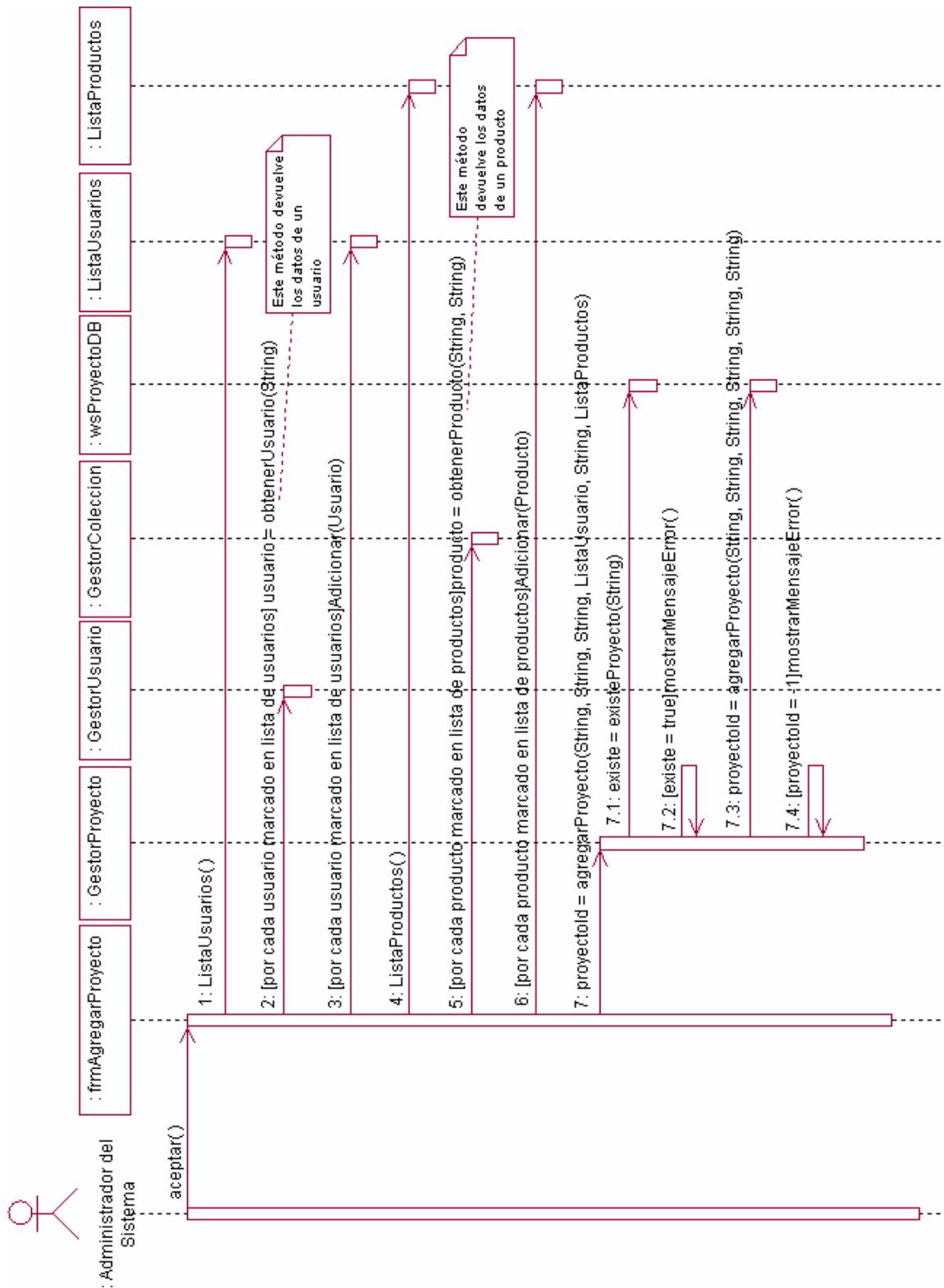
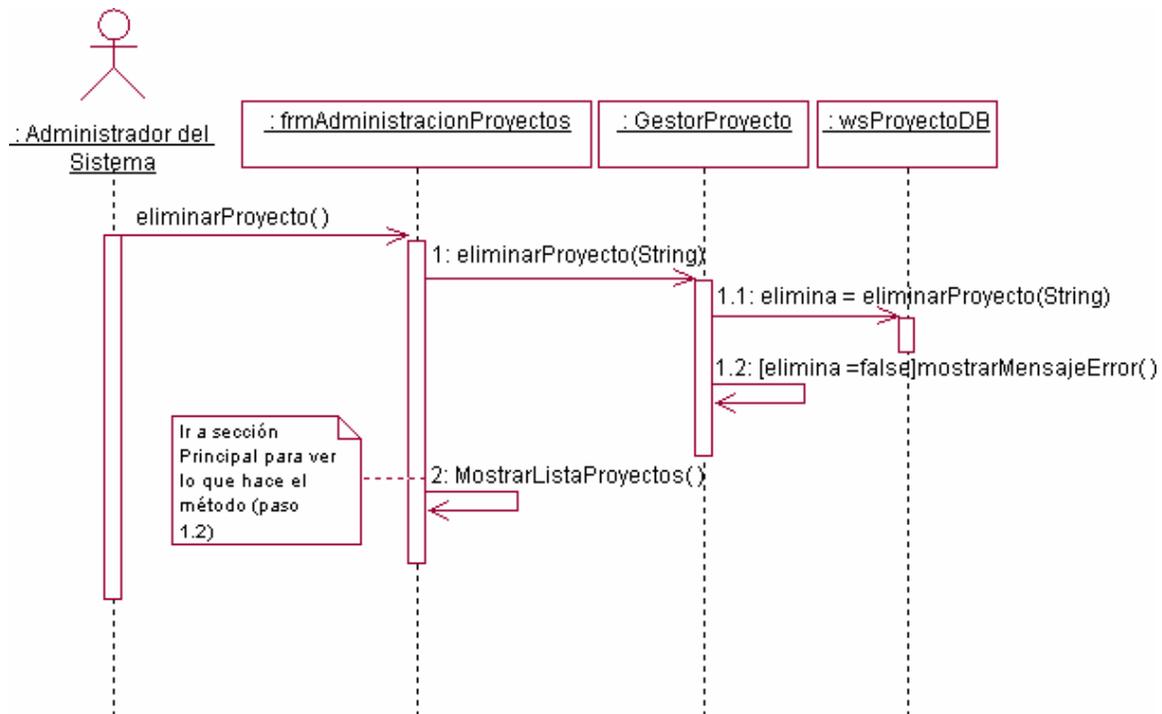
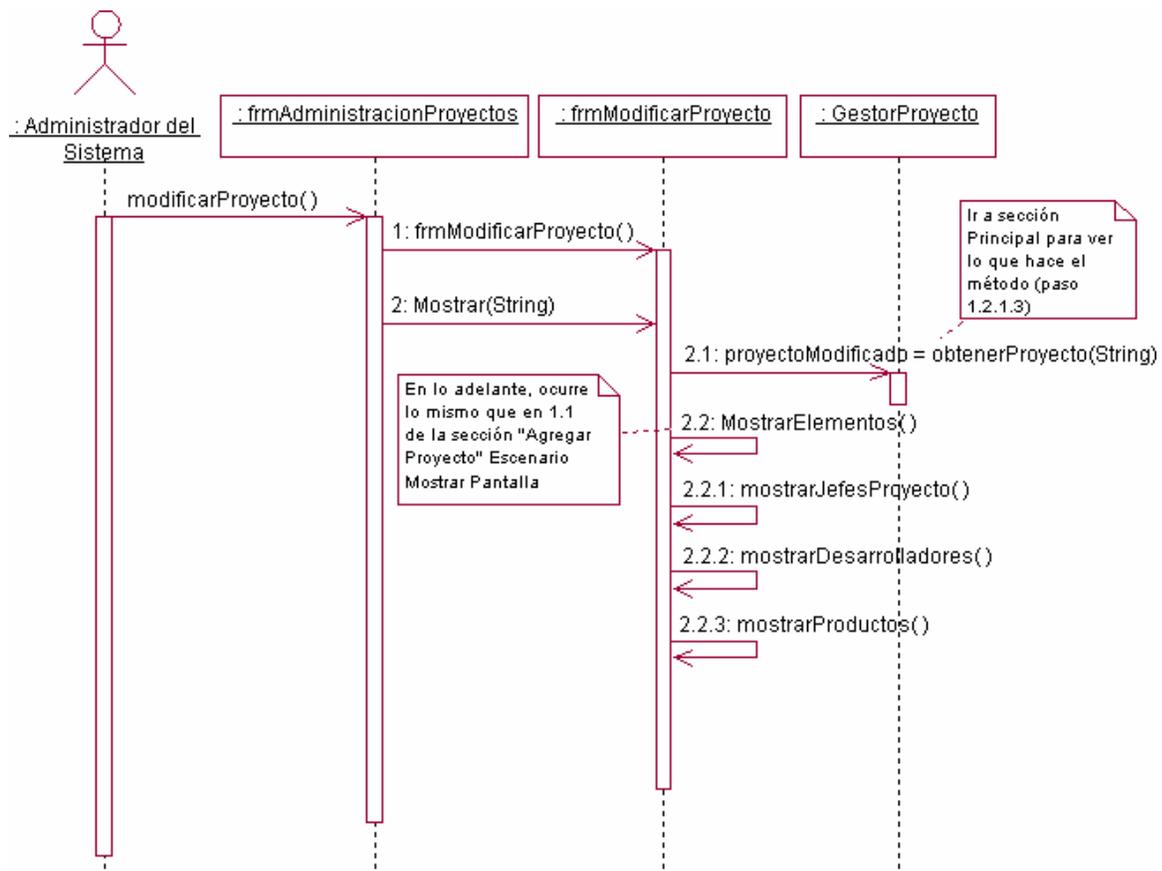


Fig. 6-26. Administrar Grupos de Proyecto (Sección "Agregar Proyecto" Escenario Agregar)



**Fig. 6-27. Administrar Grupos de Proyecto (Sección "Eliminar Proyecto")**



**Fig. 6-28. Administrar Grupos de Proyecto (Sección "Modificar Proyecto" Escenario Mostrar Pantalla)**

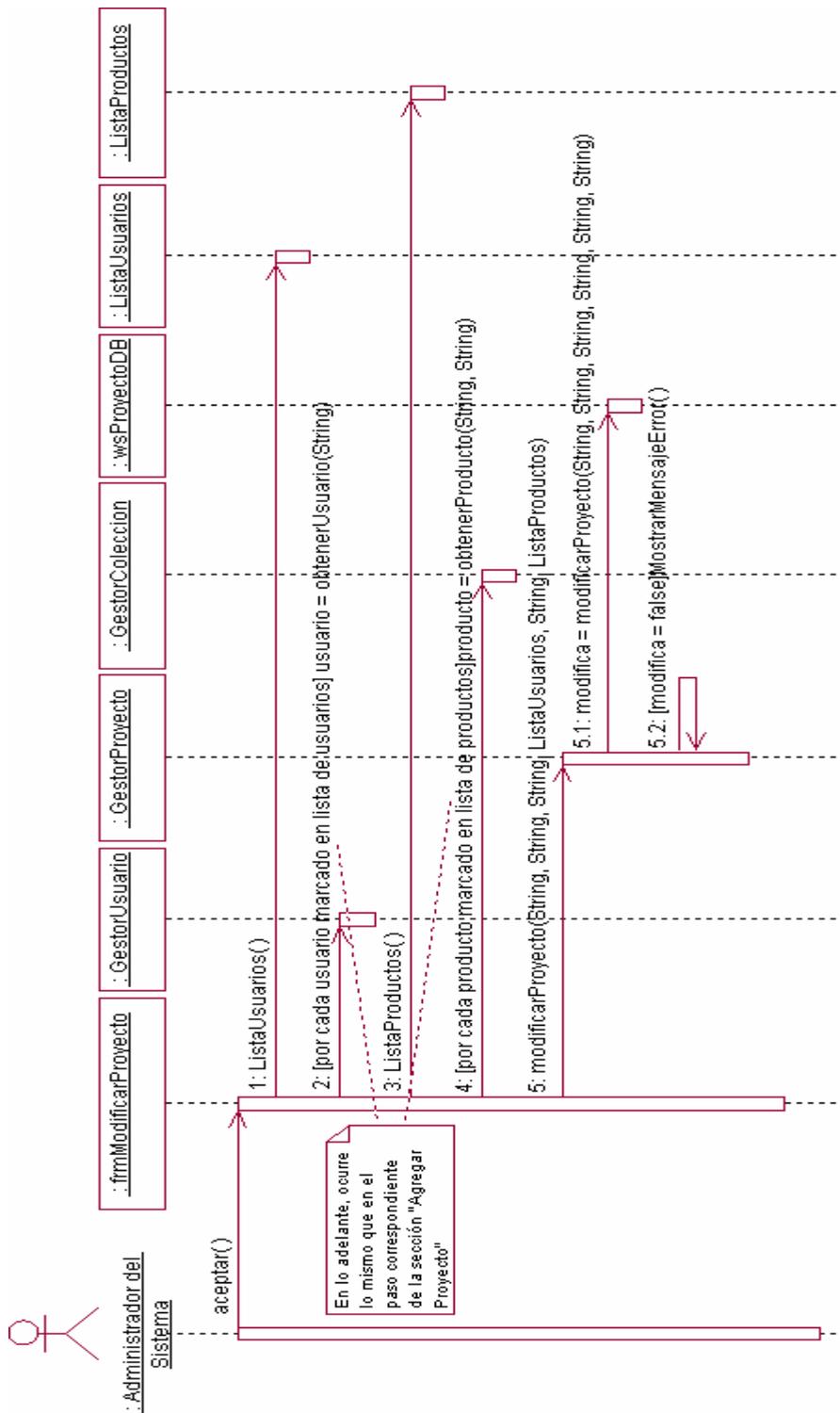


Fig. 6-29. Administrar Grupos de Proyecto (Sección "Modificar Proyecto" Escenario Modificar)

## ANEXO III. PROTOTIPOS DE INTERFAZ DE USUARIO

## Prototipos de Interfaz de usuario del CU “Realizar Tareas Asignadas”

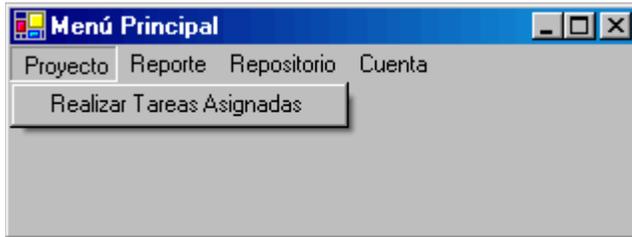


Fig. 6-30. Pantalla de Menú Desarrollador

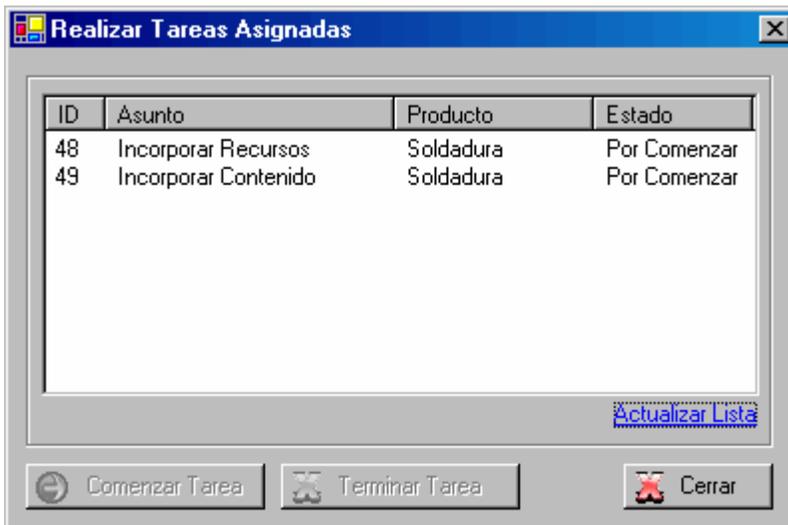


Fig. 6-31. Pantalla Realizar Tareas Asignadas

## Prototipo de Interfaz de usuario del CU “Incorporar Contenido”

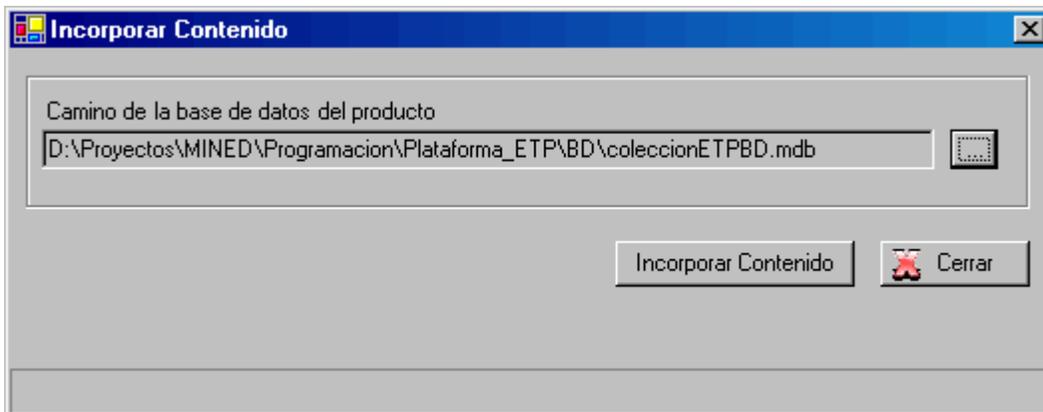
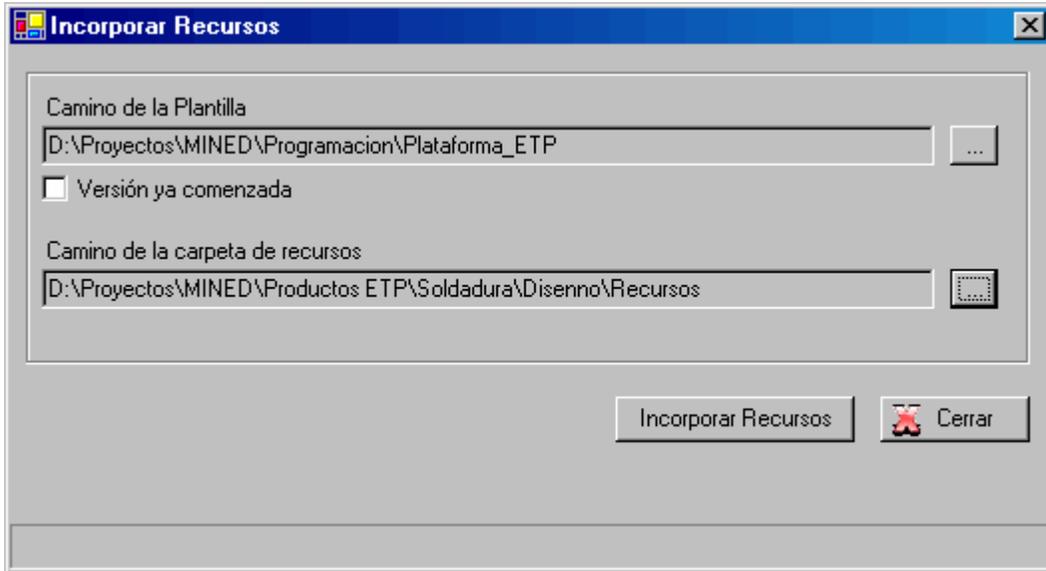
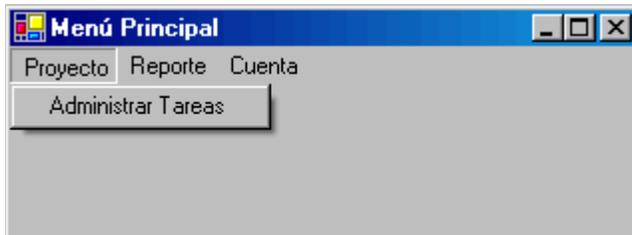
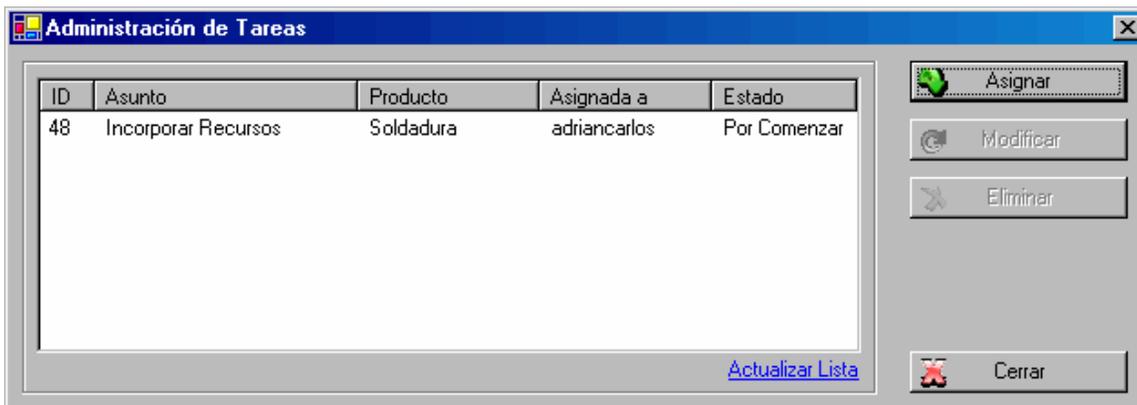
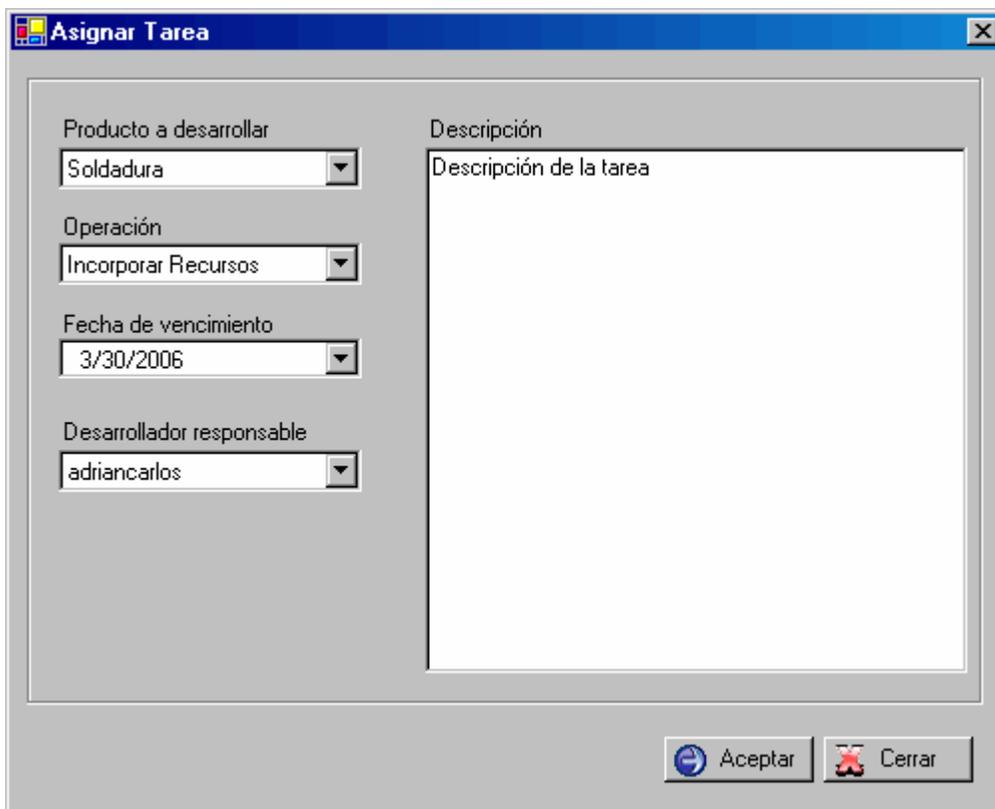


Fig. 6-32. Pantalla Incorporar Contenido

**Prototipo de Interfaz de usuario del CU “Incorporar Recursos”****Fig. 6-33. Pantalla Incorporar Recursos****Prototipos de Interfaz de usuario del CU “Administrar Tareas”****Fig. 6-34. Pantalla de Menú Jefe de Proyecto****Fig. 6-35. Pantalla de Administración de Tareas**



Asignar Tarea

Producto a desarrollar  
Soldadura

Operación  
Incorporar Recursos

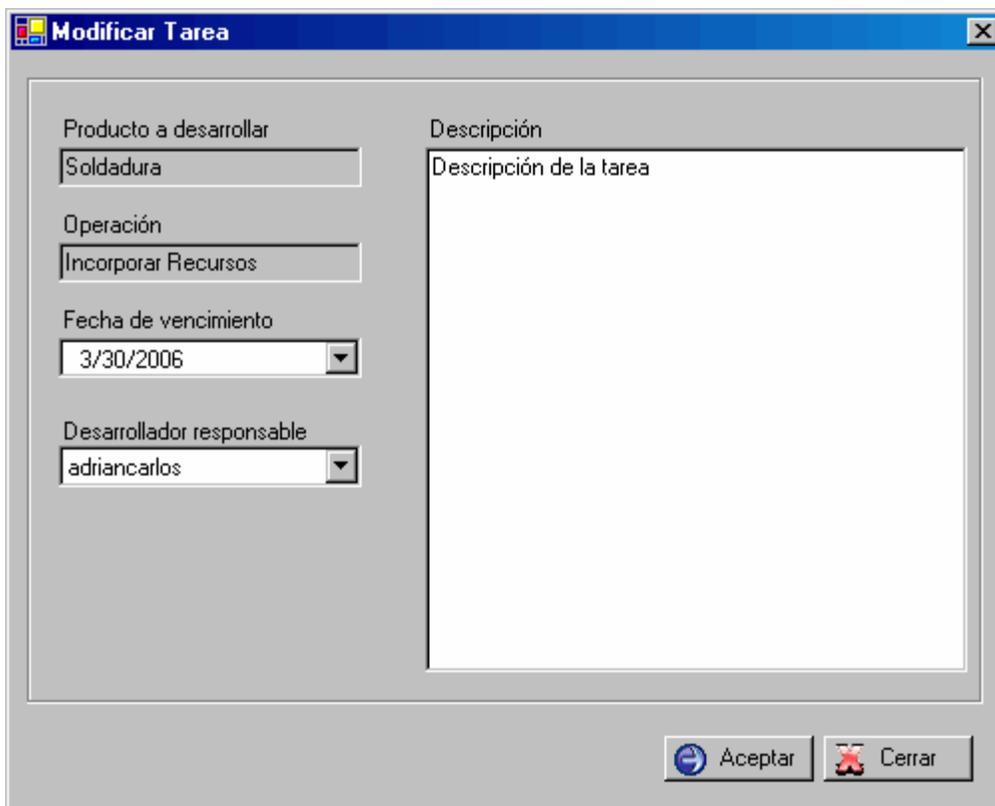
Fecha de vencimiento  
3/30/2006

Desarrollador responsable  
adriancarlos

Descripción  
Descripción de la tarea

Aceptar Cerrar

Fig. 6-36. Pantalla Asignar Tarea



Modificar Tarea

Producto a desarrollar  
Soldadura

Operación  
Incorporar Recursos

Fecha de vencimiento  
3/30/2006

Desarrollador responsable  
adriancarlos

Descripción  
Descripción de la tarea

Aceptar Cerrar

Fig. 6-37. Pantalla Modificar Tarea (Por Comenzar)

The screenshot shows a Windows-style dialog box titled "Modificar Tarea". On the left side, there are four input fields: "Producto a desarrollar" with the text "Soldadura", "Operación" with "Incorporar Recursos", "Fecha de vencimiento" with a dropdown menu showing "3/30/2006", and "Desarrollador responsable" with a dropdown menu showing "adriancarlos". On the right side, there is a large text area labeled "Descripción" containing the text "Descripción de la tarea". At the bottom right, there are two buttons: "Aceptar" (Accept) and "Cerrar" (Close).

Fig. 6-38. Pantalla Modificar Tarea (En Desarrollo y Reasignada)

This screenshot is identical to the previous one, but with an additional checkbox at the bottom left labeled "Reasignar la tarea" (Reassign the task), which is currently unchecked. The rest of the interface, including the input fields and buttons, remains the same.

Fig. 6-39. Pantalla Modificar Tarea (Terminada)

Prototipos de Interfaz de usuario del CU “Administrar Grupos de Proyecto”

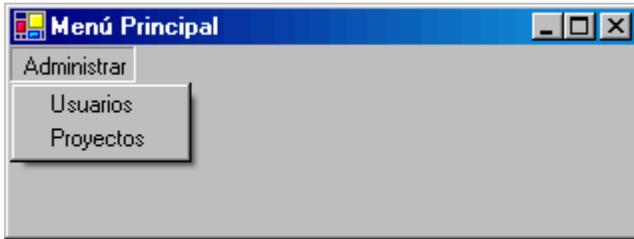


Fig. 6-40. Pantalla de Menú Administrador

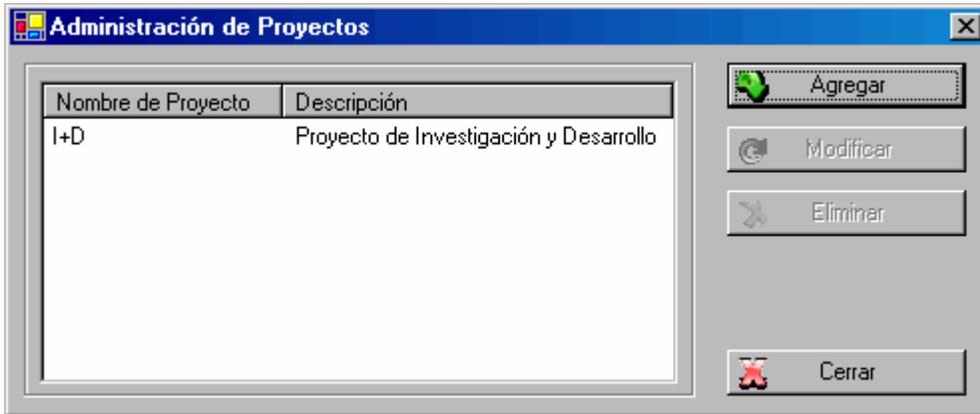


Fig. 6-41. Pantalla de Administración de Proyectos

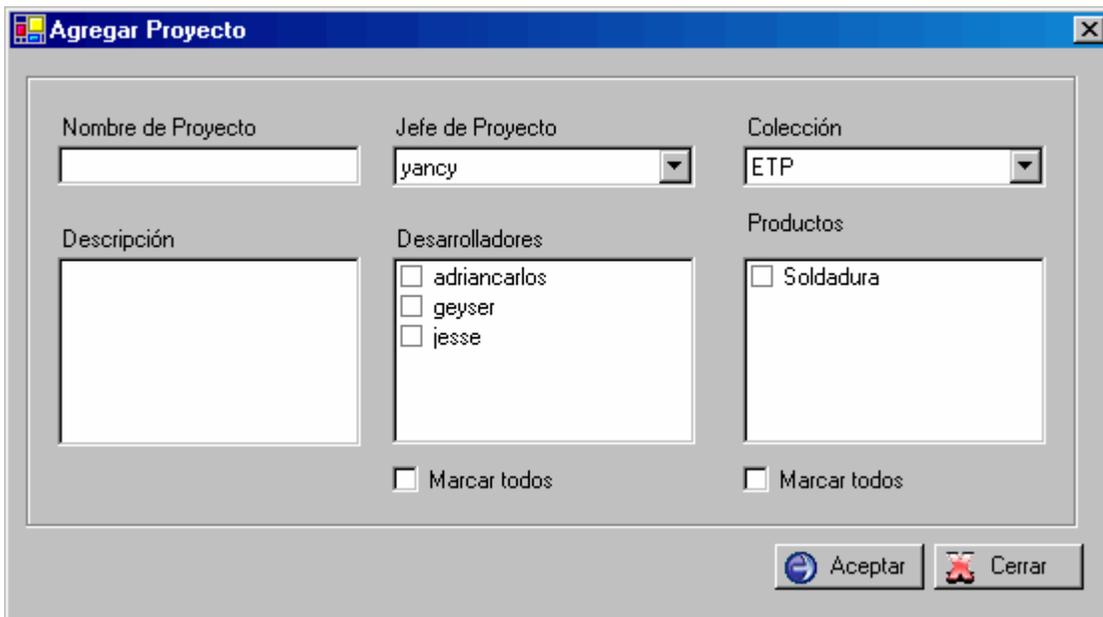


Fig. 6-42. Pantalla Agregar Proyecto

**Modificar Proyecto**

Nombre de Proyecto	Jefe de Proyecto	Colección
I+D	yancy	ETP
Descripción	Desarrolladores	Productos
Proyecto de Investigación y Desarrollo	<input checked="" type="checkbox"/> adriancarlos <input checked="" type="checkbox"/> geysler <input checked="" type="checkbox"/> jesse	<input checked="" type="checkbox"/> Soldadura

Aceptar Cerrar

**Fig. 6-43. Pantalla Modificar Proyecto**

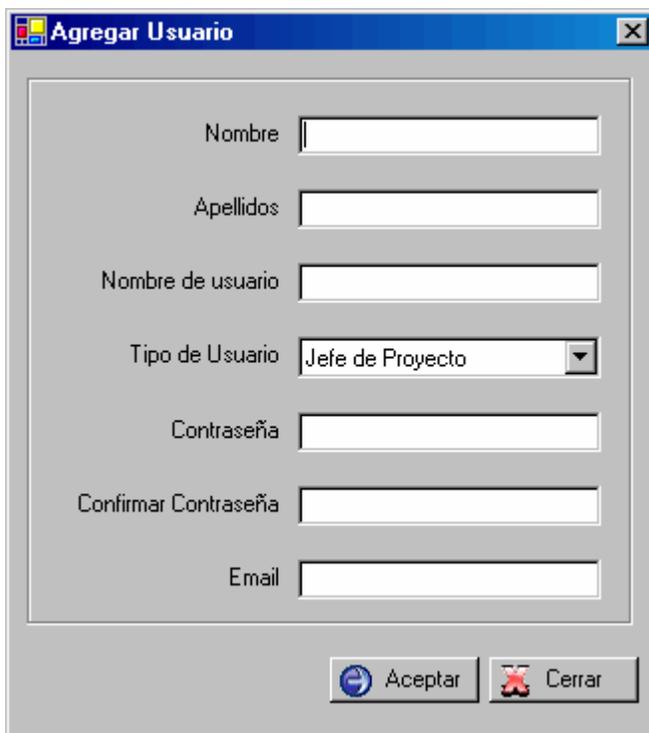
### Prototipos de Interfaz de usuario del CU “Administrar Usuarios”

**Administración de Usuarios**

Nombre de Usuario	Tipo de Usuario
adriancarlos	Desarrollador
alexey	Jefe de Proyecto
edier	Administrador
geysler	Desarrollador
jesse	Desarrollador
yancy	Jefe de Proyecto

Agregar  
Modificar  
Eliminar  
Cerrar

**Fig. 6-44. Pantalla de Administración de Usuarios**



**Agregar Usuario**

Nombre

Apellidos

Nombre de usuario

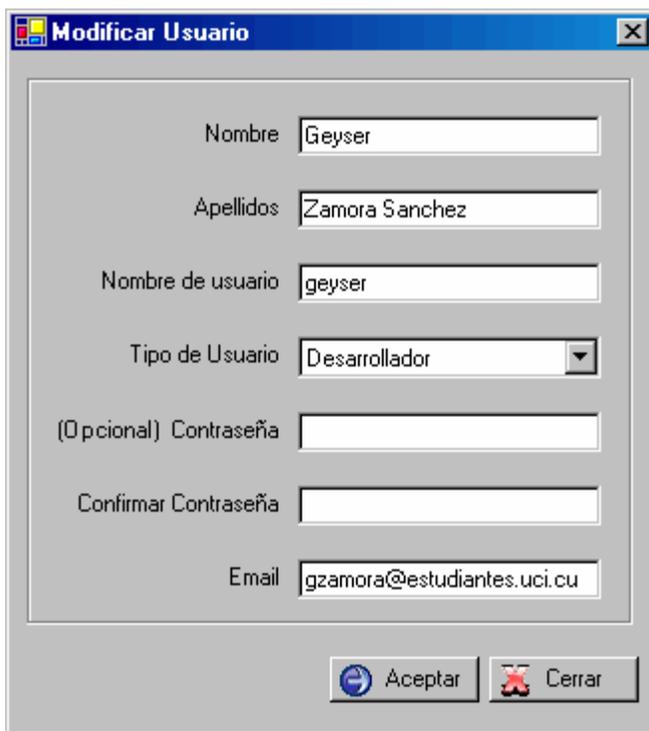
Tipo de Usuario

Contraseña

Confirmar Contraseña

Email

**Fig. 6-45. Pantalla Agregar Usuario**



**Modificar Usuario**

Nombre

Apellidos

Nombre de usuario

Tipo de Usuario

(Opcional) Contraseña

Confirmar Contraseña

Email

**Fig. 6-46. Pantalla Modificar Usuario**

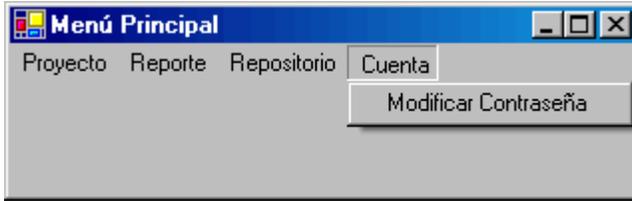


Fig. 6-47. Pantalla de Menú Personal de Proyecto

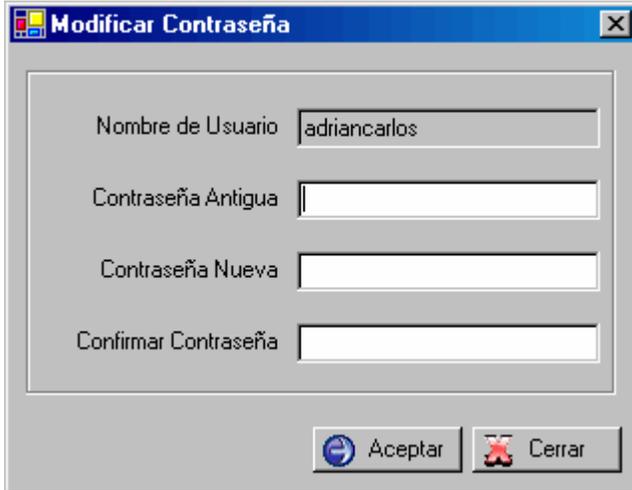


Fig. 6-48. Pantalla Modificar Contraseña

Prototipos de Interfaz de usuario del CU “Mostrar Información de Reporte”

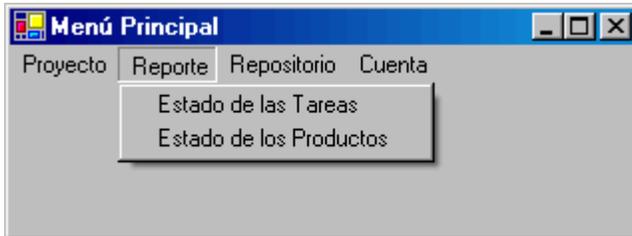


Fig. 6-49. Pantalla Menú Reporte

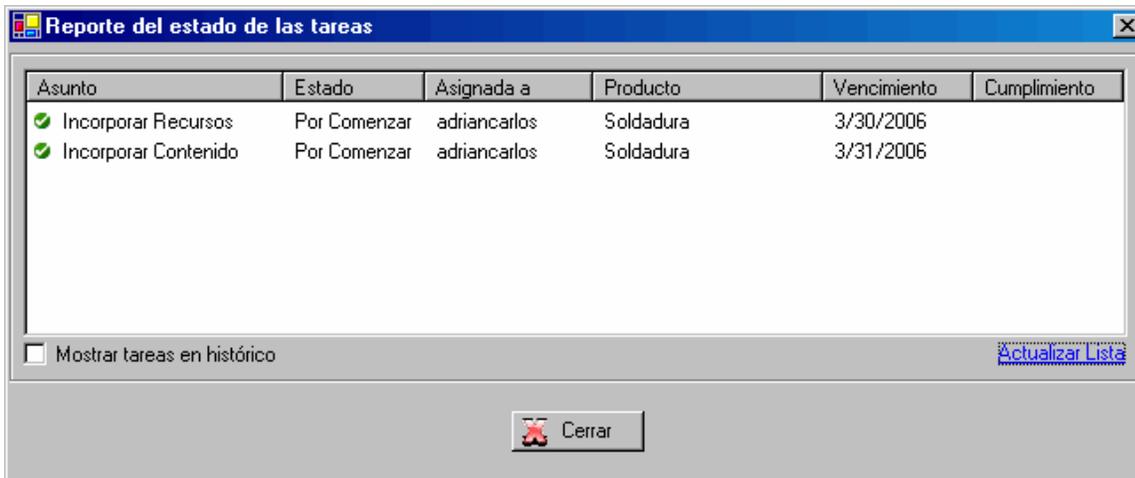


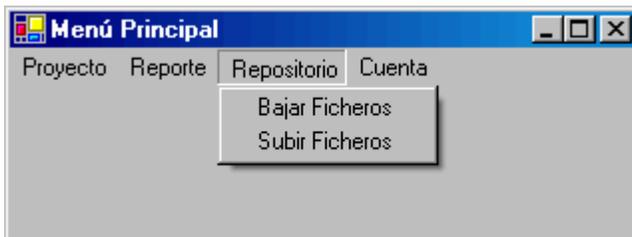
Fig. 6-50. Pantalla Reporte del estado de las tareas

**Prototipo de Interfaz de usuario del CU “Autenticar Usuario”**

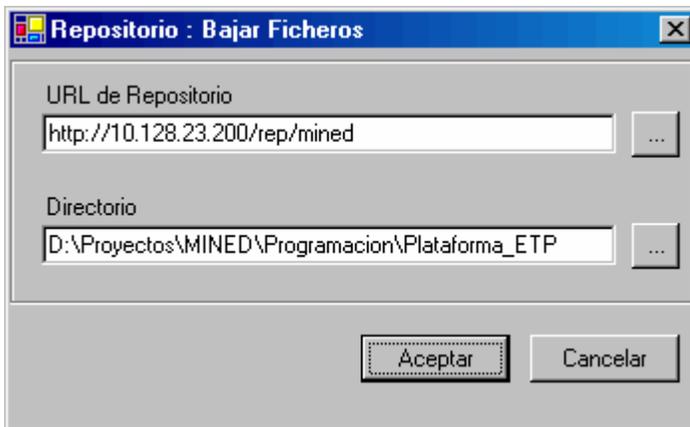


**Fig. 6-51. Pantalla de Autenticación**

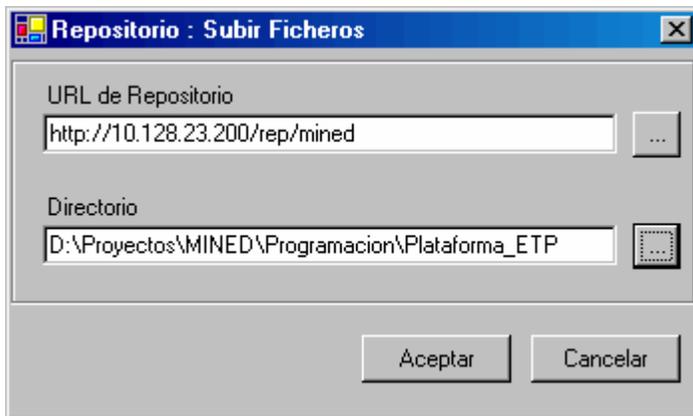
**Prototipos de Interfaz de usuario del CU “Gestionar Ficheros del Repositorio”**



**Fig. 6-52. Pantalla de Menú Repositorio**



**Fig. 6-53. Pantalla Repositorio Bajar Ficheros**



**Fig. 6-54. Pantalla Repositorio Subir Ficheros**

## ANEXO IV. DESCRIPCIÓN DE LOS PRICIPALES COMPONENTES

Tabla 6-1. Descripción de los principales componentes

Componente	Propósito	Contenido
frmAutenticacion.cs	Ventana que permiten la autenticación del usuario en el sistema.	frmAutenticacion
frmMenu.cs	Ventana desde la cuál se accederá a cada una de las opciones que tiene el usuario.	frmMenu
frmAdministracionTareas.cs	Mostrar las tareas existentes y permitir el acceso a los formularios de agregar y modificar tarea.	frmAdministracionTareas
frmAgregarTarea.cs	Agrupar a todos los controles visuales que permiten asignar tareas.	frmAgregarTarea
frmModificarTarea.cs	Agrupar a todos los controles visuales que permiten modificar tareas.	frmModificarTarea
frmRealizarTareasAsignadas.cs	Mostrar las tareas que tiene asignadas un usuario y permite la selección de una de ellas para su realización.	frmRealizarTareasAsignadas
frmIncorporarRecursos.cs	Agrupar a todos los controles visuales que permiten incorporar recursos a la plantilla	frmIncorporarRecursos
frmIncorporarContenido.cs	Agrupar a todos los controles visuales que permiten incorporar contenido a la plantilla.	frmIncorporarContenido
data_plantilla_coleccion.xml	Representa un XML donde se almacena la estructura de ficheros de la plantilla.	
Tarea.cs	Guardar la información de las tareas existentes.	Tarea ListaTareas

## ANEXOS

GestorTarea.cs	Controlar la gestión de las tareas que han sido asignadas	GestorTarea
GestorContenido.cs	Controlar la gestión del contenido que es incorporado a la base de datos de la plantilla.	GestorContenido
Contenido.cs	Controlar el proceso de obtención del contenido y su posterior incorporación en la base de datos de la plantilla.	Contenido
Reporte_coleccion_producto.xml	Representa un XML donde se almacenan las incidencias que hubo en la incorporación de recursos de un producto determinado.	
IncorporarRecursos.exe	Controlar el proceso de incorporación de recursos a la plantilla.	CC_CastLib CC_Plantilla CC_Reportes CCM_Iniciar CCDatos_Plantilla CCDatos_Recursos
Paths.xml	Representa un XML donde se almacenan los datos referentes a la incorporación de recursos de un producto. Guarda los caminos a las carpetas donde se encuentran la plantilla y los recursos que serán incorporados a esta.	
OleDbHelper.cs	Permite la gestión de los datos en la base de datos de la plantilla.	OleDbHelper
SqlHelper.cs	Permite la gestión de los datos en la base de datos del sistema.	SqlHelper SqlHelperParameterCache

## ANEXOS

---

contenidodb.asmx	Brindar servicios para la gestión de datos de los contenidos que serán incorporados.	contenidodb
tareadb.asmx	Brindar servicios para la gestión de datos de las tareas que han sido asignadas.	tareadb

## ANEXO V. FLUJO DE PROCESOS

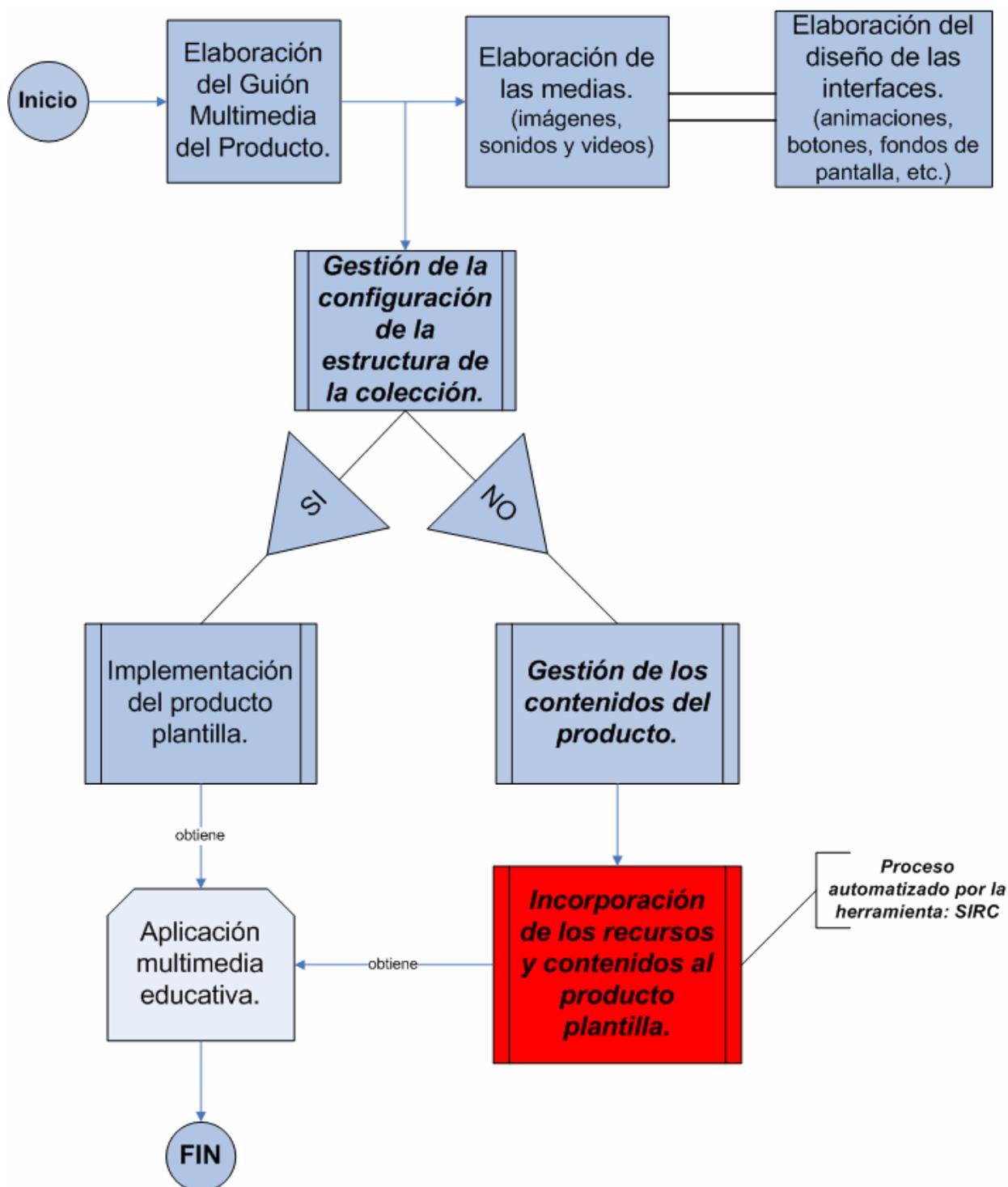


Fig. 6-55. Diagrama de flujo de procesos para la creación de productos de una colección

ANEXO VI. ARQUITECTURA DEL SISME

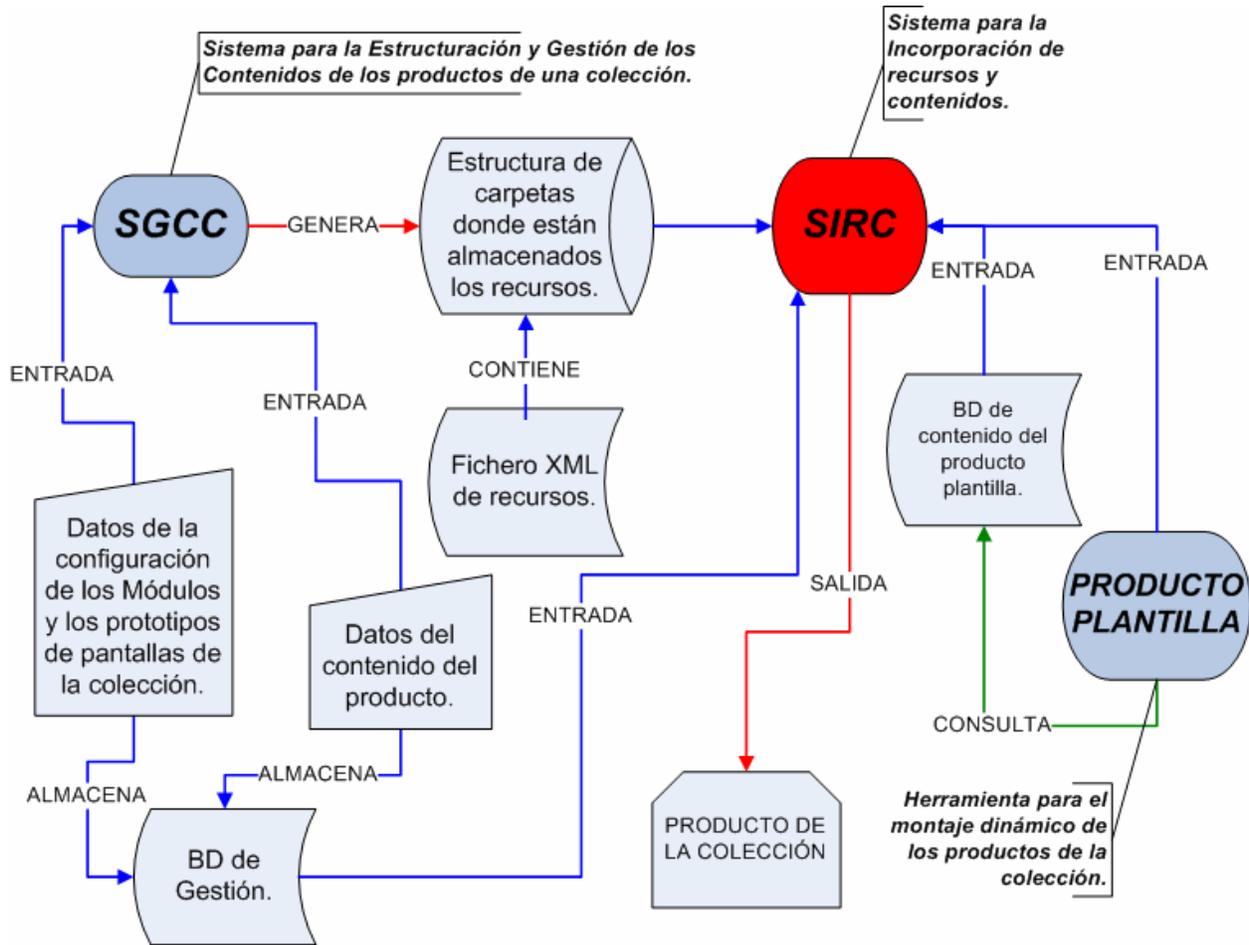


Fig. 6-56. Arquitectura del Sistema de Multimedia Educativa (SISME)