



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**SISTEMA PARA EL CONTROL DE TRABAJOS DE GRADUACIÓN, DE LA
ESCUELA DE CIENCIAS Y SISTEMAS, DE LA FACULTAD DE INGENIERÍA**

Victor Hugo Zamora Herrarte

Asesorado por la Inga. Elizabeth Domínguez Alvarado

Guatemala, mayo de 2008

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**SISTEMA PARA EL CONTROL DE TRABAJOS DE GRADUACIÓN, DE LA
ESCUELA DE CIENCIAS Y SISTEMAS, DE LA FACULTAD DE INGENIERÍA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

VICTOR HUGO ZAMORA HERRARTE

ASESORADO POR LA INGA. ELIZABETH DOMÍNGUEZ ALVARADO

AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, MAYO DE 2008

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympto Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soria
VOCAL II	Inga. Alba Guerrero de López
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. Kenneth Issur Estrada Ruiz
VOCAL V	
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympto Paiz Recinos
EXAMINADOR	Ing. Alfredo Valdés Matta
EXAMINADOR	Ing. Freiry Javier Gramajo López
EXAMINADOR	Ing. Jorge Armín Mazariegos Rabanales
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

SISTEMA PARA EL CONTROL DE TRABAJOS DE GRADUACIÓN, DE LA ESCUELA DE CIENCIAS Y SISTEMAS, DE LA FACULTAD DE INGENIERÍA,

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, en enero de 2007.

Victor Hugo Zamora Herrarte

AGRADECIMIENTOS A:

- DIOS** Por darme la sabiduría y entendimiento para poder llegar a este día.
- Mi madre** Por apoyarme incondicionalmente para alcanzar mis metas, darme ánimo en los momentos difíciles y brindarme todo su cariño, consejo y comprensión.
- Mi padre (†)** Por proporcionarme todos los recursos necesarios para alcanzar este logro.
- Mi esposa** Por estar siempre a mi lado en todo momento, y haberme brindado su amor incondicional y paciencia.
- Mis amigos** Por su leal y sincera amistad en todos los momentos que compartimos.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	VII
GLOSARIO	XIII
RESUMEN	XVII
OBJETIVOS	XIX
INTRODUCCIÓN	XXI
1. FUNDAMENTOS BÁSICOS	1
1.1. Introducción a UML	1
1.1.1. Casos de uso	1
1.1.2. Diagrama de secuencias	4
1.1.3. Diagrama de actividades	6
1.1.4. Diagrama de componentes	10
1.1.5. Diagrama de distribución	11
1.2. Introducción a bases de datos	13
1.2.1. Modelo entidad relación	14
1.2.2. Diagrama entidad relación	15
1.3. Modelos de arquitectura de <i>software</i>	18
1.3.1. Arquitectura de 1 capa	19
1.3.2. Arquitectura de 2 capas	19
1.3.3. Arquitectura de 3 capas	20
1.3.4. Arquitectura flexible de <i>n</i> capas	21
2. FASE DE ANÁLISIS	23
2.1. Descripción del sistema actual	23
2.2. Definición del problema	25
2.3. Requerimientos	27
2.4. Propuesta	29

2.4.1.	Módulo de consultas	29
2.4.2.	Módulo de administración	30
3.	FASE DE DISEÑO	33
3.1.	Diseño de la base de datos	33
3.1.1.	Entidades	33
3.1.2.	Diagrama entidad relación	36
3.2.	Utilización de UML	37
3.2.1.	Módulo de consultas	37
3.2.1.1.	Diagrama de casos de uso de consultas	37
3.2.1.2.	Diagrama de secuencias de consultas	38
3.2.1.3.	Diagrama de actividades de consultas	40
3.2.2.	Módulo de administración	41
3.2.2.1.	Diagrama de casos de uso de mantenimiento	42
3.2.2.1.1.	Diagrama de secuencias de crear en mantenimiento	43
3.2.2.1.2.	Diagrama de actividades de crear en mantenimiento	45
3.2.2.1.3.	Diagrama de secuencias de eliminar en mantenimiento	46
3.2.2.1.4.	Diagrama de actividades de eliminar en mantenimiento	48
3.2.2.1.5.	Diagrama de secuencias de actualizar en mantenimiento	49
3.2.2.1.6.	Diagrama de actividades de actualizar en mantenimiento	51
3.2.2.2.	Diagrama de casos de uso de usuarios	52
3.2.2.2.1.	Diagrama de secuencias de crear usuario	53
3.2.2.2.2.	Diagrama de actividades de crear usuario	55
3.2.2.2.3.	Diagrama de secuencias de eliminar usuario	56
3.2.2.2.4.	Diagrama de actividades de eliminar usuario	58
3.2.2.2.5.	Diagrama de secuencias de actualizar usuario	59
3.2.2.2.6.	Diagrama de actividades de actualizar usuario	61
3.2.2.2.7.	Diagrama de secuencias de asignación de usuarios	62
3.2.2.2.8.	Diagrama de actividades de asignación de usuarios	64
3.2.3.	Diagrama de distribución	65

4. MANUALES	67
4.1. Manual de usuario	67
4.1.1. Iniciar sesión	67
4.1.1.1. ¿Olvidó su contraseña?	68
4.1.2. Modificar cuenta.....	69
4.1.2.1. Editar mis datos personales.....	70
4.1.2.2. Cambiar mi contraseña.....	71
4.1.2.3. Cambiar pregunta y respuesta de seguridad.....	71
4.1.3. Cerrar sesión	72
4.1.4. Rol digitador.....	73
4.1.4.1. Crear asesores	73
4.1.4.2. Crear estudiantes.....	75
4.1.4.3. Cargar estudiantes.....	76
4.1.4.4. Crear documentos.....	78
4.1.5. Rol tutor	81
4.1.5.1. Administrar asesores	82
4.1.5.2. Administrar estudiantes.....	83
4.1.5.3. Administrar documentos	88
4.1.6. Rol administrador.....	91
4.1.6.1. Crear usuarios	91
4.1.6.2. Administrar usuarios	93
4.1.6.3. Administrar roles	94
4.1.6.4. Administrar estados	96
4.1.6.5. Administrar estudiantes (rol administrador)	97
4.1.6.6. Administrar documentos (rol administrador)	99
4.1.7. Consultas	100
4.1.7.1. Menú asesores	100
4.1.7.2. Menú protocolos.....	102
4.1.7.3. Menú documentos no finalizados.....	103
4.1.7.4. Menú documentos finalizados.....	103
4.2. Manual técnico.....	103
4.2.1. Tecnología utilizada	104

4.2.2.	Implementación de la base de datos	107
4.2.2.1.	Tablas	109
4.2.2.1.1.	Tabla <i>aspnet_Users</i>	109
4.2.2.1.2.	Tabla <i>aspnet_Membership</i>	110
4.2.2.1.3.	Tabla <i>aspnet_Roles</i>	112
4.2.2.1.4.	Tabla <i>aspnet_UsersInRoles</i>	112
4.2.2.1.5.	Tabla <i>asesor</i>	113
4.2.2.1.6.	Tabla <i>estudiante</i>	114
4.2.2.1.7.	Tabla <i>asignación</i>	115
4.2.2.1.8.	Tabla <i>estado</i>	116
4.2.2.1.9.	Tabla <i>documento</i>	117
4.2.2.1.10.	Tabla <i>historial</i>	118
4.2.2.2.	Procedimientos almacenados (<i>stored procedures</i>)	120
4.2.2.3.	Información predeterminada de la base de datos.....	122
4.2.2.3.1.	Roles predeterminados	123
4.2.2.3.2.	Usuarios predeterminados	123
4.2.2.3.3.	Estados predeterminados	124
4.2.3.	Implementación de la aplicación.....	125
4.2.3.1.	Capa de datos.....	126
4.2.3.2.	Capa de acceso a datos	126
4.2.3.3.	Capa de reglas del negocio	128
4.2.3.3.1.	Clase <i>AppUser</i>	128
4.2.3.3.2.	Clase <i>AppAdviser</i>	129
4.2.3.3.3.	Clase <i>AppStudent</i>	130
4.2.3.3.4.	Clase <i>AppDocument</i>	131
4.2.3.4.	Capa de interfaz del usuario	132
4.2.3.4.1.	Componentes adicionales	133
4.2.3.4.2.	Controles de usuario	135
4.2.3.4.3.	Páginas <i>Web</i>	136
4.2.3.5.	Configuración.....	140
4.2.3.5.1.	Archivo <i>web.config</i>	140
4.2.3.5.2.	Archivo <i>web.sitemap</i>	153

4.3.	Manual de instalación	154
4.3.1.	Requisitos mínimos.....	154
4.3.2.	Instalación con asistente.....	155
4.3.3.	Instalación manual	159
4.3.4.	Configuración de la aplicación	167
CONCLUSIONES		169
RECOMENDACIONES		171
BIBLIOGRAFÍA		173

ÍNDICE DE ILUSTRACIONES

FIGURAS

1. Actor	2
2. Caso de uso.....	2
3. Tipos de relaciones.....	3
4. Ejemplo de un diagrama de casos de uso	3
5. Línea de vida	5
6. Tipos de mensajes.....	5
7. Ejemplo de un diagrama de secuencias	6
8. Estado de acción	7
9. Estados inicial y final.....	7
10. Ejemplo de un estado de decisión	8
11. Barras de sincronización.....	8
12. Ejemplo de un diagrama de actividades	9
13. Representación de un componente	10
14. Relación de dependencia entre dos componentes	10
15. Ejemplo de un diagrama de componentes.....	11
16. Ejemplo de un diagrama de distribución	12
17. Diagrama entidad relación	16
18. Pata de gallo.....	17
19. Pata de gallo con una línea atravesada.....	17
20. Arquitectura de 1 capa.....	19
21. Arquitectura de 2 capas	20
22. Arquitectura de 3 capas	20
23. Arquitectura flexible de n capas	22

24. Diagrama entidad relación.....	36
25. Diagrama de casos de uso de consultas	38
26. Diagrama de secuencias de consultas	39
27. Diagrama de actividades de consultas	40
28. Diagrama de casos de uso de administración	41
29. Diagrama de casos de uso de mantenimiento.....	42
30. Diagrama de secuencias de crear en mantenimiento.....	44
31. Diagrama de actividades de crear en mantenimiento.....	45
32. Diagrama de secuencias de eliminar en mantenimiento	47
33. Diagrama de actividades de eliminar en mantenimiento	48
34. Diagrama de secuencias de actualizar en mantenimiento.....	50
35. Diagrama de actividades de actualizar en mantenimiento.....	51
36. Diagrama de casos de uso de usuarios	52
37. Diagrama de secuencias de crear usuario	54
38. Diagrama de actividades de crear usuario	55
39. Diagrama de secuencias de eliminar usuario	57
40. Diagrama de actividades de eliminar usuario	58
41. Diagrama de secuencias de actualizar usuario	60
42. Diagrama de actividades de actualizar usuario	61
43. Diagrama de secuencias de asignación de usuarios.....	63
44. Diagrama de actividades de asignación de usuarios.....	64
45. Diagrama de distribución de la aplicación	65
46. Cuadro para inicio de sesión	67
47. ¿Olvidó su contraseña? Paso 1.....	68
48. ¿Olvidó su contraseña? Paso 2.....	69
49. ¿Olvidó su contraseña? Paso 3.....	69
50. Opciones para usuarios que han iniciado sesión	69
51. Editar mis datos personales	70
52. Confirmación de datos personales actualizados	70

53. Cambiar mi contraseña	71
54. Confirmación de contraseña cambiada.....	71
55. Cambiar pregunta y respuesta de seguridad	72
56. Confirmación de datos personales actualizados.....	72
57. Elemento “Administración” del menú	73
58. ¿Incluir <i>curriculum vitae</i> ?	75
59. Formato archivo cargar estudiantes.....	77
60. Tabla de resultados de cargar estudiantes	78
61. ¿Incluir documento?	80
62. Opciones rol tutor.....	81
63. Administrar asesores	82
64. Confirmación para eliminar asesor	83
65. Administrar estudiantes	84
66. Opciones de asignación.....	84
67. Enviar correo electrónico	86
68. Asignar asesor	87
69. Opción de paginación	88
70. Administrar documentos	89
71. Ver contenido de un documento	89
72. Historial de un documento	90
73. Administrar usuarios	93
74. Administrar roles	94
75. Administrar estados	96
76. Administrar estudiantes (rol administrador).....	97
77. Asignar tutor.....	99
78. Menú de consultas.....	100
79. Asesores	100
80. Datos del asesor	101
81. Protocolos	102

82. Diagrama de implementación de la base de datos.....	108
83. Arquitectura de la aplicación.....	126
84. Cooperator.Framework.Web.dll.....	133
85. FredCK.FCKeditorV2.dll.....	134
86. GMDatePicker.dll.....	134
87. Instalación – prerequisite.....	156
88. Instalación – bienvenida.....	156
89. Instalación – servidor de base de datos.....	157
90. Instalación – preparado para instalar.....	157
91. Instalación – progreso.....	158
92. Instalación – finalizada.....	158
93. Copiar directorio de la aplicación.....	159
94. Crear directorio virtual.....	160
95. Alias del directorio virtual.....	161
96. Directorio de contenido del sitio <i>Web</i>	162
97. Permisos de acceso.....	162
98. Crear un <i>Application Pool</i>	163
99. Ficha documentos.....	164
100. Ficha <i>ASP.NET</i>	164
101. Ficha <i>directorio virtual</i>	165
102. Crear base de datos.....	166
103. Carga de <i>scripts</i>	167
104. Permisos de carpeta.....	168

TABLAS

I. Tabla <i>aspnet_Users</i>	109
II. Tabla <i>aspnet_Membership</i>	110
III. Tabla <i>aspnet_Roles</i>	112

IV. Tabla <i>aspnet_UsersInRoles</i>	113
V. Tabla <i>asesor</i>	113
VI. Tabla <i>estudiante</i>	114
VII. Tabla <i>asignación</i>	115
VIII. Tabla <i>estado</i>	116
IX. Tabla <i>documento</i>	117
X. Tabla <i>historial</i>	118
XI. Procedimientos almacenados.....	120
XII. Métodos y funciones de la capa de acceso a datos.....	127
XIII. Métodos y funciones de la clase <i>AppUser</i>	129
XIV. Métodos y funciones de la clase <i>AppAdviser</i>	129
XV. Métodos y funciones de la clase <i>AppStudent</i>	130
XVI. Métodos y funciones de la clase <i>AppDocument</i>	131
XVII. Controles de usuario.....	135
XVIII. Páginas Web.....	136

GLOSARIO

Administrador	Persona que gestiona un determinado servicio en la red.
Aplicación	Es un programa con el cual el usuario final interactúa, es decir, es aquel programa que permite la interacción entre el usuario y la computadora.
Asíncrono	Término utilizado para especificar la ejecución de distintos procesos de forma independiente unos de los otros, respecto al tiempo.
Asesor	Profesional universitario con conocimiento en la materia del tema aprobado de graduación.
Blog	También conocido como <i>weblog</i> o bitácora, es un sitio <i>Web</i> periódicamente actualizado que recopila cronológicamente, textos o artículos de uno o varios autores, donde el más reciente aparece primero, con un uso o temática en particular, siempre conservando el autor la libertad de dejar publicado lo que crea pertinente.
Carné	Documento de identificación, universitario en este caso, que posee los datos generales del estudiante, fotografía y un número único para uso con fines estudiantiles.
Cookie	Es un fragmento de información que se almacena en el disco duro del visitante de una página <i>Web</i> , a través de su navegador, a petición del servidor de la página.
Curriculum vitae	Documento que detalla los datos personales, la formación y la experiencia.

Escalabilidad	Es la capacidad de un sistema informático de adaptarse a un número de usuarios cada vez mayor, sin perder calidad en los servicios. En general, se podría definir como la capacidad del sistema informático de cambiar su tamaño o configuración, para adaptarse a las circunstancias cambiantes.
Firewall	Un cortafuegos (o <i>firewall</i> en inglés), es un elemento de <i>hardware</i> o <i>software</i> utilizado en una red de computadoras, para prevenir algunos tipos de comunicaciones prohibidos, según las políticas de red que se hayan definido en función de las necesidades de la organización.
Hardware	Se denomina <i>hardware</i> o soporte físico al conjunto de elementos materiales que componen un ordenador.
Interfaz	Es la parte de un programa informático que permite a éste comunicarse con el usuario o con otras aplicaciones permitiendo el flujo de información.
Java	Es un lenguaje de programación de alto nivel, orientado a objetos.
Link	Enlace, hipervínculo. Conexión con otro documento <i>Web</i> por medio de la dirección <i>URL</i> . Los enlaces aparecen en el texto de un documento <i>Web</i> en forma de texto subrayado y de distinto color.
Rol	Es un anglicismo que se refiere al conjunto de funciones, normas, comportamientos y derechos definidos que se esperan de una persona.
Servidor	El computador en el que se ejecuta un programa que realiza alguna tarea en beneficio de otras aplicaciones llamadas clientes.
Síncrono	Término utilizado para especificar la ejecución de distintos procesos de forma dependiente unos de los otros, respecto al tiempo.

Software	Es la parte lógica de una computadora, esto es, el conjunto de programas que puede ejecutar el <i>hardware</i> para la realización de las tareas de computación a las que se destina.
HTML	Acrónimo inglés de <i>Hypertext Markup Language</i> (lenguaje de etiquetado de documentos hipertextual), es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas <i>Web</i> .
Tutor	Profesional universitario nombrado por la escuela de Ciencias y Sistemas, con conocimientos específicos sobre el tema del trabajo de graduación a evaluar.
Web	Es un sistema de hipertexto que funciona sobre Internet. Para ver la información se utiliza una aplicación llamada navegador <i>Web</i> para extraer elementos de información (llamados "documentos" o "páginas <i>Web</i> ") de los servidores <i>Web</i> (o "sitios") y mostrarlos en la pantalla del usuario.
XML	Es el acrónimo del inglés <i>eXtensible Markup Language</i> (lenguaje de marcado ampliable o extensible). No sólo es un lenguaje de marcado, sino también un metalenguaje que permite describir otros lenguajes de marcado. Permite el uso ilimitado de los tipos de datos que pueden utilizarse en <i>Internet</i> , lo cual resuelve los problemas que surgen entre las organizaciones que deben intercambiar datos procedentes de estándares distintos.

RESUMEN

Dada la necesidad de un sistema en el que se puedan administrar los trabajos de graduación de la Escuela de Ciencias y Sistemas, de la Facultad de Ingeniería, se llevó a cabo el análisis, diseño e implementación de una aplicación *Web* que proporciona las herramientas necesarias para dar solución al problema de administración de los trabajos antes mencionados. El contenido del documento se divide en cuatro capítulos que se describen a continuación.

El capítulo uno presenta los fundamentos básicos que son necesarios para la construcción de la aplicación, como lo es una introducción a *UML* que es utilizado para la construcción de los diferentes diagramas que forman parte del diseño de la aplicación, el modelo entidad relación que se utiliza para el diseño de la base de datos, y las diferentes arquitecturas de *software* que son utilizadas para la implementación de la aplicación.

En el capítulo dos se describe la forma actual en la que se administran los trabajos de graduación, junto con el problema que esta administración representa. Se presentan los requerimientos indicados por el Coordinador de Privados y Revisión de Trabajos de Graduación y también la propuesta para satisfacerlos.

El capítulo tres presenta el diseño de la base de datos que incluye las tablas y sus propiedades, y el diseño de la aplicación que describe los módulos y cada uno de sus diagramas de casos de uso, de secuencias y de actividades.

El capítulo cuatro está formado por el manual de usuario que es utilizado para que los usuarios finales puedan hacer uso de la aplicación; el manual técnico que describe la tecnología utilizada para la construcción, la implementación de la base de datos, y descripción de los métodos y componentes necesarios para comprender y dar mantenimiento al sistema. Finalmente, en el manual de instalación se indican los requerimientos del equipo en el que se instalará la aplicación, y se proporciona una guía que describe los pasos para realizar la instalación.

OBJETIVOS

- **General**

Implementar un sistema de control que permita conocer los proyectos terminados, y los protocolos aprobados para los trabajos de graduación de la Escuela de Ciencias y Sistemas de la Facultad de Ingeniería, de la Universidad de San Carlos de Guatemala.

- **Específicos**

1. Conocer y analizar la forma actual de almacenar los trabajos de graduación, y así realizar la definición del problema junto con sus requerimientos.
2. Llevar a cabo el diseño de una solución *Web* para el problema definido que proporcione una interfaz para conocer los datos necesarios de los trabajos de graduación, protocolos aprobados, incluyendo los datos de los estudiantes, asesores y tutores.
3. Implementar la aplicación en el servidor de la Escuela de Ciencias y Sistemas de la Facultad de Ingeniería.

INTRODUCCIÓN

En la actualidad, no existe una forma rápida, práctica y eficiente de conocer los temas de los trabajos de graduación finalizados de la escuela de Ciencias y Sistemas, de la Facultad de Ingeniería, asimismo de los temas de los protocolos que han sido aprobados.

Especialmente en el caso de los protocolos, no es posible conocer qué temas están aprobados y su vigencia, ya que no se cuenta con un sistema que permita administrar dicha información, esta información únicamente es manejada por el catedrático a cargo del curso Seminario de Investigación, lo cual retrasa los procesos de desarrollo de protocolos debido a falta de una comunicación pronta.

Con la implementación de este trabajo de graduación se conocerá la información de los trabajos ya finalizados, los protocolos presentados por los estudiantes y su estado, cada uno con sus respectivos datos y resumen, así como la información personal de los asesores.

A partir de este trabajo de graduación, se logrará un ahorro de tiempo y trabajo consiguiendo toda la información necesaria y estará además disponible para varios usuarios a la vez.

1. FUNDAMENTOS BÁSICOS

1.1. Introducción a UML

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*), tiene como propósito proveer a los desarrolladores un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprenden el desarrollo del *software*. Una de las razones para que este lenguaje de modelado se haya convertido en un estándar, es el hecho de que sea independiente de los lenguajes de programación.

Es importante aclarar que UML es un lenguaje y no una metodología; la razón es que un lenguaje fácilmente se puede acoplar a la forma en que un negocio se conduce sin hacer cambios y una metodología no. A continuación se explican los principales diagramas de UML.

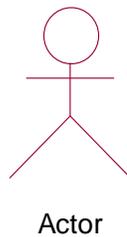
1.1.1. Casos de uso

Un caso de uso muestra la funcionalidad de la aplicación, y su propósito principal es ayudar a los desarrolladores a visualizar los requerimientos funcionales de la aplicación.

Un diagrama de casos de uso está compuesto de los siguientes elementos:

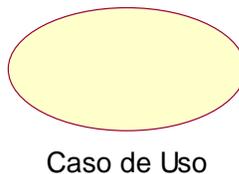
- Actor: Es un rol que un usuario o una entidad externa o interna juega al utilizar servicios que son proveídos por el sistema y que a la vez puede proveer servicios al sistema. Ver figura 1.

Figura 1. **Actor**



- Caso de uso: Es una descripción de los servicios que el sistema provee y que se puede ejecutar a petición de un actor o de otro caso de uso. Ver figura 2.

Figura 2. **Caso de uso**



- Relaciones: Hay tres tipos de relación.
 - Asociación: Se utiliza para indicar que un actor o caso de uso requiere del servicio de otro caso de uso. Ver figura 3.
 - Dependencia: Se utiliza para indicar dependencia entre clases, es decir que una clase depende de otra. Ver figura 3.

- Generalización: Esta relación es de uso exclusivo para casos de uso y no de actores. Se divide en dos tipos, de uso <<uses>> o de herencia <<extend>>. La generalización de herencia se recomienda cuando hay un caso de uso que es similar a otro en características. La generalización de uso se utiliza cuando se tienen características similares en más de un caso de uso y no se quieren tener copias de las características. Ver figura 3.

Figura 3. Tipos de relaciones

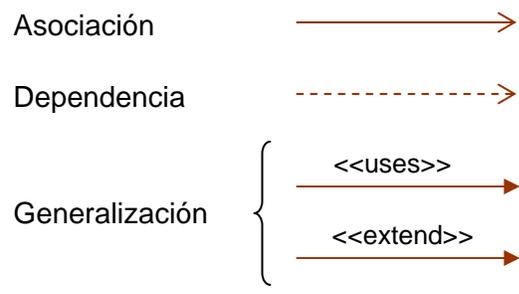
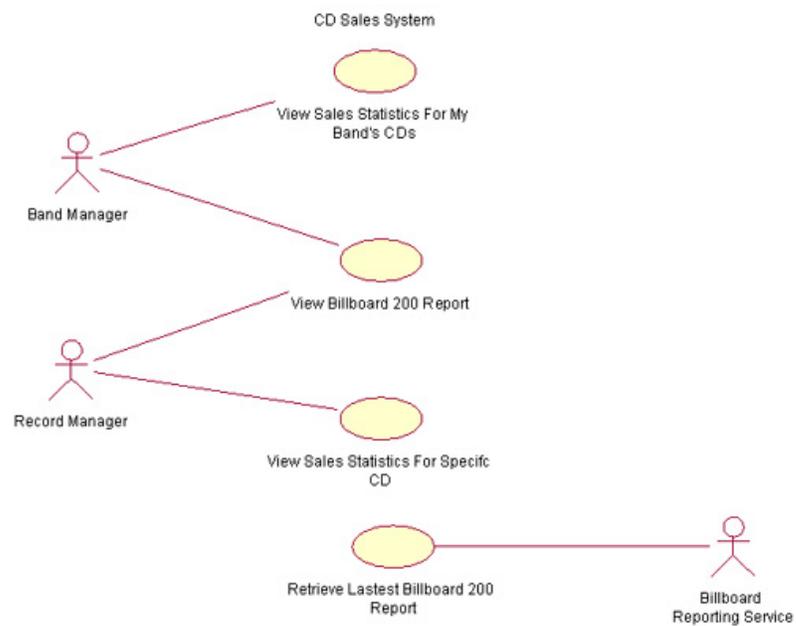


Figura 4. Ejemplo de un diagrama de casos de uso



Fuente. Donald Bell. **UML basics: an introduction to the Unified Modeling Language.**

En la figura 4 se muestra el ejemplo de un caso de uso en el que el sistema permite al gerente de una banda ver el reporte de ventas de los CD de su banda y el reporte del *Billboard* 200. También permite al gerente de grabación ver el reporte de ventas de un CD específico y el reporte del *Billboard* 200. El diagrama también nos muestra que el reporte del *Billboard* 200 es obtenido a través de un sistema externo llamado *Billboard Reporting Service*.

1.1.2. Diagrama de secuencias

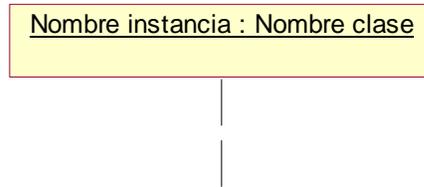
Muestra el flujo de un caso de uso específico o una parte de un caso uso y muestra las llamadas entre los objetos en una secuencia de tiempo en un orden específico. Los casos de uso pueden ser convertidos en uno o más diagramas de secuencia, y estos también pueden ser utilizados para documentar cómo interactúan los objetos en un sistema existente.

Un diagrama de secuencias tiene dos dimensiones: La dimensión vertical muestra la secuencia de los mensajes o llamadas conforme van ocurriendo en el tiempo, y la dimensión horizontal muestra de izquierda a derecha los objetos a los que son enviados los mensajes.

Un diagrama de secuencias está compuesto de los siguientes elementos:

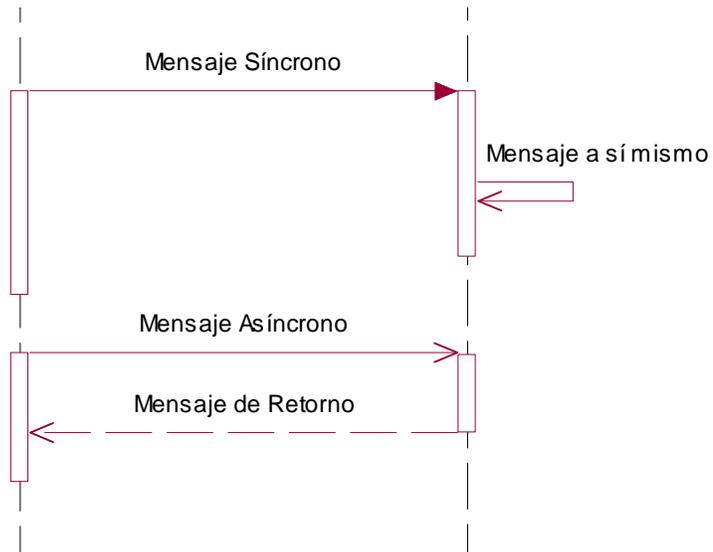
- Línea de vida: Representan instancias de objetos o roles en la secuencia que está siendo modelada. Como se puede ver en la figura 5, el rectángulo representa la instancia en particular y se nombra con la notación: "Nombre instancia : Nombre clase", y la línea punteada se utiliza para representar el flujo del objeto o rol en el tiempo.

Figura 5. **Línea de vida**



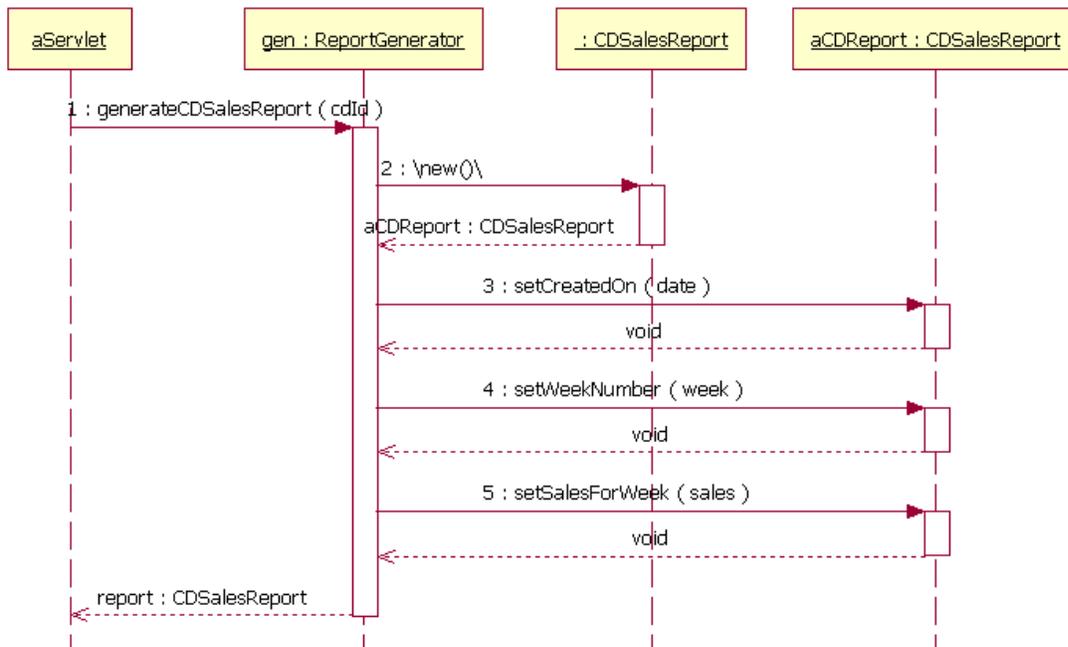
- **Mensajes:** Son utilizados para representar operaciones o métodos que son implementados en el objeto que recibe el mensaje, pueden ser síncronos o asíncronos. Los mensajes síncronos se representan con una línea que tiene una cabeza de flecha sólida y los mensajes asíncronos con una línea que tiene una cabeza de flecha abierta. Los mensajes pueden estar dirigidos a otro objeto o a sí mismos. También existen mensajes de retorno que son utilizados para aumentar el grado de comprensión del diagrama de secuencia y se representan con una línea punteada y una cabeza de flecha abierta, los mensajes de retorno son opcionales. En la figura 6 se pueden ver los diferentes tipos de mensajes.

Figura 6. **Tipos de mensajes**



En la figura 7 se puede observar cómo crear un reporte de ventas de CD. Como se puede ver los mensajes también pueden incluir parámetros que son entregados por el objeto que envía el mensaje. Se escriben en paréntesis luego del nombre del mensaje.

Figura 7. Ejemplo de un diagrama de secuencias



Fuente. Donald Bell. *UML basics: an introduction to the Unified Modeling Language.*

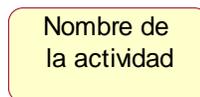
1.1.3. Diagrama de actividades

Se utiliza para modelar el flujo de control entre uno o más objetos mientras se procesa una actividad. Pueden ser utilizados para modelar casos de uso con un mayor nivel de detalle, pero también se usan para modelar la forma en que una compañía hace o le gustaría hacer sus procesos de negocios.

Se enfoca únicamente en las acciones internas sin hacer referencia hacia las acciones externas. Un diagrama de actividades está compuesto de los siguientes elementos:

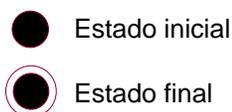
- Estado de acción: Se representa con un rectángulo con bordes redondeados (ver figura 8) y también es llamado acción o actividad. En el interior se escribe el nombre la actividad que representa.

Figura 8. **Estado de acción**



- Estados: Como un diagrama de actividades muestra un flujo de acciones se debe indicar el inicio y el final de este flujo (ver figura 9). Para ello se utiliza un círculo sólido para indicar el estado inicial que es el punto a partir del cual se inicia la lectura del diagrama, es importante no olvidar que únicamente puede existir un estado inicial y que este puede estar conectado únicamente a un estado de acción. Para el estado final se utiliza un círculo que rodea a un círculo sólido más pequeño, este estado indica el final de la lectura del diagrama y en este caso si es posible que existan uno o más estados finales.

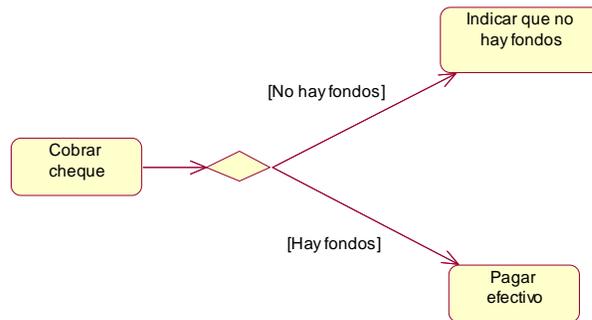
Figura 9. **Estados inicial y final**



- Transiciones: Son flechas que se utilizan para conectar los estados de acción con eventos implícitos.

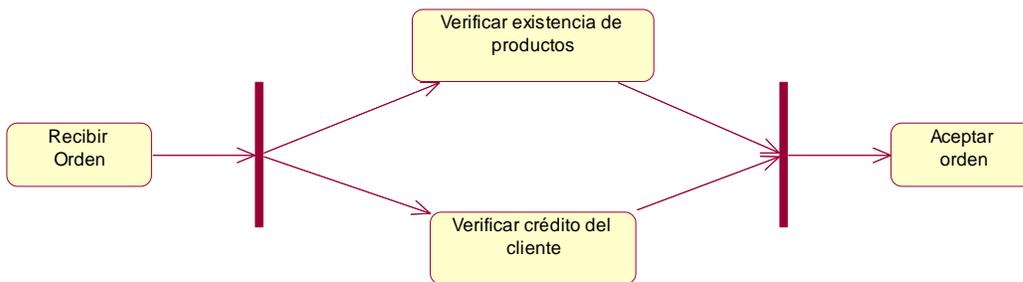
- Decisiones: Se representa mediante un diamante (ver figura 10), de este salen un conjunto de transiciones en el cual cada camino tiene una etiqueta distinta. El estado de decisión representa una condición.

Figura 10. Ejemplo de un estado de decisión



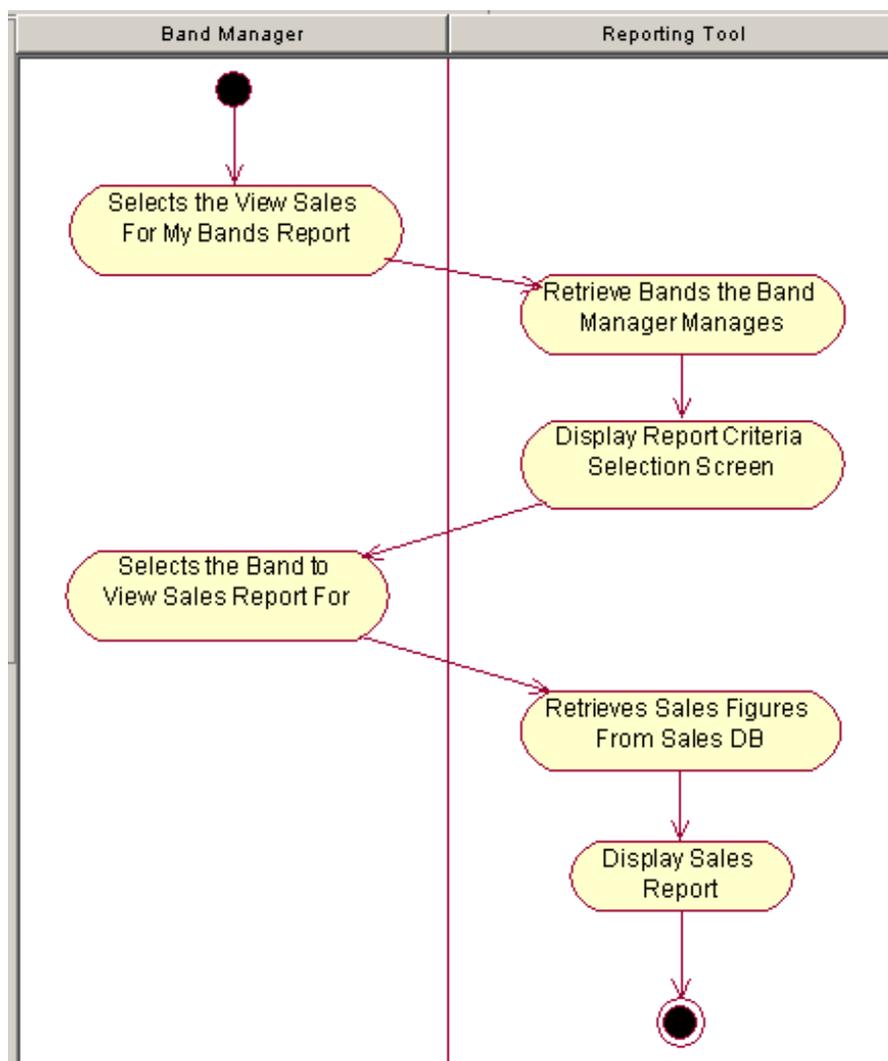
- Barra de sincronización: Se representa mediante una línea gruesa (ver figura 11). Se puede utilizar de dos formas, cuando dos o más transiciones salen de ella indica un estado de de-sincronización y se hace para indicar que las acciones se pueden realizar en paralelo. Cuando dos o más transiciones entran en ella indica un estado de sincronización y se utiliza para unir el flujo de actividades.

Figura 11. Barras de sincronización



En la figura 12 se muestra el ejemplo de un diagrama de actividades que utiliza líneas divisoras o *swinlanes* por su nombre en inglés que son usadas para separar las actividades que realizan los distintos objetos, en este caso el gerente de la banda y la herramienta de reportes. En este diagrama se muestra la secuencia para poder mostrar un reporte de ventas.

Figura 12. Ejemplo de un diagrama de actividades



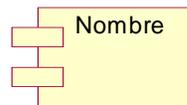
Fuente. Donald Bell. *UML basics: an introduction to the Unified Modeling Language*.

1.1.4. Diagrama de componentes

Su principal propósito es mostrar las relaciones estructurales entre los componentes del sistema y mostrar una vista física del mismo. Este diagrama es útil para poder hacer un mapa de cómo se va a implementar el *software* en el sistema.

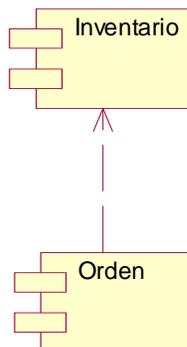
Los componentes se representan por medio un rectángulo con dos pequeños rectángulos sobresaliendo de su lado izquierdo (ver figura 13), y en el interior se escribe el nombre del componente.

Figura 13. Representación de un componente



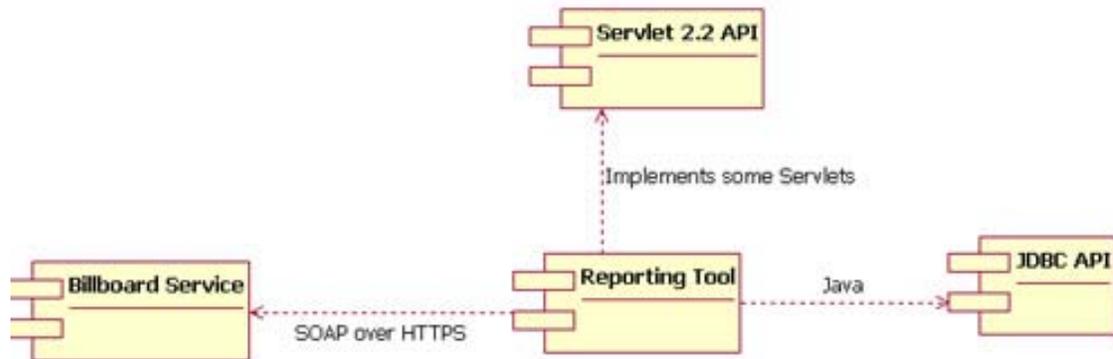
Las relaciones que unen a los componentes indican dependencia y se utiliza una flecha con trama punteada y cabeza abierta para representarlas. En la figura 14 podemos ver un ejemplo relación de dependencia, en este caso una orden depende la existencia de productos en el inventario.

Figura 14. Relación de dependencia entre dos componentes



En la figura 15 se muestra el diagrama de componentes de una herramienta de reportes y las dependencias que tiene para poder funcionar.

Figura 15. Ejemplo de un diagrama de componentes



Fuente. Donald Bell. *UML basics: an introduction to the Unified Modeling Language*.

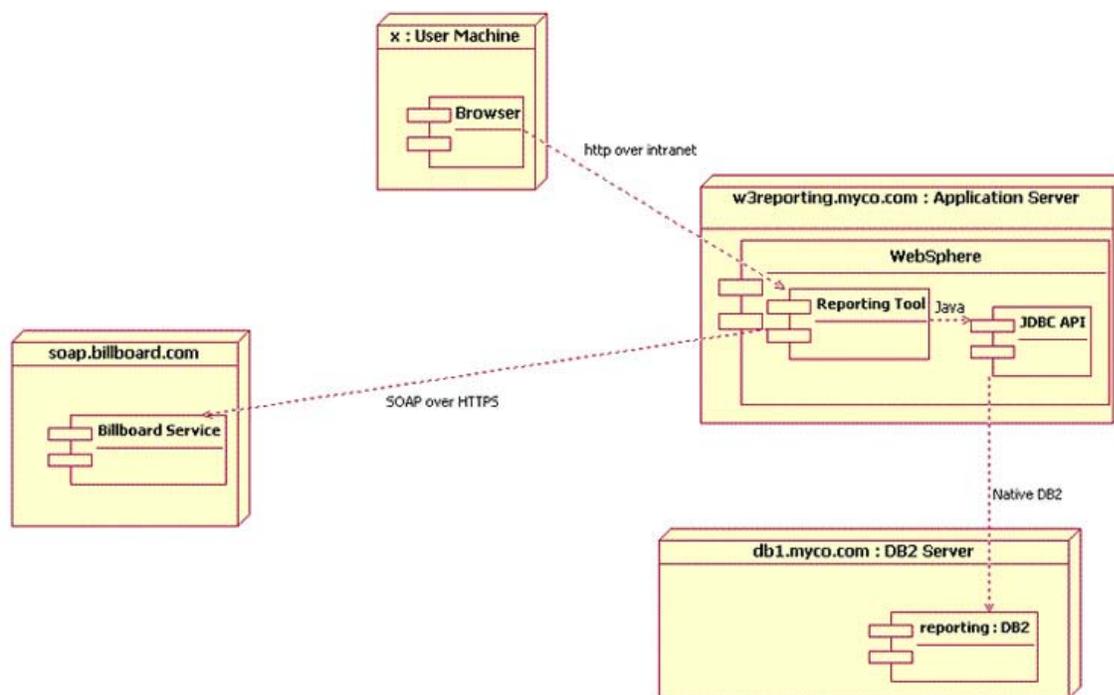
1.1.5. Diagrama de distribución

Muestra como el sistema está físicamente distribuido en el ambiente de *hardware* y como se comunican los componentes unos con otros.

Este diagrama utiliza los mismos elementos que un diagrama de componentes con la adición del elemento nodo que se utiliza para representar una máquina física o virtual. Un nodo se representa por medio de un cubo con el nombre en la parte superior y escrito con la notación: "Nombre instancia : Tipo instancia".

En la figura 16 se observa el diagrama de distribución de una herramienta de reportes y como interactúa con diferentes sistemas externos. Los usuarios acceden a través de su máquina personal por medio de un explorador que se comunica con la herramienta de reportes que reside en un servidor de aplicación. La base de datos reside en otro servidor y la herramienta de reportes se comunica con ella por medio de un componente que utiliza lenguaje Java. Finalmente el componente de la herramienta de reportes se comunica con un servicio proporcionado por *Billboard* para obtener el listado canciones.

Figura 16. Ejemplo de un diagrama de distribución



Fuente. Donald Bell. *UML basics: an introduction to the Unified Modeling Language*.

1.2. Introducción a bases de datos

Un sistema de base de datos es básicamente un sistema de información lógicamente relacionado que permite que diferentes usuarios y aplicaciones interactúen. En realidad la información puede ser todo lo que los usuarios quieran almacenar, recuperar y actualizar y además sea de importancia para auxiliarse en procesos que se realizan en las tareas diarias de los usuarios o la empresa.

Las ventajas de la utilización de una base de datos se mencionan a continuación:

- Compactación: Se eliminan las cantidades voluminosas de papel.
- Velocidad: No es necesario buscar en archivos o libros la información que se necesita, una computadora puede almacenar, recuperar y actualizar los datos de una forma rápida y transparente. De esta forma se ahorra tiempo en la manipulación de la información.
- Menos trabajo: No es necesario llevar el control de la información a mano. La computadora hace la mayor parte del trabajo.
- Actualidad: La información se encuentra disponible en todo momento y para varios usuarios a la vez.
- Seguridad: Se tiene control de quien puede acceder a la información y quien no, por lo tanto los datos almacenados tienen restricciones de seguridad según el rol del usuario que los solicita.

1.2.1. Modelo entidad relación

El modelo entidad relación es uno de los más utilizados hoy en día ya que proporciona fácil entendimiento y es sencillo de elaborar. Se utiliza para representar una base de datos que a la vez representa la realidad de un sistema. A continuación se definen los componentes del modelo entidad relación:

- Entidades: Son objetos que llevan información útil para la empresa. Se clasifican en entidades normales y entidades débiles. Las entidades débiles son aquellas que para poder existir dependen de la existencia de alguna otra entidad, las entidades normales son aquellas que existen sin la necesidad de otra entidad.
- Atributos: Son las propiedades de la entidades, estos califican, cuantifican e identifican a una entidad. Los atributos pueden ser:
 - Simples o compuestos: Un atributo compuesto es el que está formado por varios atributos simples. Un ejemplo de un atributo compuesto es la cédula de vecindad, se compone de los atributos simples registro y número.
 - Clave: Un atributo puede ser una clave primaria o una clave foránea o extranjera. Las claves primarias deben cumplir con unicidad y minimalidad, la unicidad se refiere al hecho de que no pueden existir dos registros dentro una tabla que sean iguales y la minimalidad se refiere al hecho de que la clave (si es compuesta) debe contener el número de atributos mínimos para ser único.

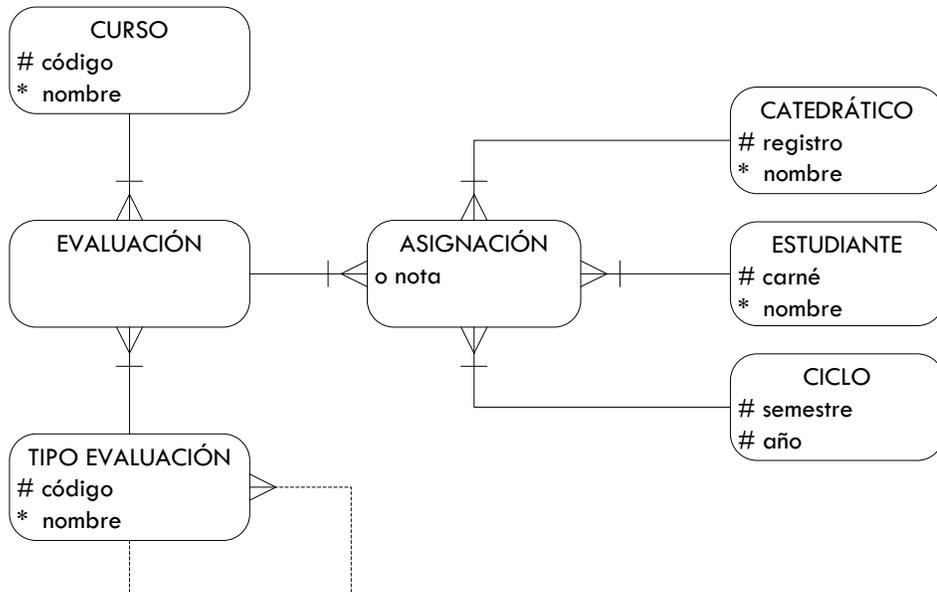
Una observación importante es que las claves primarias no pueden ser nulas, es decir, siempre debe ser asignado un valor a la clave primaria. Una clave foránea o extranjera es una clave en una entidad que hace referencia a una clave primaria de otra entidad, las claves foráneas pueden ser nulas siempre y cuando no formen parte de la clave primaria de la entidad a la que pertenecen.

- Vínculos: También son llamados relaciones, se definen como la asociación entre una o más entidades. Las entidades involucradas en un vínculo son llamadas participantes y al número de participantes en un vínculo se le llama grado. La cardinalidad especifica el número de ejemplares de vínculos en los que puede participar una entidad. Un vínculo puede ser uno a uno, uno a muchos o muchos a muchos.

1.2.2. Diagrama entidad relación

Constituye una técnica para representar de forma gráfica la estructura de la distribución de los datos almacenados en un sistema. A continuación en la figura 17 se muestra un diagrama entidad relación que se será utilizado para explicar cómo se aplican cada unos de los componentes del modelo entidad relación.

Figura 17. Diagrama entidad relación



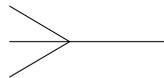
- Entidades: Se representan con un rectángulo con las esquinas redondeadas, el nombre de la entidad se escribe con mayúsculas.
- Atributos: Están contenidos en las entidades y pueden ser de tres tipos:
 - Claves primarias: Se presentan con el símbolo #.
 - Atributos opcionales: Se representan con el símbolo o.
 - Atributos obligatorios: Se representa con el símbolo *.
- Vínculos: Son representados por líneas que enlazan a las entidades y pueden ser de dos tipos:
 - Opcionales: Se representan con una línea con trama punteada.

- Obligatorios: Se representan con una línea con trama continua.

La cardinalidad de un vínculo puede ser de dos tipos:

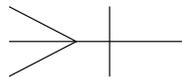
- Uno: Se representa con una línea.
- Varios o muchos: Se representa con una pata de gallo, ver figura 18.

Figura 18. **Pata de gallo**



Las claves foráneas se crean en las entidades que tienen cardinalidad de varios o muchos, si el vínculo es opcional la clave foránea es opcional de lo contrario es obligatoria. Cuando el vínculo de varios o muchos tiene una línea atravesada (ver figura 19), indica que la clave foránea forma parte de la clave primaria de la entidad.

Figura 19. **Pata de gallo con una línea atravesada**



1.3. Modelos de arquitectura de *software*

Una analogía que explica muy bien los modelos de arquitectura de *software* es la comparación entre la construcción de una casa y el desarrollo de *software*, en ambos casos se determina la estructura del producto final conociendo que se utilizará, donde se utilizará y cuando se utilizará. Cada detalle debe cumplir con los requerimientos de los clientes.

Para una casa el arquitecto realiza los planos de lo que será la casa y realiza los cambios requeridos por el cliente hasta que este está satisfecho. Para el *software* esta fase, es la fase de análisis, en este punto el equipo de desarrollo crea un documento que contiene las especificaciones y requerimientos del proyecto y el cliente puede hacer todas las observaciones que considere necesarias hasta que el equipo construya un documento que realmente satisfaga todo los puntos importantes que el cliente requiere. Luego en la construcción de la casa se definen los elementos como las tuberías, el alambrado eléctrico, ventilación, etc.

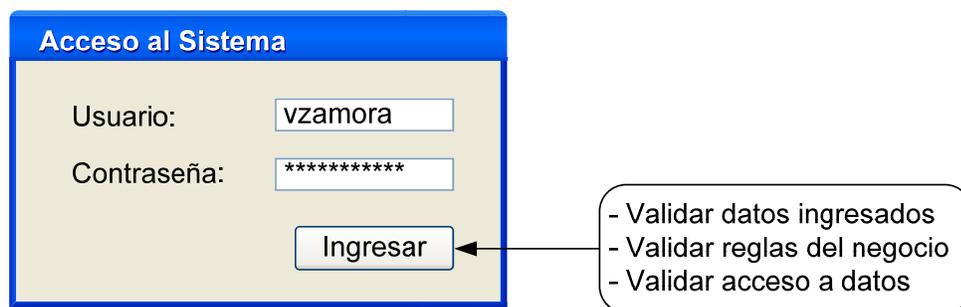
En la construcción de *software* se selecciona el manejador de base de datos, lenguaje de programación, las clases a utilizar y se decide como se va a mover la información a través del sistema. Es este punto se puede iniciar a construir la casa y de la misma forma se tiene suficiente información para iniciar a desarrollar el *software*. De aquí en adelante toda la construcción (casa y *software*) es constantemente monitoreada y revisa que cada fase esté cumpliendo con los requerimientos para lograr terminar el proyecto con éxito.

A continuación se mostrará una pequeña descripción de los tipos de arquitectura utilizados en el pasado y los utilizados en la actualidad.

1.3.1. Arquitectura de 1 capa

En este tipo de arquitectura todo lo que forma parte de la aplicación está en el mismo lugar (ver figura 20), por lo que las aplicaciones se vuelven muy grandes por lo tanto es difícil darles mantenimiento o actualizarlas. Hoy en día esta arquitectura no es utilizada ya que siempre se busca separar la interfaz del usuario, la lógica o reglas del negocio y los datos para construir aplicaciones pequeñas.

Figura 20. **Arquitectura de 1 capa**

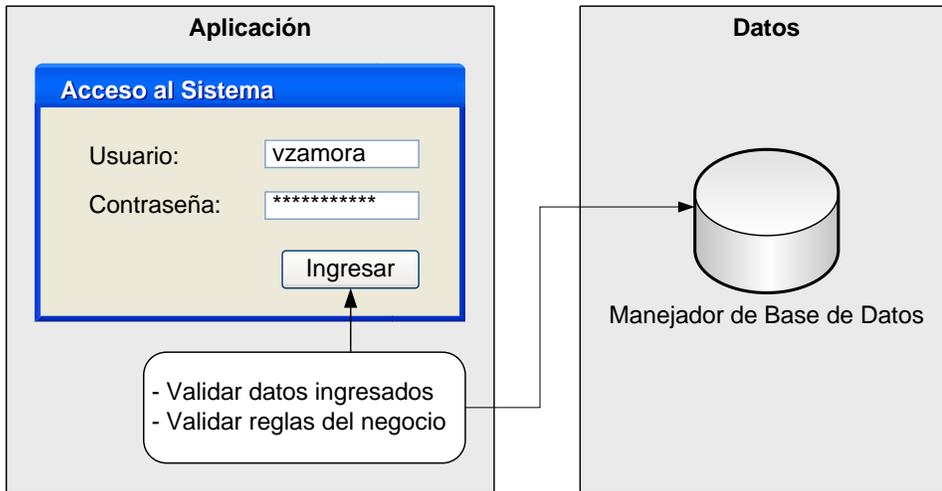


1.3.2. Arquitectura de 2 capas

Aquí se separa la aplicación de los datos. En esta arquitectura no solo se separan los datos sino que también el manejador de la base de datos, esto se hace para lograr que el acceso y manejo de los datos sea óptimo e independiente de la aplicación.

La interfaz del usuario y la lógica o reglas del negocio aún se encuentran en el mismo lugar (ver figura 21), pero el acceso a los datos soporta cientos de usuarios conectados de forma concurrente, al mismo tiempo esto representa una desventaja ya que en sistemas muy extensos se puede crear un cuello de botella lo que puede ser perjudicial para el rendimiento y la escalabilidad.

Figura 21. **Arquitectura de 2 capas**

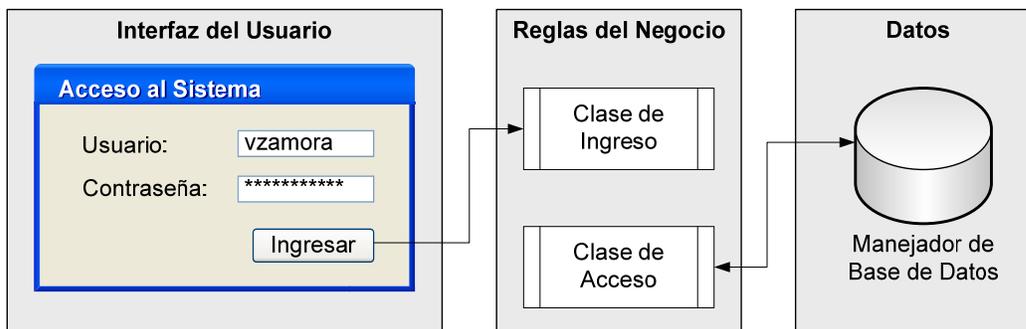


Muchas aplicaciones de hoy en día utilizan esta arquitectura ya que es fácil de implementar y aunque tiene sus ventajas es necesario considerar la arquitectura de 3 capas.

1.3.3. **Arquitectura de 3 capas**

La idea principal de esta arquitectura es separar completamente la interfaz del usuario, la lógica o reglas del negocio y los datos para que estas se conviertan en entidades completamente separadas (ver figura 22) para lograr gran escalabilidad e independencia.

Figura 22. **Arquitectura de 3 capas**



Aunque las 3 capas están lógicamente separadas aún existe la posibilidad de que estén implementadas en el mismo servidor o en tres servidores o en más sistemas independientes.

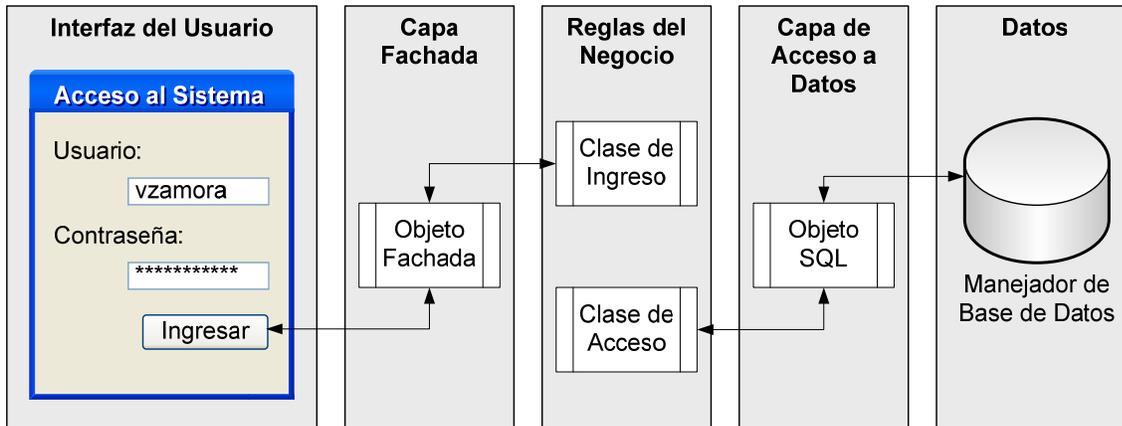
1.3.4. Arquitectura flexible de n capas

Hay dos cosas que hacen que la arquitectura de 3 capas no sea la mejor. Primero, hay muchas aplicaciones que repiten la lógica del negocio con diferentes interfaces para cada cliente, lo que no es una buena solución ya que lleva a tener secciones de código duplicadas que son difíciles de mantener y actualizar.

La solución a este problema es crear una capa llamada fachada, en lugar de escribir secciones de código duplicadas para la lógica del negocio, se envuelven las reglas del negocio en la capa fachada y se deja que los clientes que solicitan información hagan los cambios. Segundo, el acceso a los datos tiene el mismo problema, muchas veces se repite el código de acceso a los datos o el código de manipulación cuando es implementado en la arquitectura de 3 capas.

El problema se da cuando hay algún cambio en el manejador de la base de datos, cuando se encuentra una mejor forma de acceder los datos o cuando ocurre algún cambio de tecnología en el acceso a los datos; la mejor solución es separar estas porciones de código en una nueva capa llamada capa de acceso de datos que contiene todo el código utilizado para el acceso y manipulación de los datos. Esta nueva capa brinda la posibilidad de poder actualizar y dar mantenimiento al código de una forma sencilla y rápida.

Figura 23. Arquitectura flexible de n capas



En la figura 23 se puede observar un ejemplo de la arquitectura flexible de n capas, esta arquitectura es llamada flexible ya que no hay ninguna regla que especifique la cantidad de capas que deben ser implementadas, podría ser que no se necesite la capa fachada o la de acceso a datos, esto se da en aplicaciones *Web* que son muy pequeñas y que son fáciles de mantener o actualizar y que el hecho de implementar una capa más implica más trabajo.

También se puede dar el caso que se necesite una capa más para poder hacer más fácil la actualización o mantenimiento en aplicaciones muy extensas, la idea es poder dividir la aplicación en las capas que sean necesarias y que minimicen la cantidad de código duplicado y a la vez hagan que el mantenimiento de la aplicación sea rápido y sencillo.

2. FASE DE ANÁLISIS

2.1. Descripción del sistema actual

A continuación se describe el proceso que se debe realizar para poder llevar a cabo la realización del trabajo de graduación en la Facultad de Ingeniería y cada una de las entradas y salidas de cada función.

a. Aprobar curso propedéutico

Entradas: Constancia de tener aprobados 200 créditos.

Salidas: Constancia de aprobación del curso propedéutico.

b. Aprobar protocolo

Entradas: Protocolo de trabajo de graduación.

Salidas: Carta de aprobación de protocolo, ficha de seguimiento de trabajos de graduación.

c. Trabajo con el asesor

Entradas: Entregas periódicas de los avances en el trabajo de graduación con sus respectivas correcciones.

Salidas: Retroalimentación del trabajo de graduación, carta de aprobación de trabajo de graduación, aprobación en la ficha de seguimiento.

d. Aprobación del revisor

Entradas: Carta de aprobación del asesor, trabajo de graduación.

Salidas: Carta de aprobación del revisor, visto bueno en la ficha de seguimiento.

e. Revisión en lingüística

Entradas: Trabajo de graduación con el estilo formato con el que será impreso.

Salidas: Retroalimentación de cambios necesarios, visto bueno en la ficha de seguimiento.

f. Aprobación del Director de Escuela

Entradas: Trabajo de graduación revisado por el departamento de lingüística, carta de aprobación del asesor y revisor.

Salidas: Carta de aprobación del Director de Escuela, visto bueno en la ficha de seguimiento.

g. Orden de impresión

Entradas: Trabajo de graduación sin corregir y corregido. Cartas de aprobación del asesor, revisor y Director de Escuela originales y ficha de seguimiento.

Salidas: Carta con autorización de impresión.

h. Revisión en lingüística

Entradas: Trabajo de graduación impreso.

Salidas: Visto bueno en la ficha de seguimiento.

i. Trámites en Control Académico

Entradas: Trabajo de graduación impreso, ficha de seguimiento, documentos para necesario para tramitar examen público de graduación.

Salidas: Autorización para realizar examen público de graduación.

j. Examen público de graduación.

2.2. Definición del problema

Actualmente no es posible conocer de una forma accesible los temas de los trabajos de graduación finalizados para los estudiantes de la escuela de Ciencias y Sistemas de la Facultad de Ingeniería, así como los temas de los protocolos que han sido aprobados, ya que no hay una base de datos que contenga dicha información.

En el caso de los protocolos no hay forma de conocer cuáles temas están aprobados y cuál es su vigencia, ya que esta información es manejada en el curso de Seminario de Investigación que pertenece al décimo semestre de la carrera. Este curso actualmente es impartido por los Ingenieros Carlos Azurdia, Edgar Santos, Sonia Castañeda y Stanley Barrios, que a la vez ejercen la función de Tutor en el desarrollo del trabajo de graduación.

El Coordinador de Privados y Revisión de Trabajos de Graduación es el Ing. Carlos Azurdia, la información que es recolectada en el curso de Seminario de Investigación por los docentes-tutores es administrada por el Ing. Azurdia, pero no le es posible publicarla o colocarla al alcance de los estudiantes, asesores y tutores, ya que no se cuenta con sistema que permita administrar dicha información.

Un problema que se ha detectado a raíz de lo antes mencionado, es la posible falta de información para futuros protocolos o trabajos de graduación si por algún motivo el Ing. Azurdia no pudiera prestar sus servicios en la Escuela de Ciencias y Sistemas. Uno de los principales problemas se presentaría con el hecho de no conocer los temas o contenido de los temas para tomar decisiones al momento de aprobar o rechazar una propuesta de un estudiante.

En una reunión realizada el día martes 04 de octubre de 2005 con el Ing. Azurdia él expresó su preocupación ante dicho problema y a la vez planteo cuáles son los requerimientos que el sistema para control de trabajos de graduación debe cumplir. Estos requerimientos son detallados más adelante.

El beneficio será para los estudiantes, asesores y tutores, ya que tendrán un acceso rápido a la información y de esta forma se evitará que haya propuestas que sean rechazadas al conocer los temas de trabajos de graduación finalizados y los temas de protocolos vigentes.

A continuación se presentan datos estadísticos proporcionados por el Ing. Carlos Azurdia:

- Se hacen trabajos de graduación en la Escuela de Ciencias y Sistemas desde el año 1983.
- Protocolos por año:

○ Aprobados	65
○ Rechazados	65
○ Total presentados	130
- En promedio se finalizan 50 trabajos de graduación por año.
- El tiempo promedio para finalizar un trabajo de graduación es de 1 año.

2.3. Requerimientos

A continuación se detallan los requerimientos planteados por el Ing. Azurdia para el desarrollo del sistema de control de trabajos de graduación de la Escuela de Ciencias y Sistemas:

- Se debe conocer la información de los trabajos de graduación finalizados, esto incluye los datos del estudiante y asesor, el título, el contenido o índice y un resumen.
- Se debe conocer la información de los protocolos presentados por los estudiantes, esto incluye los datos del estudiante y asesor, el título, el contenido o índice, un resumen y el estado en el que se encuentra el protocolo (rechazado, aprobado, su vigencia, fase de revisión), también existe la posibilidad que el protocolo cambie su estado a trabajo finalizado. También se debe manejar un historial de todos los cambios que ha sufrido un protocolo.
- La información de los asesores debe estar disponible, datos personales, datos de contacto, si es catedrático actualmente en la Escuela de Ciencias y Sistemas, su *curriculum vitae*, los trabajos de graduación que ha asesorado y si actualmente se encuentra asesorando a uno o más estudiantes.

- Los roles que se deben manejar en la aplicación son:
 - Estudiante y asesor: Únicamente tienen privilegios de consulta.
 - Tutor: Maneja toda la información en el sistema, puede crear y cambiar el estado a los protocolos, agregar o modificar asesores y modificar estudiantes, un tutor no puede eliminar o crear estudiantes.
 - Administrador: Posee los mismos privilegios que un tutor, pero también puede crear y modificar usuarios y a la vez asignarles estudiante para tuturar.
 - Digitador: Únicamente puede ingresar datos al sistema, no puede hacer modificaciones ni eliminaciones. El usuario y contraseña de este rol es proporcionado por el administrador y puede ser activado y desactivado en cualquier momento.
- Se debe implementar la aplicación y proporcionar una carga inicial para poder utilizar el sistema. Para la recolección de los datos se utilizarán recursos proporcionados por la Escuela de Ciencias y Sistemas.
- El sistema debe ser implementado a través de una aplicación *Web* para que su acceso sea posible desde cualquier lugar.

2.4. Propuesta

Para implementar una solución que cumpla con los requerimientos descritos anteriormente, se utilizará la herramienta de desarrollo *Microsoft Visual Studio 2005* y se creará una aplicación *Web ASP.NET*. La aplicación se divide en dos módulos: Consultas y administración. Estos se describen a continuación.

2.4.1. Módulo de consultas

En este módulo los usuarios podrán realizar consultas para obtener información de los siguientes tipos:

- a. Trabajos de graduación: Se mostrará una lista de todos los documentos con los datos del estudiante y asesor, el título, el contenido o índice y un resumen. También se muestra la fase en la que se encuentra el documento: Trabajo con el asesor, aprobación del asesor, aprobación del revisor, revisión en lingüística I, aprobación del Director de Escuela, orden de impresión, revisión en lingüística II, trámites en Control Académico.
- b. Protocolo: Se mostrará una lista de todos los documentos con los datos del estudiante y asesor, el título, el contenido o índice y un resumen. También se muestra el estado del protocolo, el cual puede ser rechazado, aprobado o finalizado. En caso que el protocolo tenga el estado de aprobado se muestra la vigencia del mismo. Se puede conocer de la misma forma el historial de cambios de estado que ha sufrido el protocolo.

- c. Asesor: Se mostrará una lista de los datos personales de los asesores, si es catedrático actualmente en la Escuela de Ciencias y Sistemas, un *link* para descargar su *curriculum vitae*, se podrá obtener un listado de los trabajos de graduación que ha asesorado y los que se encuentra asesorando actualmente.

Sobre cada uno de estos tipos de información se pueden hacer búsquedas para filtrar la información de interés para el usuario.

2.4.2. Módulo de administración

En este módulo los usuarios pueden dar mantenimiento a los estudiantes que tienen asignados, a los asesores, a los protocolos y trabajos de graduación.

Las operaciones que se pueden realizar son:

- a. Mantenimiento de estudiantes: Se pueden crear, eliminar o actualizar estudiantes.
- b. Mantenimiento de asesores: Se pueden realizar las operaciones de agregar, modificar y eliminar asesores.
- c. Mantenimiento de protocolos y trabajos de graduación: Se pueden ingresar documentos, modificar o eliminar y cambiar su estado en cualquier momento.

- d. Mantenimiento de usuarios: Aquí se pueden crear, modificar o eliminar usuarios. También se asignan un rol a cada usuario y existe la opción de activar o desactivar la cuenta del usuario.

- e. Asignación: Se realiza la asignación de estudiantes a cada usuario para trabajar en los trabajos de graduación.

3. FASE DE DISEÑO

3.1. Diseño de la base de datos

A continuación se describe cada una de las entidades que están incluidas en el diagrama entidad relación que contiene la información de la aplicación. También se describe cada uno de los atributos de la entidad.

3.1.1. Entidades

- Estudiante: Almacena la información de los estudiantes que tienen al menos un documento en la aplicación.
 - Carné: Identificador único para cada estudiante.
 - Nombre: Nombre del estudiante.
 - Apellido: Apellido del estudiante.
 - Teléfono: Teléfono del estudiante.
 - Celular: Teléfono móvil del estudiante.
 - E-mail: Dirección de correo electrónico del estudiante.

- Asesor: Almacena la información de los asesores que han asesorado a uno o más estudiantes o se encuentran asesorando actualmente:
 - Código: Identificador único para cada asesor.
 - Nombre: Nombre del asesor.
 - Apellido: Apellido del asesor.
 - Teléfono: Teléfono del asesor.
 - Celular: Teléfono móvil del asesor.
 - E-mail: Dirección de correo electrónico del asesor.

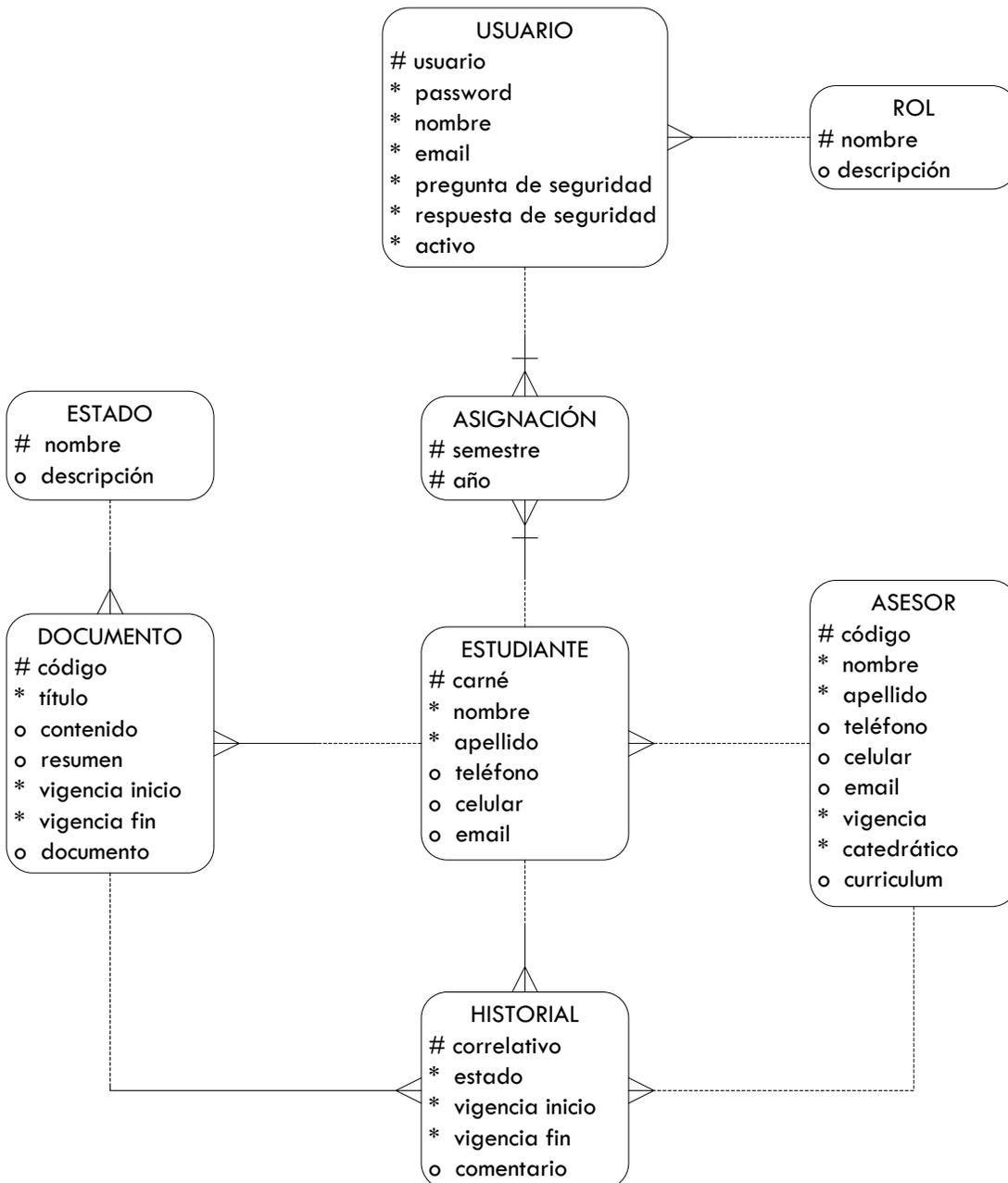
- Vigencia: Indica si el asesor esta se encuentra en vigencia para trabajar con un estudiante.
 - Catedrático: Indica si el asesor es actualmente catedrático de la Escuela de Ciencias y Sistemas o no.
 - *Curriculum*: Hace referencia a un documento que contiene el *curriculum vitae* del asesor.
- Documento: Almacena la información de los documentos presentados por los estudiantes.
 - Código: Identificador único para cada documento.
 - Título: Título del documento.
 - Contenido: Hace referencia al índice o contenido del documento.
 - Resumen: Hace referencia a un resumen que describe el documento.
 - Vigencia inicio: Indica la fecha en la que inicia la validez de un documento.
 - Vigencia fin: Indica la fecha en la que finaliza la validez de un documento.
 - Documento: Hace referencia al documento completo.
- Estado: Almacena los diferentes estado que pueden ser asignados a un documento.
 - Nombre: Nombre único del estado.
 - Descripción: Descripción del estado.
- Usuario: Almacena la información de los usuario que hacen uso del aplicación.
 - Usuario: Identificador único para cada usuario.
 - *Password*: Clave utilizada para seguridad del usuario.

- Nombre: Nombre real del usuario.
 - E-mail: Dirección de correo electrónico del usuario.
 - Pregunta de seguridad: Se utiliza para recuperar el *password* si el usuario lo solicita.
 - Respuesta de seguridad: Es la respuesta a la pregunta de seguridad.
 - Activo: Si utiliza para establecer si un usuario está activo o inactivo.
- Rol: Almacena los diferentes roles que pueden ser asignados a un usuario.
 - Nombre: Nombre del rol.
 - Descripción: Descripción del rol.
- Asignación: Almacena los diferentes roles que pueden ser asignados a un usuario.
 - Semestre: Número de semestre de la asignación.
 - Año: Año de la asignación.
- Historial: Almacena el historial de cambios que ha sufrido un documento, por ejemplo, cambio de estudiante, cambio de asesor, de título, de estado, etc.
 - Correlativo: Identificador único para cada entrada en el historial.
 - Estado: Estado del documento en el momento de la creación de la entrada en el historial.
 - Vigencia inicio: Fecha en la que inicia la validez de un documento.
 - Vigencia fin: Fecha en la que finaliza la validez de un documento.
 - Comentario: Observaciones de la entrada en el historial.

3.1.2. Diagrama entidad relación

En la figura 24 se muestra el diagrama entidad relación que contiene las entidades, atributos y los vínculos del sistema.

Figura 24. Diagrama entidad relación



3.2. Utilización de UML

A continuación se describe la funcionalidad de cada módulo por medio de los diagramas de UML.

3.2.1. Módulo de consultas

Las consultas se pueden realizar sobre los documentos, asesores y estudiantes. A continuación se muestran los diagramas de casos de uso, de secuencias y de actividades para este módulo.

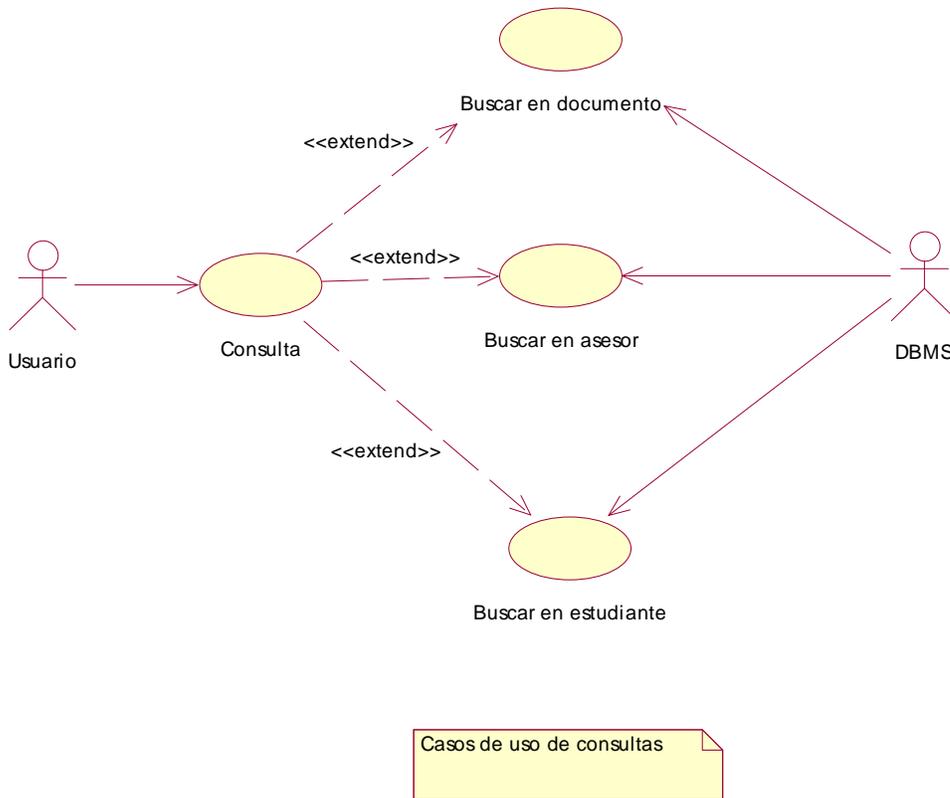
3.2.1.1. Diagrama de casos de uso de consultas

El módulo de consultas permite a los usuarios obtener información por medio de búsquedas en los datos almacenados. Las búsquedas pueden ser personalizadas por medio del uso de filtros con los que se indica de forma específica el tipo de información que se desea obtener.

Después de realizar una búsqueda el usuario tendrá la opción de ver la información completa del documento, asesor o estudiante que seleccione en el resultado que haya obtenido.

En la figura 25 se muestra el diagrama de casos de uso de consultas de la aplicación.

Figura 25. Diagrama de casos de uso de consultas



3.2.1.2. Diagrama de secuencias de consultas

A continuación se describe la secuencia que un usuario debe seguir para poder realizar una búsqueda:

- Se ingresa la búsqueda deseada.
- Se valida la información de la búsqueda.
- Se hace la consulta en la base de datos.
- Se retornan los resultados.
- Si hay resultados, se selecciona el documento, asesor o estudiante.
- Se busca la información en la base de datos.
- Se retorna la información

La figura 26 muestra el diagrama de secuencias de consultas.

Figura 26. Diagrama de secuencias de consultas

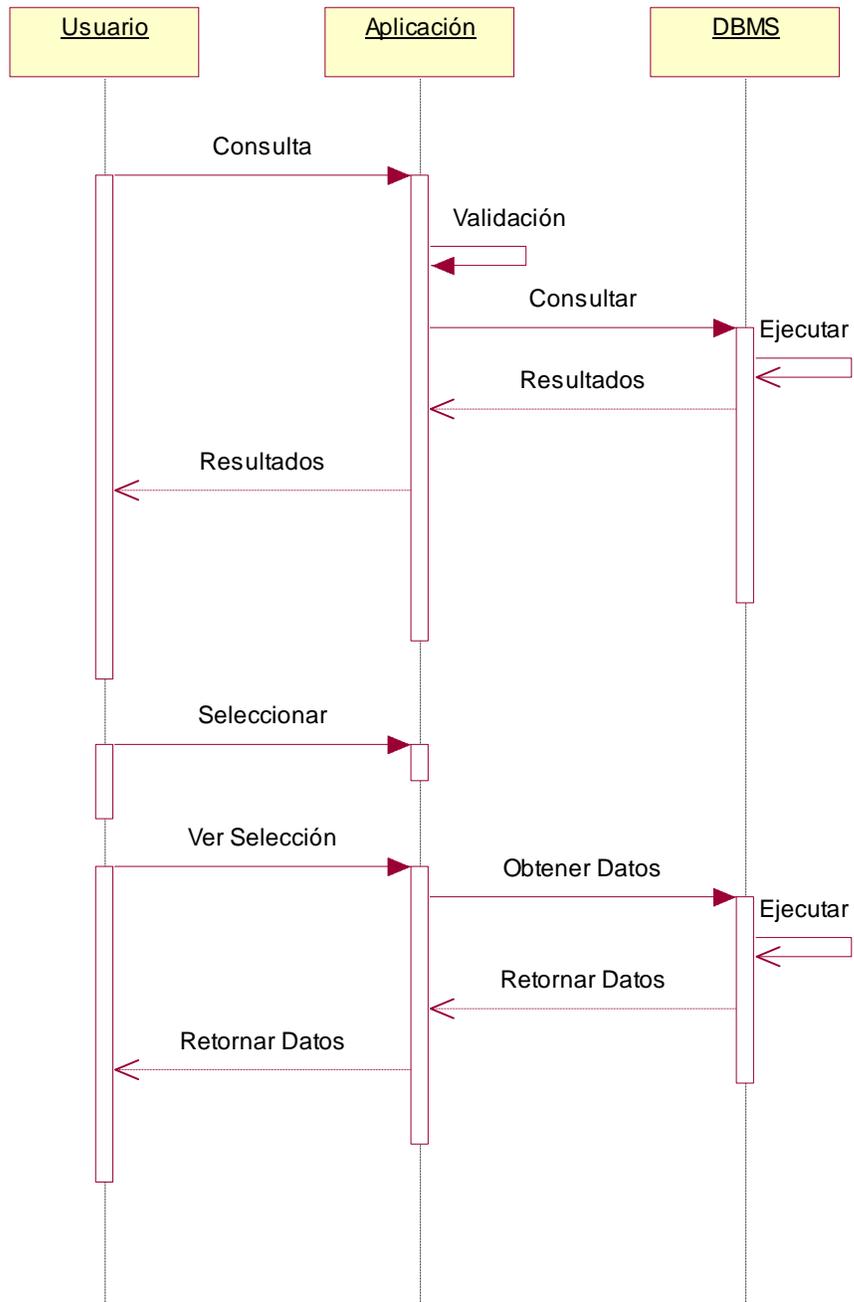
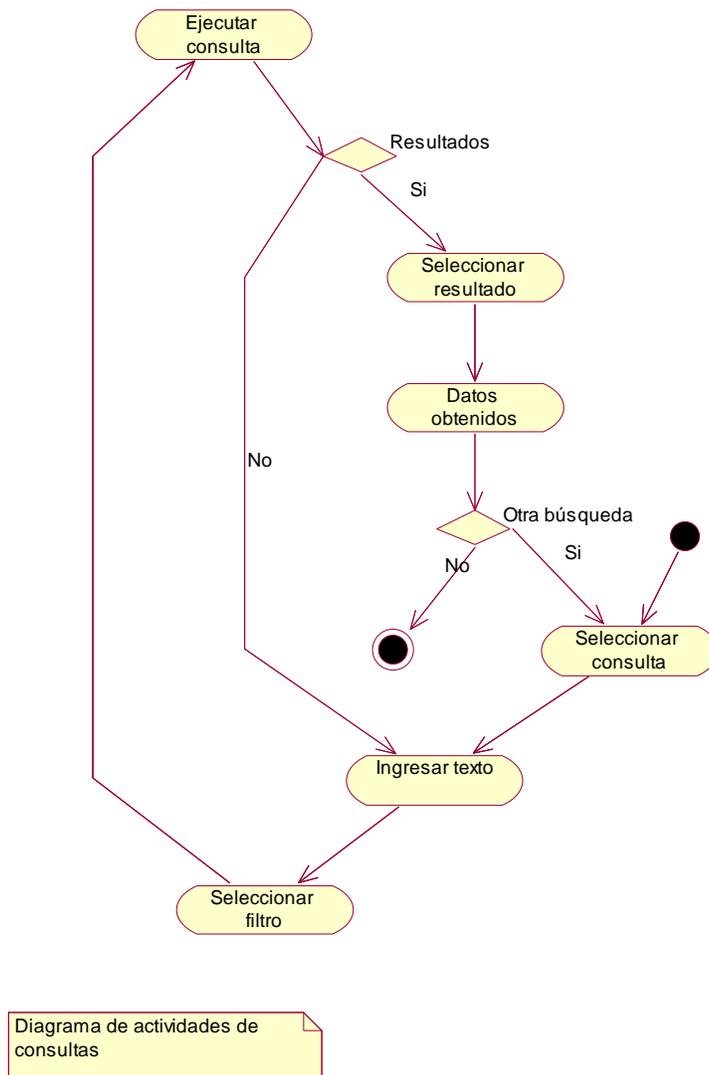


Diagrama de secuencias de consultas

3.2.1.3. Diagrama de actividades de consultas

En la figura 27 se muestra el diagrama de actividades que un usuario debe seguir para realizar una búsqueda en el sistema.

Figura 27. Diagrama de actividades de consultas

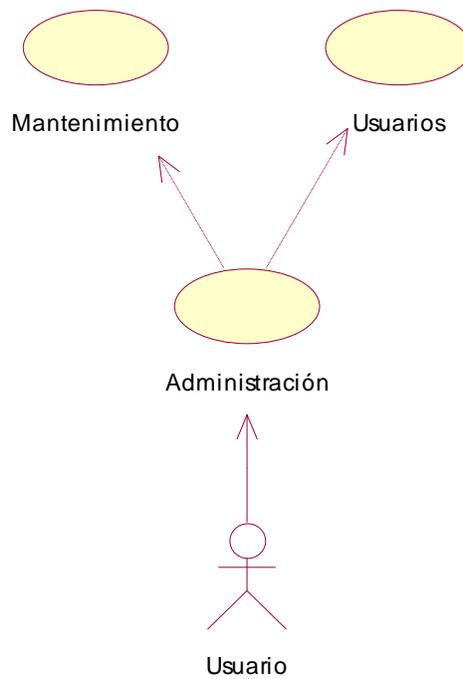


3.2.2. Módulo de administración

Este módulo se divide en dos casos de uso, el caso de uso de mantenimiento y el caso de uso de usuarios. En los siguientes apartados se explica en detalle cada uno de estos casos de uso y sus respectivos diagramas de secuencias y actividades.

En la figura 28 se muestra el diagrama de casos de uso del módulo de administración.

Figura 28. Diagrama de casos de uso de administración



Casos de uso de administración

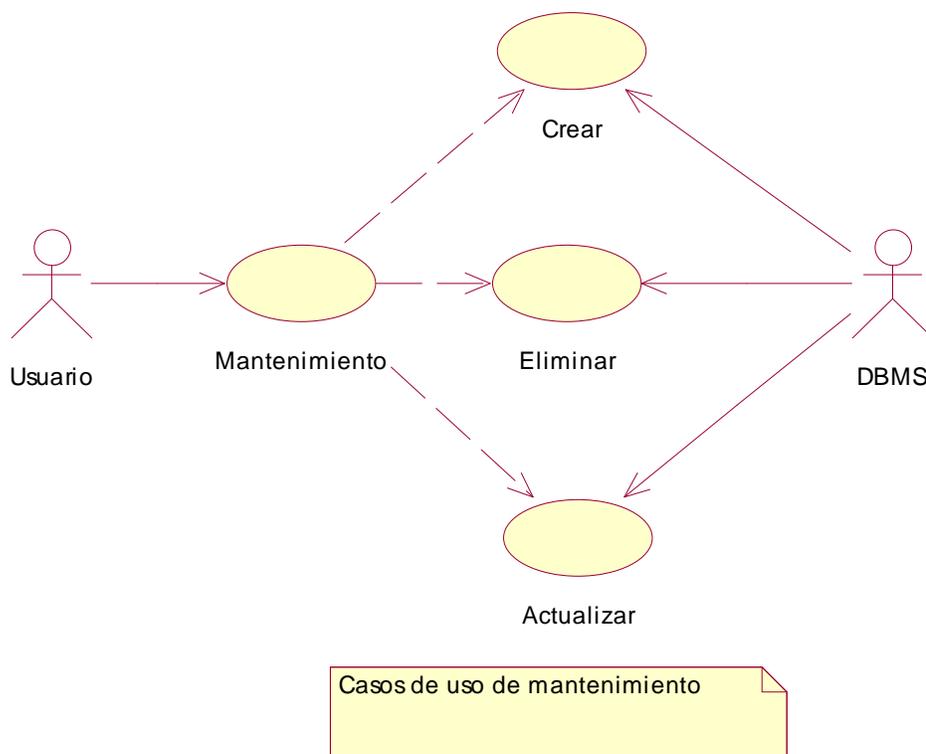
3.2.2.1. Diagrama de casos de uso de mantenimiento

El mantenimiento se puede realizar sobre las siguientes entidades:

- Documentos.
- Asesores.
- Estudiantes.

Para cada una se realizan operaciones de crear, eliminar y actualizar información en la base de datos. En la figura 29 se muestra el diagrama de casos de uso de mantenimiento.

Figura 29. Diagrama de casos de uso de mantenimiento



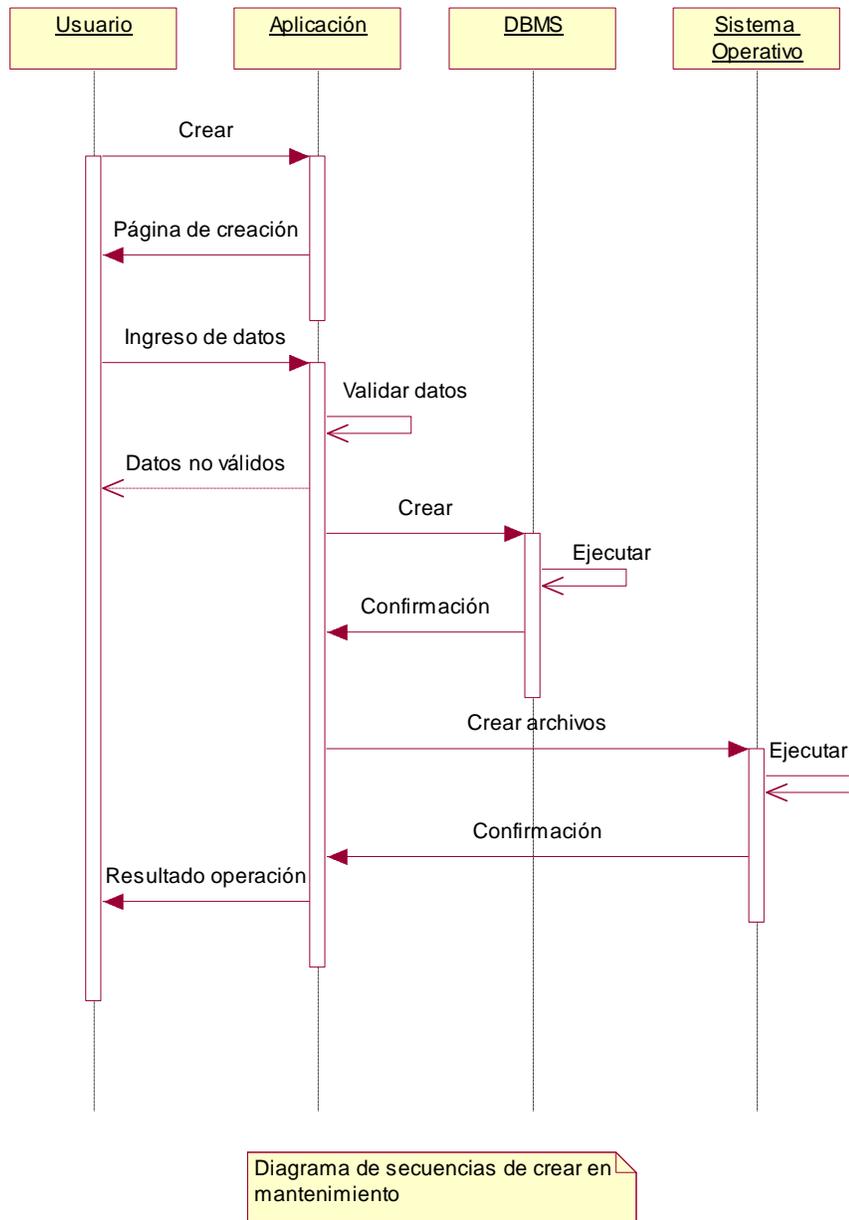
3.2.2.1.1. Diagrama de secuencias de crear en mantenimiento

A continuación se describe la secuencia que un usuario debe seguir para poder crear un documento, asesor o estudiante:

- El usuario debe ingresar los datos.
- La aplicación valida la información ingresada.
- Si los datos tienen algún valor no válido, se informa al usuario para haga las correcciones necesarias.
- Si los datos son válidos, la aplicación ejecuta los procedimientos necesarios para almacenar la información.
- Se muestra al usuario el resultado de la operación

En la figura 30 se muestra el diagrama de secuencias de crear en mantenimiento.

Figura 30. Diagrama de secuencias de crear en mantenimiento



3.2.2.1.2. Diagrama de actividades de crear en mantenimiento

En la figura 31 se muestra el diagrama de actividades que un usuario debe seguir para crear un documento, asesor o estudiante en la aplicación.

Figura 31. Diagrama de actividades de crear en mantenimiento

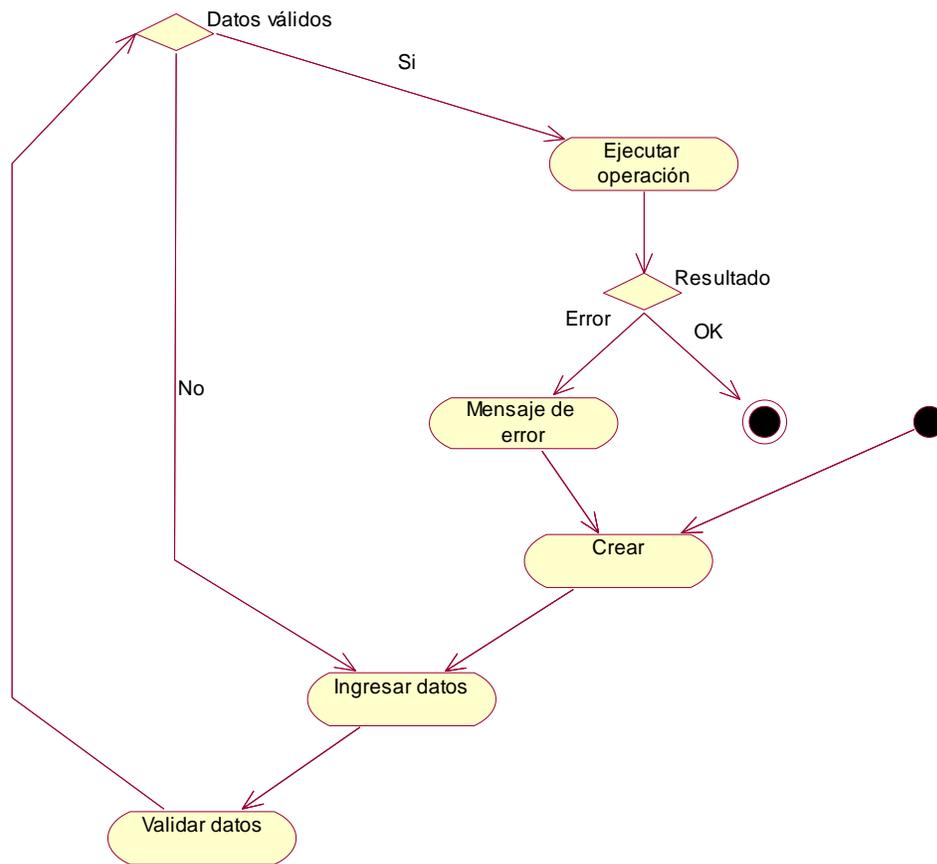


Diagrama de actividades de crear en mantenimiento

3.2.2.1.3. Diagrama de secuencias de eliminar en mantenimiento

A continuación se describe la secuencia que un usuario debe seguir para eliminar un documento, asesor o estudiante:

- La aplicación muestra al usuario un listado de los documentos, asesores o estudiantes.
- El usuario selecciona el documento, asesor o estudiante que desea eliminar.
- La aplicación envía una confirmación al usuario para realizar la eliminación.
- Si la respuesta del usuario es afirmativa la aplicación lleva a cabo la eliminación del documento, asesor o estudiante.
- Se muestra al usuario el resultado de la operación

En la figura 33 se muestra el diagrama de secuencias de eliminar en mantenimiento.

Figura 32. Diagrama de secuencias de eliminar en mantenimiento

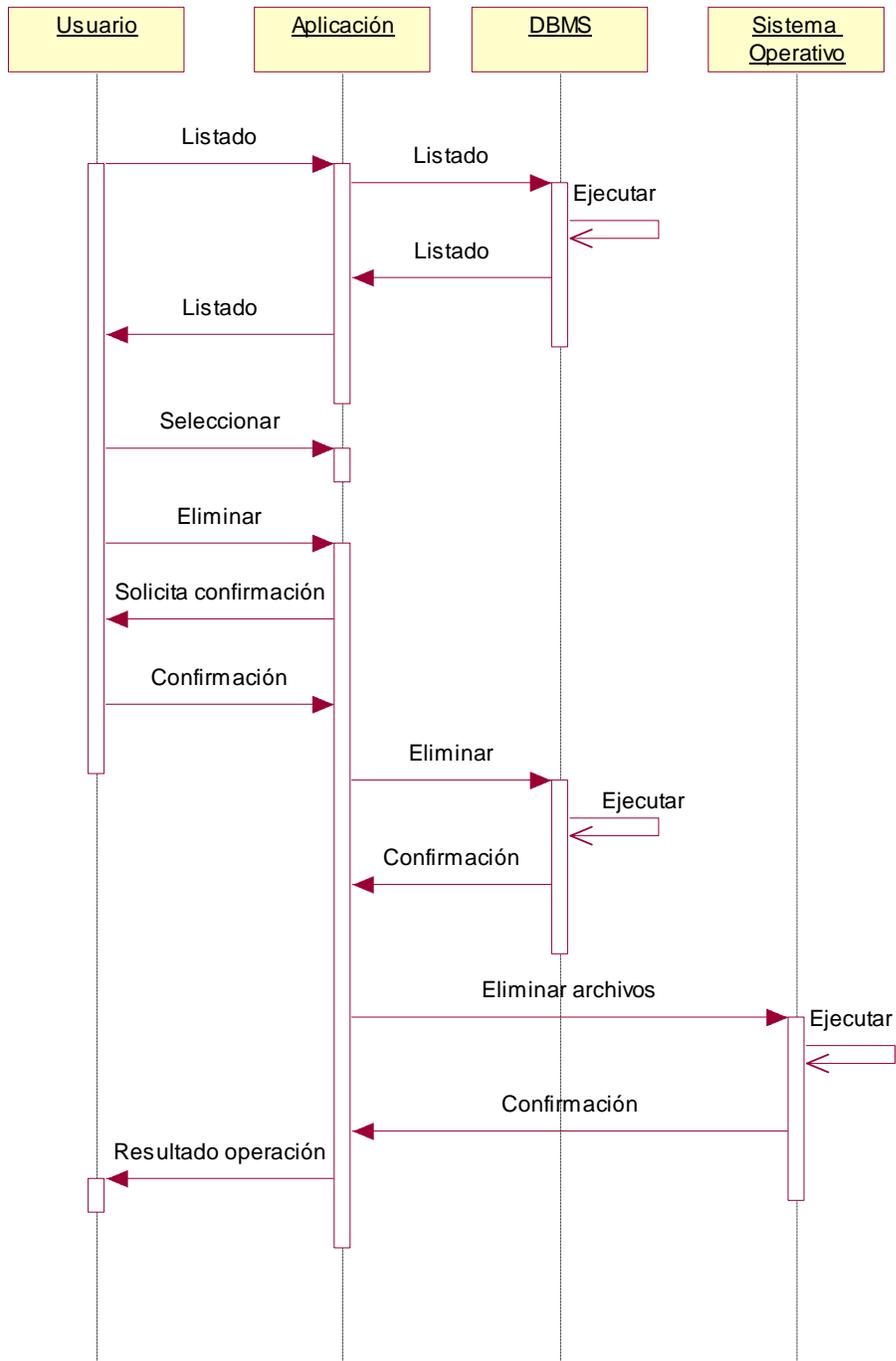


Diagrama de secuencias de eliminar en mantenimiento

3.2.2.1.4. Diagrama de actividades de eliminar en mantenimiento

En la figura 33 se muestra el diagrama de actividades que un usuario debe seguir para eliminar un documento, asesor o estudiante en la aplicación.

Figura 33. Diagrama de actividades de eliminar en mantenimiento

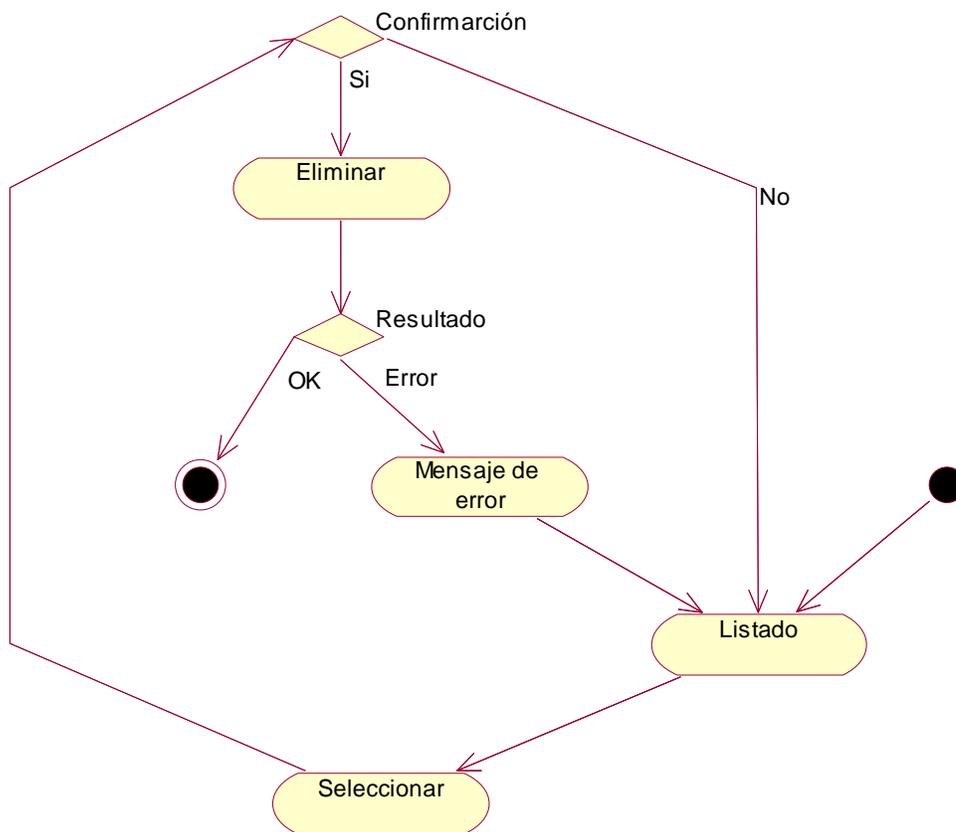


Diagrama de actividades de eliminar en mantenimiento

3.2.2.1.5. Diagrama de secuencias de actualizar en mantenimiento

A continuación se describe la secuencia que un usuario debe seguir para poder actualizar un documento, asesor o estudiante:

- La aplicación muestra al usuario un listado de los documentos, asesores o estudiantes.
- El usuario selecciona el documento, asesor o estudiante que desea actualizar.
- La aplicación genera una forma con los datos actuales y la presenta al usuario.
- El usuario debe modificar los datos deseados.
- La aplicación valida la información ingresada.
- Si los datos tienen algún valor no válido, se informa al usuario para haga las correcciones necesarias.
- Si los datos son válidos, la aplicación ejecuta los procedimientos necesarios para actualizar la información.
- Se muestra al usuario el resultado de la operación

En la figura 34 se muestra el diagrama de secuencias de actualizar en mantenimiento.

Figura 34. Diagrama de secuencias de actualizar en mantenimiento

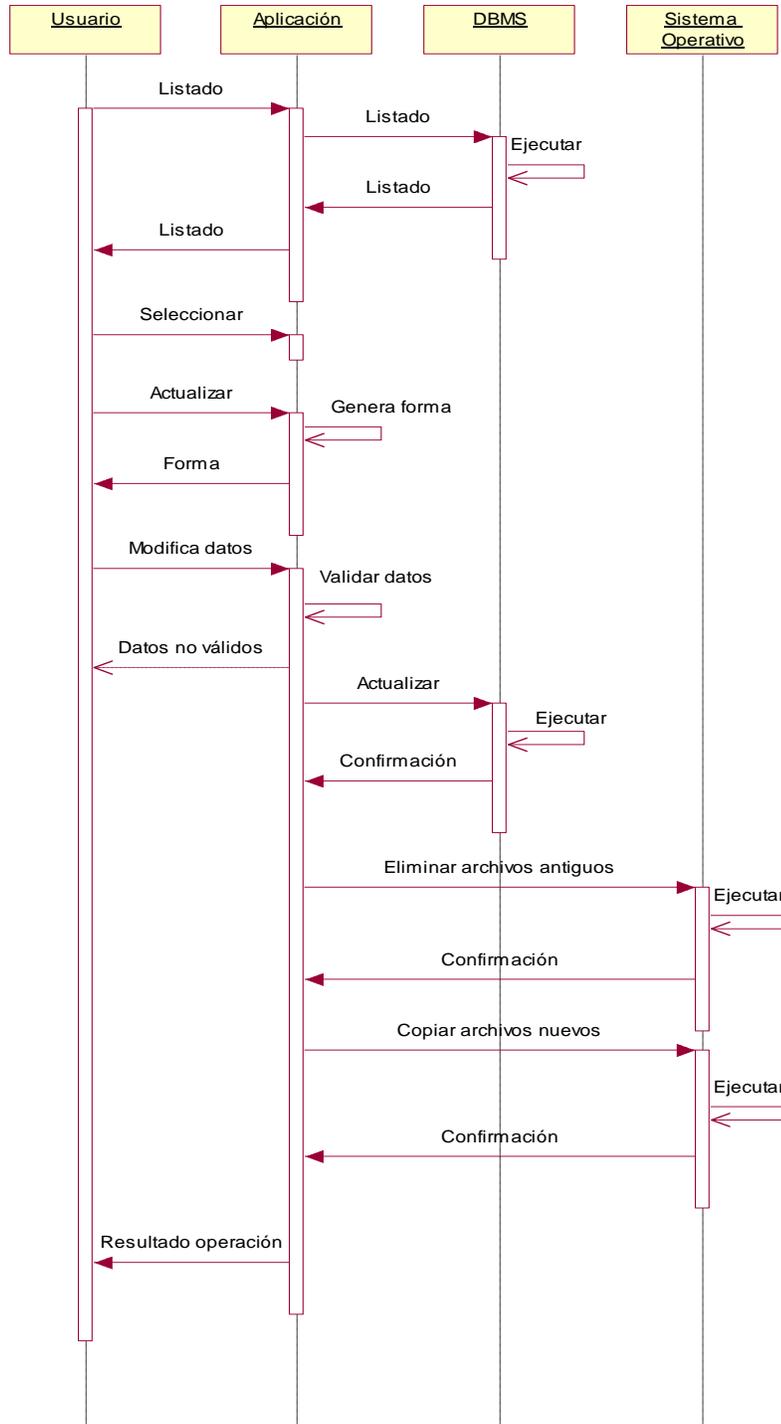


Diagrama de secuencias de actualizar en mantenimiento

3.2.2.1.6. Diagrama de actividades de actualizar en mantenimiento

En la figura 35 se muestra el diagrama de actividades que un usuario debe seguir para actualizar un documento, asesor o estudiante en la aplicación.

Figura 35. Diagrama de actividades de actualizar en mantenimiento

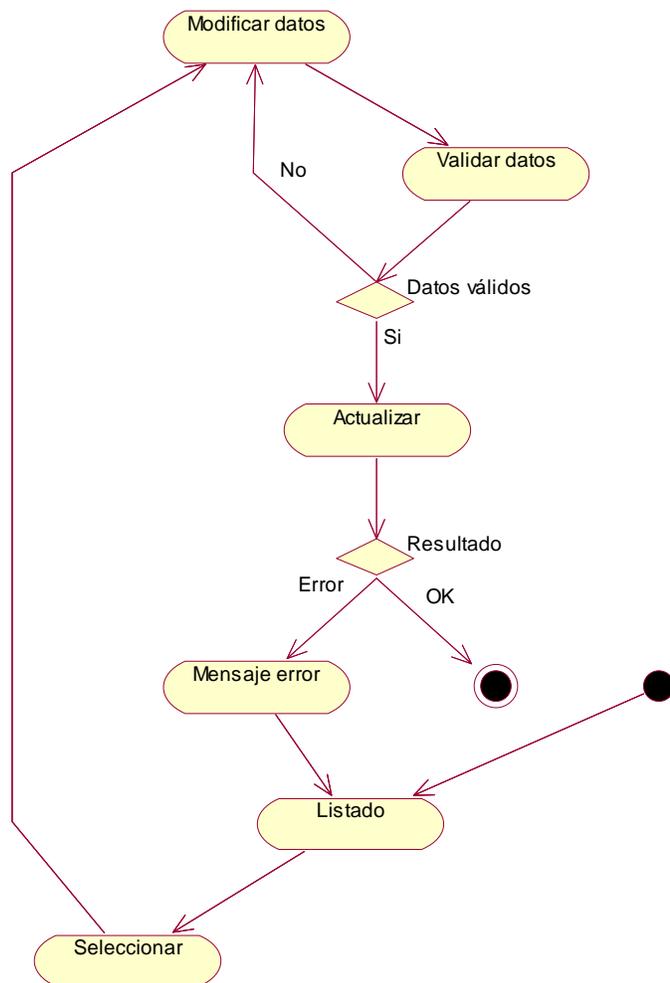


Diagrama de actividades de actualizar en mantenimiento

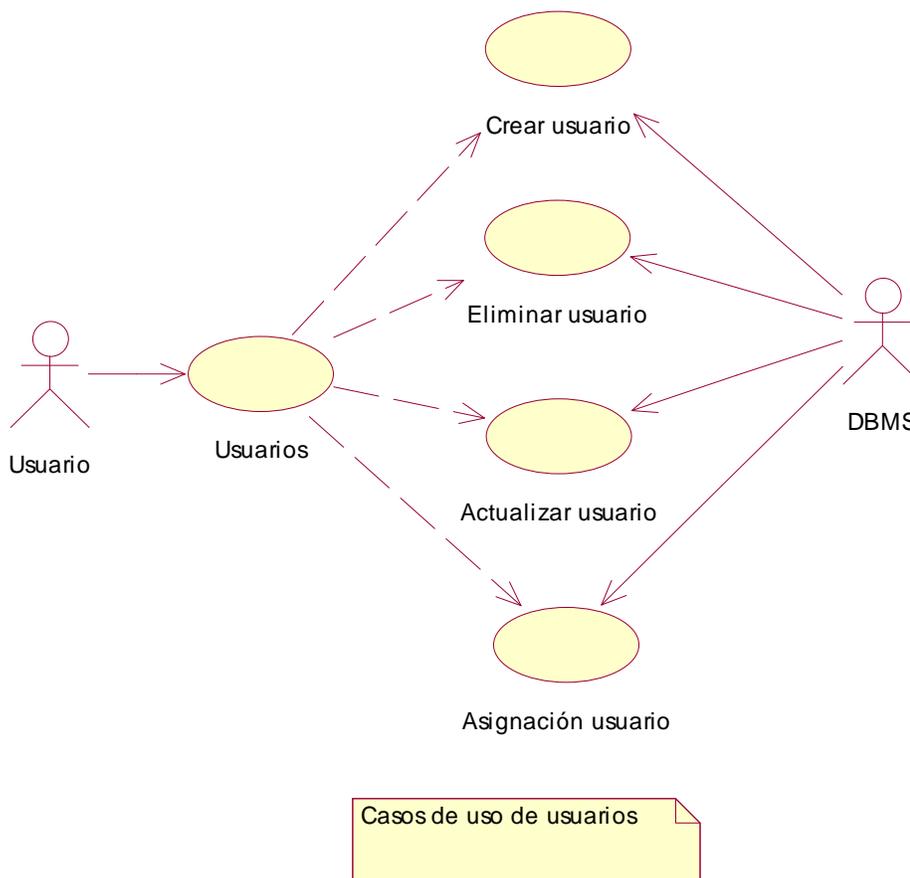
3.2.2.2. Diagrama de casos de uso de usuarios

Las operaciones que se pueden realizar sobre los usuarios son:

- Creación: Se crea un nuevo usuario en el sistema.
- Eliminación: Se elimina un usuario del sistema.
- Actualización: Se modifica un usuario existente en el sistema.
- Asignación: Se asignan o desasignan estudiantes de un usuario.

En la figura 36 se muestra el diagrama de casos de uso de usuarios.

Figura 36. Diagrama de casos de uso de usuarios



3.2.2.2.1. Diagrama de secuencias de crear usuario

A continuación se describe la secuencia que un usuario debe seguir para poder crear un nuevo usuario:

- El usuario debe ingresar los datos.
- La aplicación valida la información ingresada.
- Si los datos tienen algún valor no válido, se informa al usuario para haga las correcciones necesarias.
- Si los datos son válidos, la aplicación ejecuta los procedimientos necesarios para almacenar la información.
- Se muestra al usuario el resultado de la operación

En la figura 37 se muestra el diagrama de secuencias de crear usuario.

Figura 37. Diagrama de secuencias de crear usuario

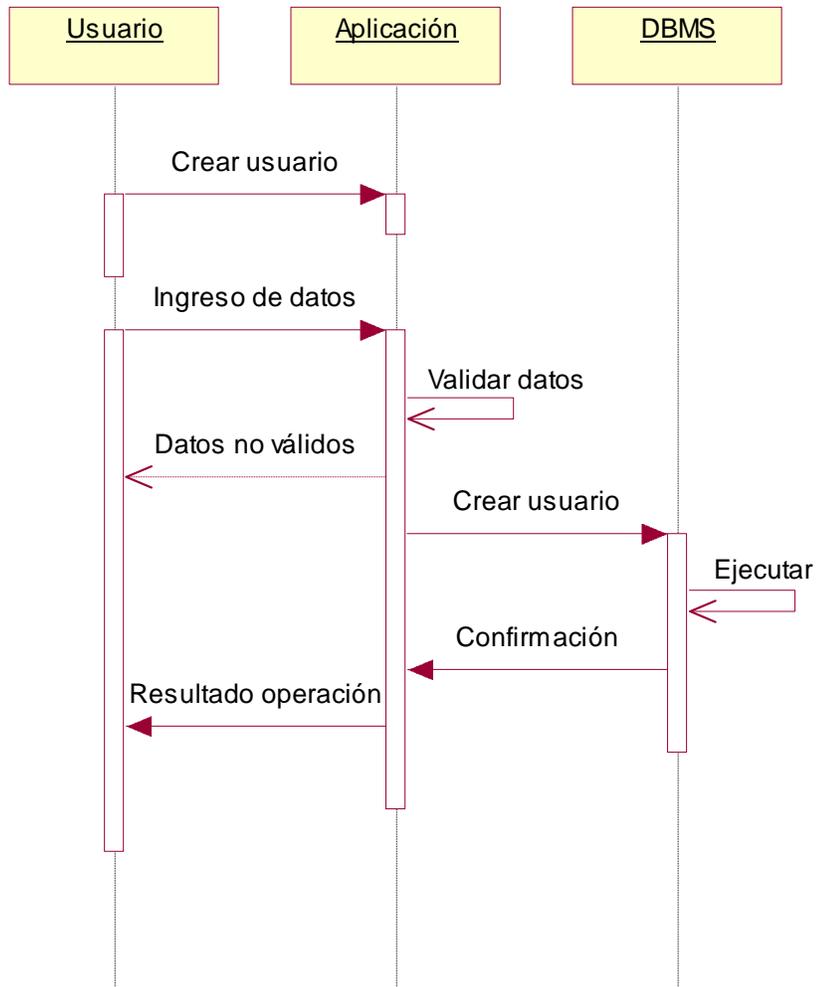


Diagrama de secuencias de crear usuario

3.2.2.2. Diagrama de actividades de crear usuario

En la figura 38 se muestra el diagrama de actividades que un usuario debe seguir para crear un nuevo usuario en la aplicación.

Figura 38. Diagrama de actividades de crear usuario

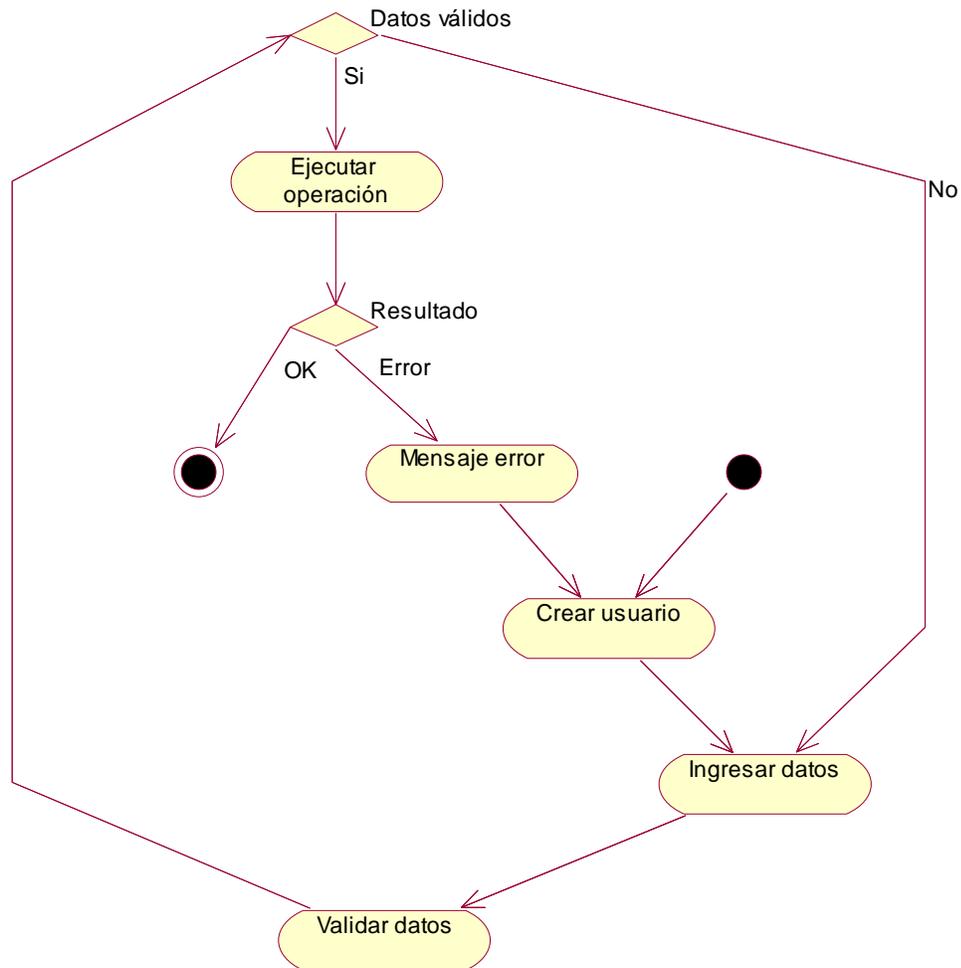


Diagrama de actividades de crear usuario

3.2.2.2.3. Diagrama de secuencias de eliminar usuario

A continuación se describe la secuencia que un usuario debe seguir para poder eliminar un usuario de la aplicación:

- La aplicación muestra al usuario un listado de los usuarios existentes.
- El usuario selecciona el usuario que desea eliminar.
- La aplicación envía una confirmación al usuario para realizar la eliminación.
- Si la respuesta del usuario es afirmativa la aplicación lleva a cabo la eliminación del usuario.
- Se muestra al usuario el resultado de la operación.

En la figura 39 se muestra el diagrama de secuencias de eliminar usuario.

Figura 39. Diagrama de secuencias de eliminar usuario

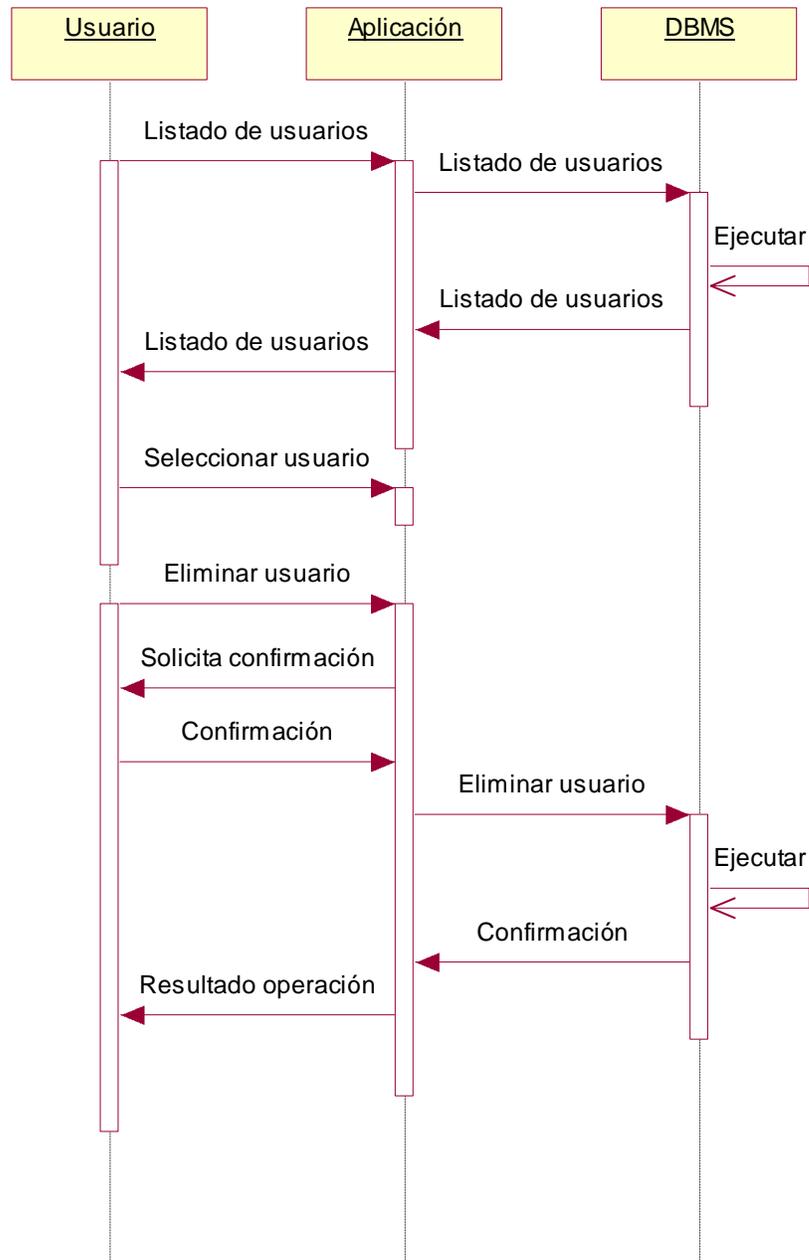
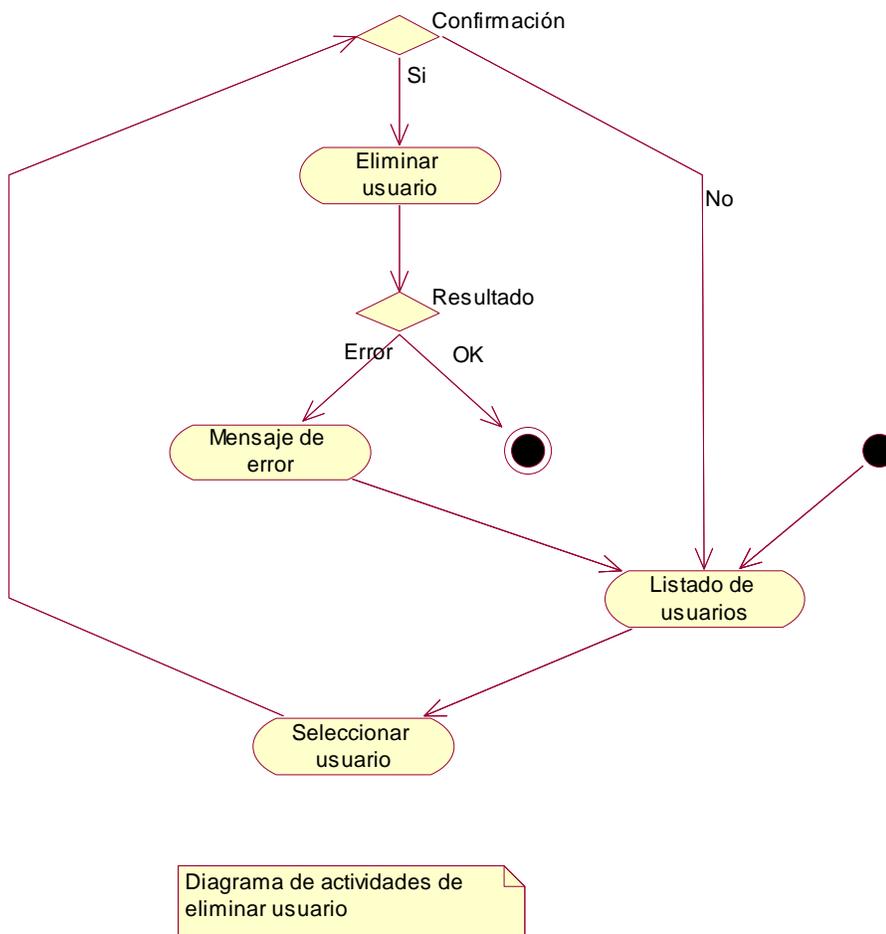


Diagrama de secuencias de eliminar usuario

3.2.2.4. Diagrama de actividades de eliminar usuario

En la figura 40 se muestra el diagrama de actividades que un usuario debe seguir para eliminar un usuario existente en la aplicación.

Figura 40. Diagrama de actividades de eliminar usuario



3.2.2.2.5. Diagrama de secuencias de actualizar usuario

A continuación se describe la secuencia que un usuario debe seguir para poder actualizar la información de un usuario de la aplicación:

- La aplicación muestra al usuario un listado de los usuarios existentes.
- El usuario selecciona el usuario que desea actualizar.
- La aplicación genera una forma con los datos actuales y la presenta al usuario.
- El usuario debe modificar los datos deseados.
- La aplicación valida la información ingresada.
- Si los datos tienen algún valor no válido, se informa al usuario para haga las correcciones necesarias.
- Si los datos son válidos, la aplicación ejecuta los procedimientos necesarios para actualizar la información.
- Se muestra al usuario el resultado de la operación

En la figura 41 se muestra el diagrama de secuencias de actualizar usuario.

Figura 41. Diagrama de secuencias de actualizar usuario

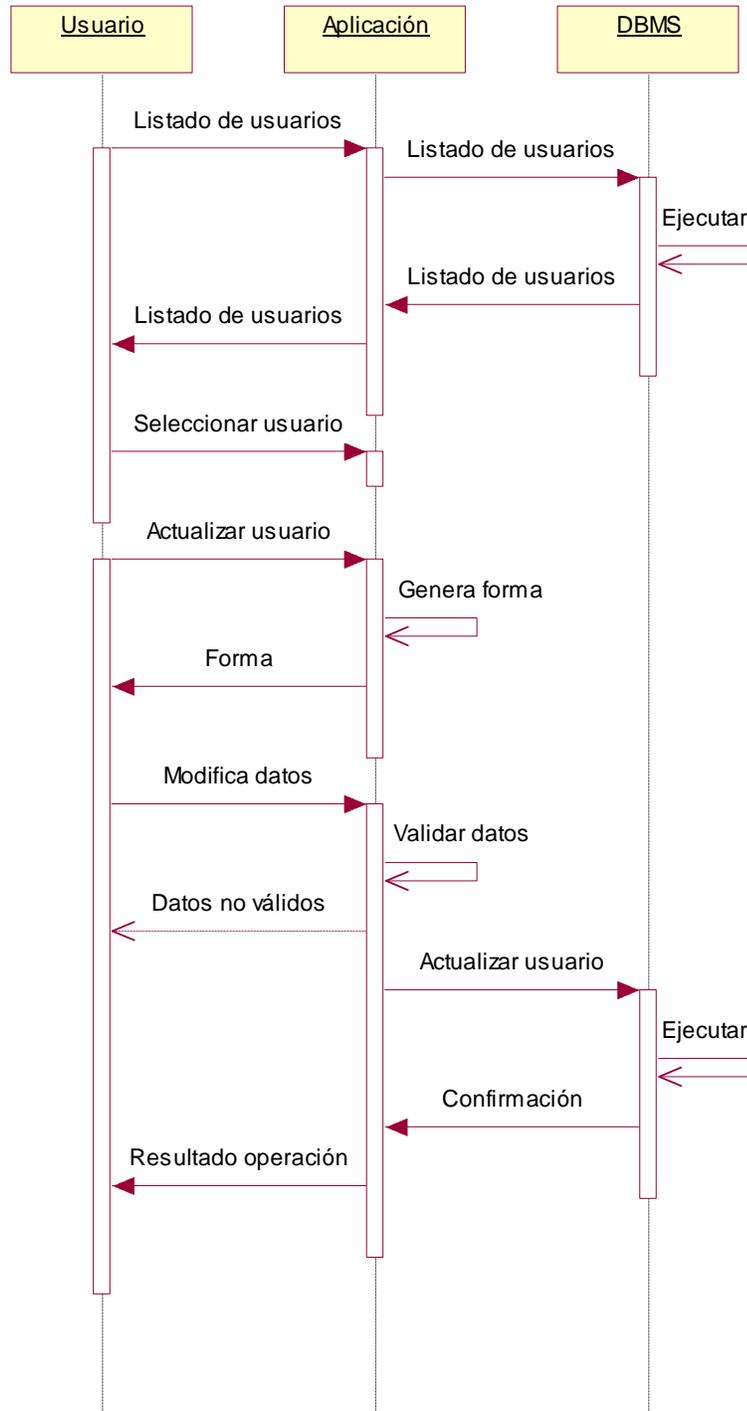


Diagrama de secuencias de actualizar usuario

3.2.2.2.6. Diagrama de actividades de actualizar usuario

En la figura 42 se muestra el diagrama de actividades que un usuario debe seguir para actualizar un usuario existente en la aplicación.

Figura 42. Diagrama de actividades de actualizar usuario

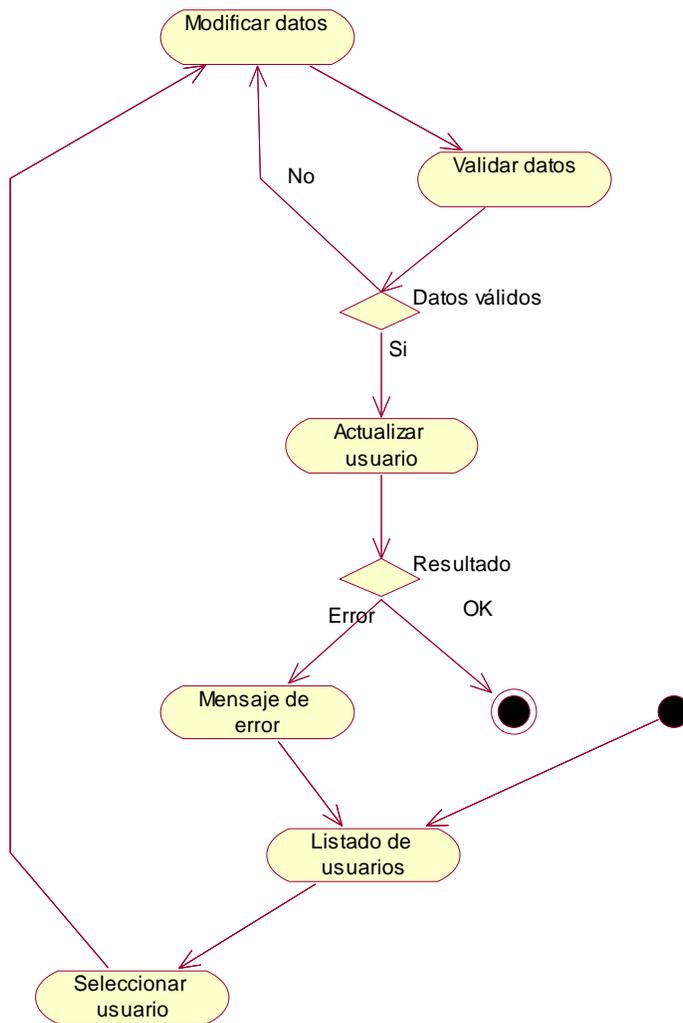


Diagrama de actividades de actualizar usuario

3.2.2.2.7. Diagrama de secuencias de asignación de usuarios

A continuación se describe la secuencia que un usuario debe seguir para poder actualizar la información de un usuario de la aplicación:

- Se debe seleccionar el usuario al que se le asignaran o desasignaran estudiantes.
- La aplicación muestra al usuario un listado de los estudiantes existentes.
- El usuario selecciona los estudiantes que desea asignar al usuario previamente seleccionado.
- Se realiza la asignación o desasignación en la base de datos.
- Se muestra al usuario el resultado de la operación

En la figura 43 se muestra el diagrama de secuencias de asignación de usuarios.

Figura 43. Diagrama de secuencias de asignación de usuarios

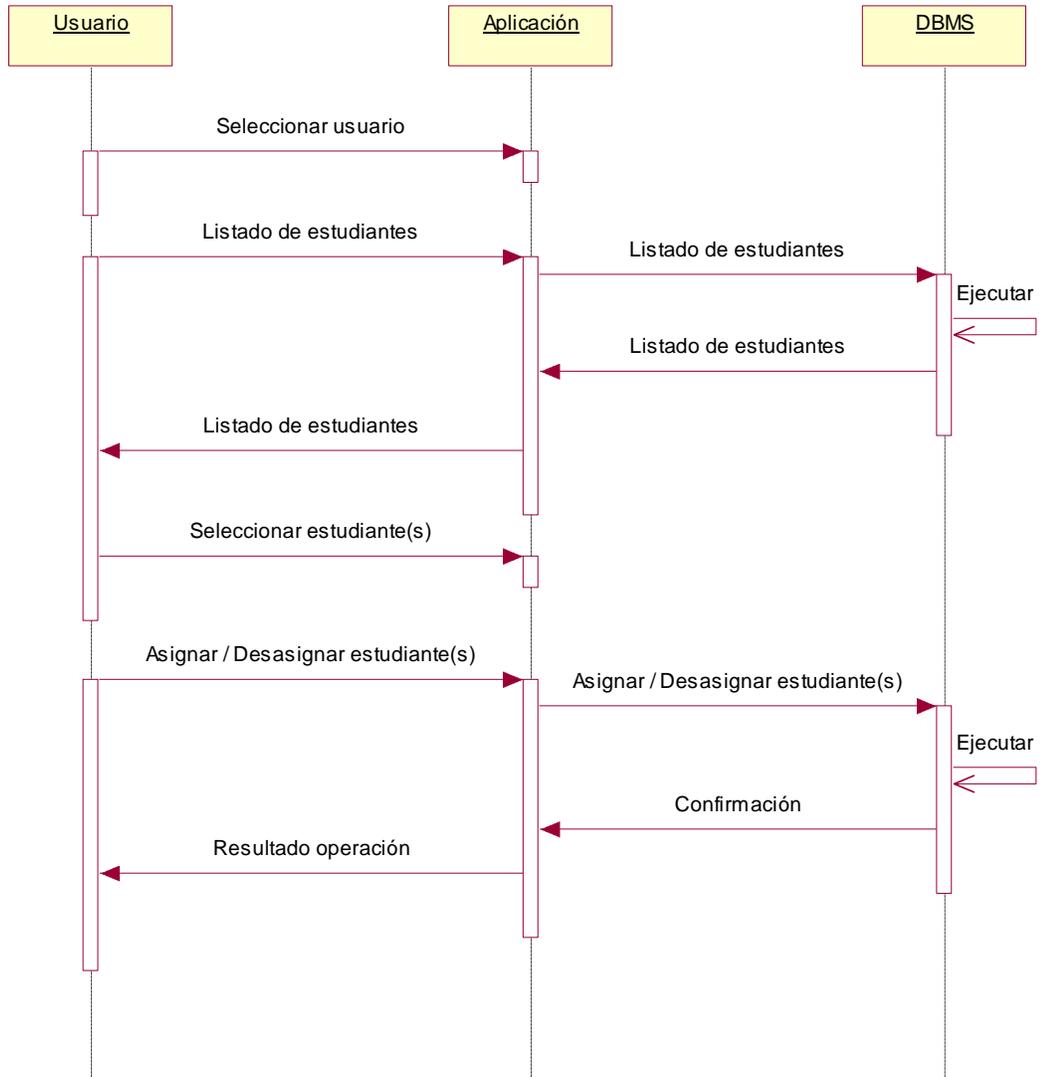


Diagrama de secuencias de asignación de usuarios

3.2.2.2.8. Diagrama de actividades de asignación de usuarios

En la figura 44 se muestra el diagrama de actividades que un usuario debe seguir para realizar una asignación de estudiantes a un usuario existente en la aplicación.

Figura 44. Diagrama de actividades de asignación de usuarios

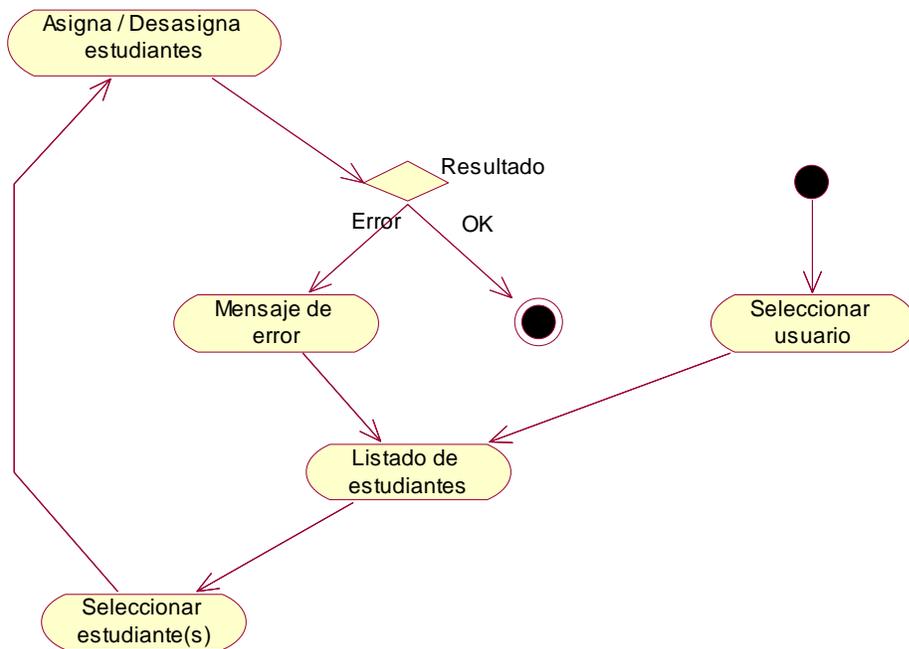


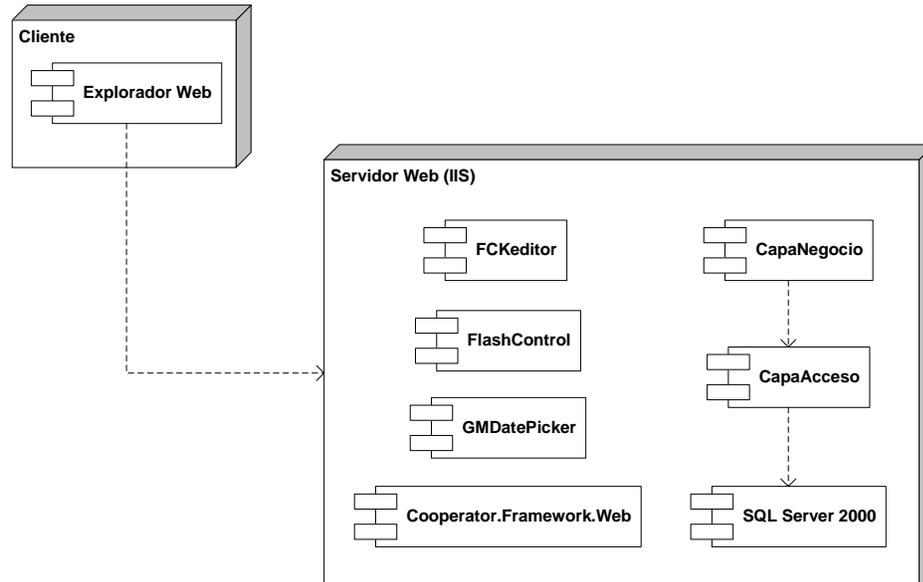
Diagrama de secuencias de asignación de usuarios

3.2.3. Diagrama de distribución

Este diagrama muestra la forma en que se encuentran distribuidos los componentes de la aplicación en el hardware disponible.

En la figura 45 se muestra el diagrama de distribución y cada uno de sus componentes.

Figura 45. Diagrama de distribución de la aplicación



A continuación se describen los componentes que forman parte del servidor *Web* en el que se encuentra la aplicación:

- *FCkeditor*: Proporciona un editor de texto HTML que brinda la facilidad de dar formato al texto en el ambiente *Web*.
- *FlashControl*: Es un componente gratuito que permite agregar una película *Flash* a los proyectos *ASP.NET 2.0*.

- *GMDatePicker*: Este componente se utiliza para seleccionar fechas desde un calendario gráfico y luego insertar el valor en un cuadro de texto o etiqueta.
- *Cooperator.Framework.Web*: Por medio de este componente se pueden mostrar alertas y confirmaciones en el ambiente *Web*.
- *CapaNegocio*: En este componente se encuentra toda la información del negocio de la aplicación, por ejemplo: Creación de usuarios, eliminación de asesores, actualización de documentos, etc.
- *CapaAcceso*: Este es el componente que se comunica directamente con el manejador de la base de datos. En él se encapsulan los métodos para poder obtener, insertar, eliminar y actualizar los datos almacenados.
- *SQL Server 2000*: Es el manejador de la base de datos, se encarga de administrar la forma en que son almacenados y manipulados los datos.

A continuación se describen los componentes que forman parte del cliente que tiene acceso a la aplicación:

- *Explorador Web*: El explorador proporciona a los clientes la herramienta para poder navegar en la aplicación y así poder tener acceso a la información almacenada.

4. MANUALES

4.1. Manual de usuario

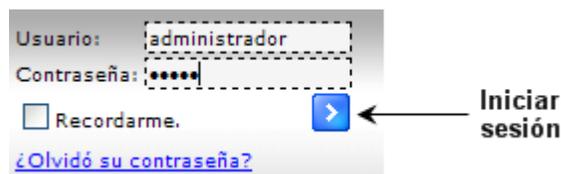
La aplicación maneja tres roles predefinidos que son: Administrador, Tutor y Digitador. Cada rol tiene permisos diferentes para poder trabajar en la aplicación. El rol con menos privilegios en la aplicación es el rol de Digitador, que únicamente puede ingresar datos; luego sigue el rol de Tutor que puede administrar documentos, estudiantes (que han sido asignados a su usuario) y administrar asesores, también puede realizar todas las tareas del rol Digitador, a excepción de las opciones “Crear estudiante” y “Cargar estudiantes”; finalmente el rol Administrador tiene todos los privilegios sobre la aplicación.

Antes de detallar cada rol, se explica cómo iniciar sesión, modificar los datos de los usuarios y cerrar sesión.

4.1.1. Iniciar sesión

Para iniciar sesión debe ingresar su usuario y contraseña en el cuadro que se encuentra en la esquina superior derecha de la aplicación, y luego dar un clic en el botón de iniciar sesión, ver figura 46.

Figura 46. Cuadro para inicio de sesión



Usuario: administrador
Contraseña: ●●●●
 Recordarme.  ← Iniciar sesión
[¿Olvidó su contraseña?](#)

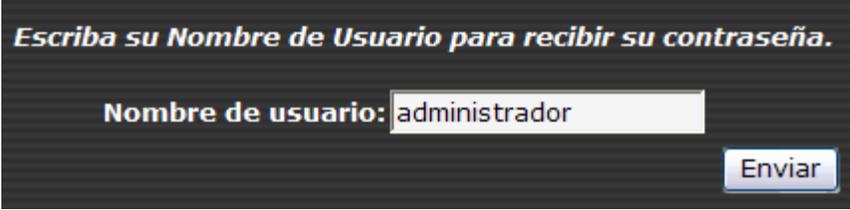
Si desea que la próxima vez que utilice la aplicación no sea necesario ingresar su usuario y contraseña debe marcar la opción “Recordarme”. Para que la opción tenga efecto no debe cerrar su sesión al terminar de utilizar la aplicación.

A continuación se explican los pasos a seguir si ha olvidado su contraseña.

4.1.1.1. ¿Olvidó su contraseña?

Si ha olvidado su contraseña debe dar clic en el vínculo que se encuentra en el cuadro de inicio de sesión (ver figura 46), luego se mostrará en pantalla un cuadro en el que deberá escribir su nombre de usuario como se muestra en la figura 47, y luego dar clic en el botón “Enviar”.

Figura 47. ¿Olvidó su contraseña? Paso 1



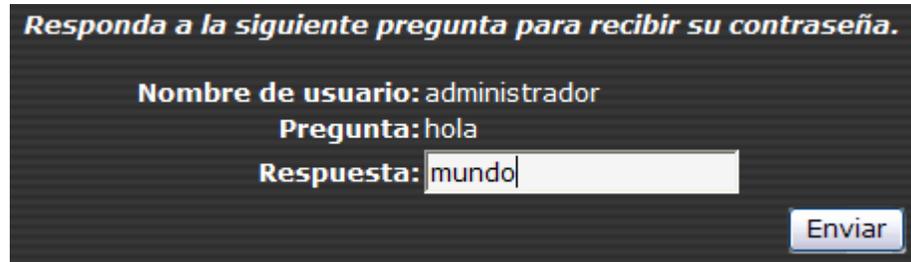
Escriba su Nombre de Usuario para recibir su contraseña.

Nombre de usuario:

Enviar

En el siguiente paso debe escribir la respuesta a la pregunta de seguridad que fue proporcionada en el momento de la creación del usuario y luego dar clic en el botón “Enviar”, ver figura 48.

Figura 48. ¿Olvidó su contraseña? Paso 2



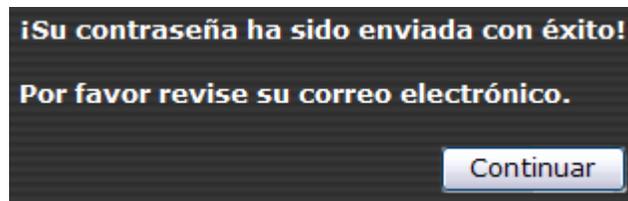
Responda a la siguiente pregunta para recibir su contraseña.

Nombre de usuario: administrador
Pregunta: hola
Respuesta:

Enviar

Si la respuesta es correcta se muestra una confirmación para indicar que la operación se realizó con éxito (ver figura 49). Luego de ver la confirmación deber revisar el correo electrónico que le fue enviado e ingresar sus datos en el cuadro de inicio de sesión. Si da clic en el botón “Continuar” se mostrará la página principal.

Figura 49. ¿Olvidó su contraseña? Paso 3



¡Su contraseña ha sido enviada con éxito!
Por favor revise su correo electrónico.

Continuar

4.1.2. Modificar cuenta

Para modificar su cuenta debe iniciar sesión primero y luego seleccionar la opción “Modificar mi cuenta” que aparece en la esquina superior derecha de la aplicación (en donde se encontraba el cuadro de inicio de sesión), ver figura 50.

Figura 50. Opciones para usuarios que han iniciado sesión



Bienvenido: administrador
[Modificar mi cuenta](#)
[Cerrar sesión](#)

Luego de seleccionar la opción de modificar cuenta se mostrará una página con tres opciones:

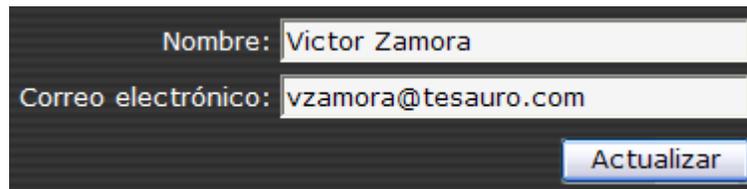
1. Editar mis datos personales.
2. Cambiar mi contraseña.
3. Cambiar pregunta y respuesta de seguridad.

A continuación se detalla cada una de las opciones.

4.1.2.1. Editar mis datos personales

Aquí puede cambiar su nombre real (no su nombre de usuario), y su dirección de correo electrónico. Cuando haya realizado los cambios debe dar clic en el botón “Actualizar”. Ver figura 51.

Figura 51. **Editar mis datos personales**



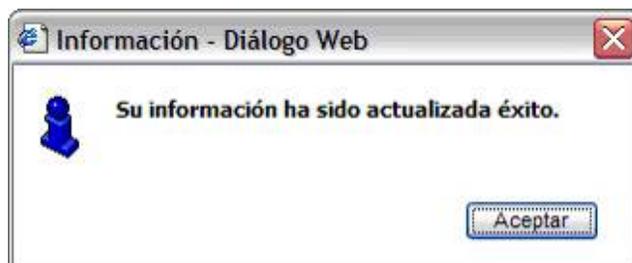
Nombre: Victor Zamora

Correo electrónico: vzamora@tesauro.com

Actualizar

Si la operación se realiza con éxito se mostrará un cuadro de diálogo como el que se puede ver en la figura 52.

Figura 52. **Confirmación de datos personales actualizados**



4.1.2.2. Cambiar mi contraseña

Aquí puede cambiar su contraseña. Para esto debe escribir su contraseña actual, su nueva contraseña y la confirmación de su nueva contraseña, luego debe dar clic en el botón “Cambiar contraseña”. Ver figura 53.

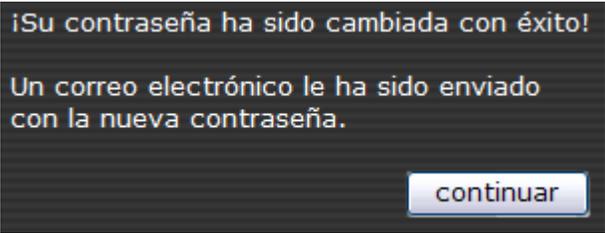
Figura 53. Cambiar mi contraseña



El formulario tiene un fondo oscuro con el título "Cambiar mi Contraseña" en blanco. Incluye tres campos de entrada de texto con marcadores de posición de puntos grises para ocultar el contenido: "Contraseña:", "Nueva contraseña:" y "Confirmar la nueva contraseña:". Debajo de los campos hay un botón rectangular con el texto "Cambiar contraseña".

Si los datos son correctos se mostrara una confirmación como en la figura 54. La próxima vez que inicie sesión deberá utilizar la nueva contraseña.

Figura 54. Confirmación de contraseña cambiada

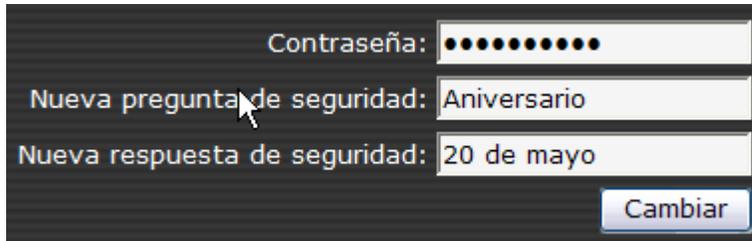


El mensaje de confirmación tiene un fondo oscuro con el texto en blanco. El primer renglón dice "¡Su contraseña ha sido cambiada con éxito!". El segundo renglón dice "Un correo electrónico le ha sido enviado con la nueva contraseña.". Debajo del texto hay un botón rectangular con el texto "continuar".

4.1.2.3. Cambiar pregunta y respuesta de seguridad

Aquí puede cambiar su pregunta y respuesta de seguridad, es necesario que escriba su contraseña y luego ingrese los nuevos datos. Cuando haya realizado los cambios debe dar clic en el botón “Cambiar”. Ver figura 55.

Figura 55. **Cambiar pregunta y respuesta de seguridad**



Contraseña: ●●●●●●●●

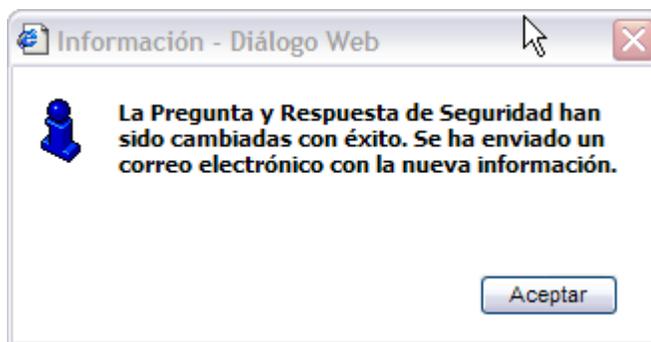
Nueva pregunta de seguridad: Aniversario

Nueva respuesta de seguridad: 20 de mayo

Cambiar

Si la operación se realiza con éxito se mostrará un cuadro de diálogo como el que se puede ver en la figura 56.

Figura 56. **Confirmación de datos personales actualizados**



4.1.3. Cerrar sesión

Para cerrar su sesión debe dar clic en la opción “Cerrar sesión” que se encuentra en la esquina superior derecha de la aplicación. Puede ingresar nuevamente en cualquier momento.

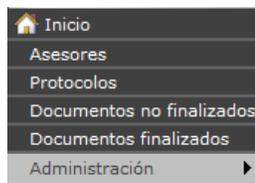
Recuerde que si ha marcado “Recordarme” al iniciar sesión no debe cerrar su sesión para que opción funciones correctamente.

A continuación se detalla el uso de la aplicación para cada rol y se explican los permisos que posee.

4.1.4. Rol digitador

El rol digitador posee únicamente permiso para crear asesores, estudiantes o documentos en la aplicación. Para ingresar a las opciones del rol digitador haga clic en el elemento “Administración” del menú que se encuentra al lado izquierdo de la aplicación como se muestra en la figura 57.

Figura 57. Elemento “Administración” del menú



En la pantalla se mostrarán las acciones permitidas para el rol digitador:

- Crear asesores.
- Crear estudiantes.
- Cargar estudiantes.
- Crear documentos.

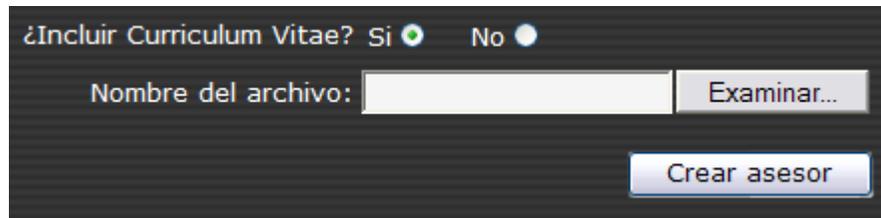
A continuación se describe cada una de las acciones para el rol digitador:

4.1.4.1. Crear asesores

Aquí puede crear los asesores que forman parte del sistema y trabajan directamente con los estudiantes durante el desarrollo de su trabajo de graduación, los datos requeridos se describen a continuación:

- Nombre: Este campo es obligatorio, debe escribir el nombre del asesor.
- Apellido: Este campo es obligatorio, debe escribir el apellido del asesor.
- Teléfono: Este campo es opcional, debe escribir el número de teléfono del asesor. Se pueden utilizar únicamente números.
- Celular: Este campo es opcional, debe escribir el número de celular o teléfono móvil del asesor. Se pueden utilizar únicamente números.
- Correo electrónico: Este campo es opcional, debe escribir una dirección de correo electrónico válida.
- ¿Vigente?: Este campo se utiliza para conocer si el asesor se encuentra vigente para poder asesorar a un estudiante o no.
- ¿Es catedrático?: Este campo se utiliza para indicar si el asesor es actualmente catedrático en la Escuela de Ciencias y Sistemas
- ¿Incluir *curriculum vitae*?: Este campo se utiliza para indicar si se desea adjuntar un archivo que contenga el *curriculum vitae* del asesor, si la opción seleccionada es afirmativa se mostrará un cuadro como el que se muestra en la figura 58. Para seleccionar un archivo de un clic en el botón “Examinar...”, se mostrará un cuadro de diálogo en el cual podrá seleccionar el archivo a adjuntar.

Figura 58. ¿Incluir *curriculum vitae*?



¿Incluir Curriculum Vitae? Si No

Nombre del archivo: Examinar...

Crear asesor

Después de introducir toda la información necesaria de un clic en el botón “Crear asesor”, si toda la información es válida y no se presenta ningún problema en la creación del asesor se mostrará un cuadro de diálogo indicando el éxito de la operación y el asesor habrá sido creado.

4.1.4.2. Crear estudiantes

Aquí puede crear los estudiantes que forman parte del sistema y están trabajando o han terminado su trabajo de graduación, los datos requeridos se describen a continuación:

- Carné: Este campo es obligatorio, debe escribir el carné del estudiante utilizando únicamente números.
- Nombre: Este campo es obligatorio, debe escribir el nombre del estudiante.
- Apellido: Este campo es obligatorio, debe escribir el apellido del estudiante.
- Teléfono: Este campo es opcional, debe escribir el número de teléfono del estudiante. Se pueden utilizar únicamente números.

- Celular: Este campo es opcional, debe escribir el número de celular o teléfono móvil del estudiante. Se pueden utilizar únicamente números.
- Correo electrónico: Este campo es opcional, debe escribir una dirección de correo electrónico válida.
- ¿Asignar Tutor?: Si selecciona esta opción podrá asignar un tutor que guiará al estudiante, se habilitan las siguientes opciones:
 - Tutor asignado: Debe seleccionar de una lista de tutores el que será asignado al estudiante que está siendo creado.
 - Año: Indica el año en el que será asignado el estudiante al tutor.
 - Semestre: Indica el semestre en el que será asignado el estudiante al tutor.

Después de introducir toda la información necesaria de un clic en el botón “Crear estudiante”, si toda la información es válida y no se presenta ningún problema en la creación del estudiante se mostrará un cuadro de diálogo indicando el éxito de la operación y el estudiante habrá sido creado.

4.1.4.3. Cargar estudiantes

Para crear estudiantes de una forma más fácil y rápida se puede utilizar esta opción, aquí puede crear un archivo con los datos de los estudiantes y cargarlos al sistema en un solo paso. El formato del archivo se muestra en la figura 59 y se explica a continuación.

Figura 59. **Formato archivo cargar estudiantes**

	A	B	C	D	E	F	G	H	I	J
1	Carné	Nombre	Apellido	Teléfono	Celular	Correo Electrónico	Tutor Asignado	Año	Semestre	(Relleno)
2	200010805	Víctor	Zamora	66350123	54073827	vzamora@tesauro.com				#
3	200010806	Luis	López				administrador	2006	1	#
4	200010807	Brenda	Gutiérrez							#

El formato se compone por 10 campos, el Carné, Nombre y Apellido son obligatorios y Teléfono, Celular y Correo Electrónico son opcionales.

Si desea realizar una asignación a un tutor que forma parte del sistema debe llenar los campos: Tutor Asignado, Año y Semestre, siendo los tres campos obligatorios; si no desea realizar la asignación debe dejar los tres campos en blanco.

El campo especial (Relleno) se utiliza para indicar el final de un registro de estudiante, se puede utilizar cualquier carácter pero se recomienda el uso del símbolo numeral (#).

El archivo con el formato descrito anteriormente puede ser descargado en la página de carga de estudiantes, el formato utilizado es CSV (*Coma Separated Values*) que significa un archivo delimitado por comas, este puede ser editado en *Microsoft Excel*.

Al tener listo el archivo con los estudiantes que desea crear, debe dar un clic el botón “Examinar...”, buscar el archivo con los estudiantes y luego dar un clic en el botón “Cargar”. El resultado de la operación para cada estudiante es mostrado en una tabla de resultados como la que se muestra en la figura 60.

Figura 60. **Tabla de resultados de cargar estudiantes**

Carné	Operación	Resultado
200010805	Crear Estudiante	El Estudiante ya existe en el Sistema.
200010806	Crear Estudiante	El Estudiante fue creado con éxito.
200010806	Asignar Tutor	El Estudiante fue asignado con éxito al Tutor: administrador, en el semestre: 1-2006.
200010807	Crear Estudiante	El Estudiante fue creado con éxito.

En la tabla de resultados se muestran en color *cyan* o celeste las operaciones que retornaron un resultado positivo, en color naranja las operaciones que presentaron algún problema, aunque no grave, por ejemplo en la figura 60 se puede observar que el estudiante con carné 200010805 ya existe en el sistema por cual el resultado se muestra en color naranja. Los resultados que presenta un problema grave se muestran en color rojo.

4.1.4.4. Crear documentos

Aquí puede crear los estudiantes que forman parte del sistema y están trabajando o han terminado su trabajo de graduación, los datos requeridos se describen a continuación:

- Carné: Aquí se selecciona el carné del estudiante al que pertenece el documento que está siendo creado.
- Título: Este campo es obligatorio, debe escribir el título del documento que está siendo creado.
- Contenido: Este campo es opcional, se utiliza para incluir el contenido o índice del documento. Para poder incluir texto con formato desde *Microsoft Word* utilice el icono  de la barra de herramientas.

- Resúmen: Este campo es opcional, se utiliza para incluir un resúmen del documento. Para poder incluir texto con formato desde *Microsoft Word* utilice el icono  de la barra de herramientas.
- Estado: Aquí se selecciona el estado actual del documento, existen 8 predefinidos en la aplicación, se describen a continuación:
 - Pendiente aprobación: Indica que el documento no ha sido revisado y por lo tanto se encuentra en un estado pendiente de aprobación.
 - Rechazado: Indica que el documento ya ha sido revisado y su contenido ha sido rechazado.
 - Aprobado: Indica que el documento ya ha sido revisado y su contenido ha sido aprobado.
 - Revisión asesor: Indica que el documento está en el proceso de elaboración y el asesor se encuentra trabajando con el estudiante.
 - Revisión revisor: Indica que el documento está siendo revisado por el revisor de la Escuela de Ciencias y Sistemas.
 - Revisión Lingüística: Indica que el documento se encuentra en el proceso de revisión de ortografía, gramática, estilo y formato en el departamento de lingüística de la Facultad de Ingeniería.
 - Revisión Director: Indica que el documento está siendo revisado por el Director de la Escuela de Ciencias y Sistemas.

- Finalizado: Indica que el documento es un trabajo de graduación que ha sido terminado y ha cumplido con todos los requisitos previos.
- Inicio vigencia: En algunos estados se puede dar un tiempo de validez para la realización del trabajo de graduación, este campo se utiliza para establecer el inicio de la validez de un documento en un estado específico.
- Fin vigencia: Aquí se establece el fin de la validez de un documento en un estado específico. Este campo se utiliza para indicar la fecha de finalización de un trabajo de graduación, cuando su estado cambia a finalizado.
- ¿Incluir documento?: Este campo se utiliza para indicar si se desea adjuntar un archivo que contenga el contenido completo del documento, si la opción seleccionada es afirmativa se mostrará un cuadro como el que se muestra en la figura 61. Para seleccionar un archivo de un clic en el botón “Examinar...”, se mostrará un cuadro de diálogo en el cual podrá seleccionar el archivo a adjuntar.

Figura 61. ¿Incluir documento?



¿Incluir documento? Si No

Archivo:

- Comentario: Este campo se utiliza para poder agregar comentarios del estado del documento o bien cualquiera que el usuario quiera incluir.

Después de introducir toda la información necesaria de un clic en el botón “Crear documento”, si toda la información es válida y no se presenta ningún problema en la creación del documento se mostrará un cuadro de diálogo indicando el éxito de la operación y el documento habrá sido creado.

Adicionalmente, en el momento de creación o modificación de un documento se crea una entrada en un historial, este almacena el estado, el inicio de vigencia, fin de vigencia y el comentario; este historial puede ser consultado en el listado de documentos, que será explicado más adelante.

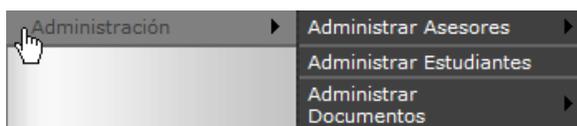
4.1.5. Rol tutor

El rol tutor puede realizar todas las acciones del rol digitador (exceptuando la opción “Crear estudiante” y “Cargar estudiantes”) y puede además realizar las siguientes acciones:

- Administrar asesores.
- Administrar estudiantes.
- Administrar documentos.

Para ver estas acciones coloque el puntero del *mouse* sobre el elemento “Administración” del menú que se encuentra al lado izquierdo de la aplicación como se muestra en la figura 62.

Figura 62. Opciones rol tutor



A continuación se explica cada una de las opciones del menú de administración para el rol tutor:

4.1.5.1. Administrar asesores

La pantalla que aparece al seleccionar esta opción se puede observar en la figura 63, aparece el listado de los asesores que han sido creados en el sistema.

Figura 63. Administrar asesores

Encabezado

Nombre	Apellido	Teléfono	Celular	Correo Electrónico	¿Vigente?	¿Es Catedrático?	Curriculum Vitae		
Elizabeth	Dominguez			edominguez@tesauro.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Ver ...		
Rodolfo	Zamora	12345678		rzamora@tesauro.com	<input type="checkbox"/>	<input type="checkbox"/>	Ver ...		
David	Esquivel		87654321		<input checked="" type="checkbox"/>	<input type="checkbox"/>	Ver ...		

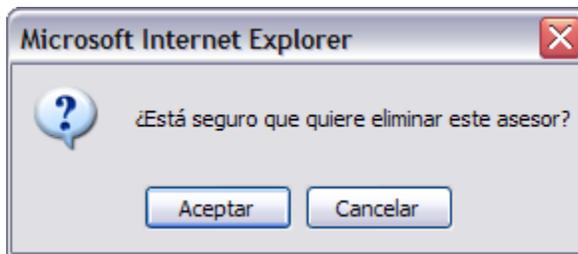
Editar Eliminar

Las opciones de administrar asesores son descritas a continuación:

- Crear asesor: Al dar clic este botón nos envía a la opción de crear asesor. Para más información ver el apartado: 4.1.4.1 Crear asesores.
- Encabezado: Aparece el título de cada columna, aquí se tiene la opción se ordenar la información de forma alfabética al dar clic en el título donde desea hacer el ordenamiento.

- **Imagen editar:** Al dar clic en la imagen tendremos la opción de modificar los datos del asesor, esta opción se utiliza de la misma manera que la opción Crear asesor, la diferencia es que aquí se modifican los datos de un asesor seleccionado.
- **Botón de eliminar:** Se utiliza para eliminar un asesor del sistema. Al dar clic aparecerá una confirmación como la que se puede ver en la figura 64. Si se selecciona la opción “Aceptar” el asesor será eliminado, si se selecciona la opción “Cancelar” no sucederá nada.

Figura 64. **Confirmación para eliminar asesor**

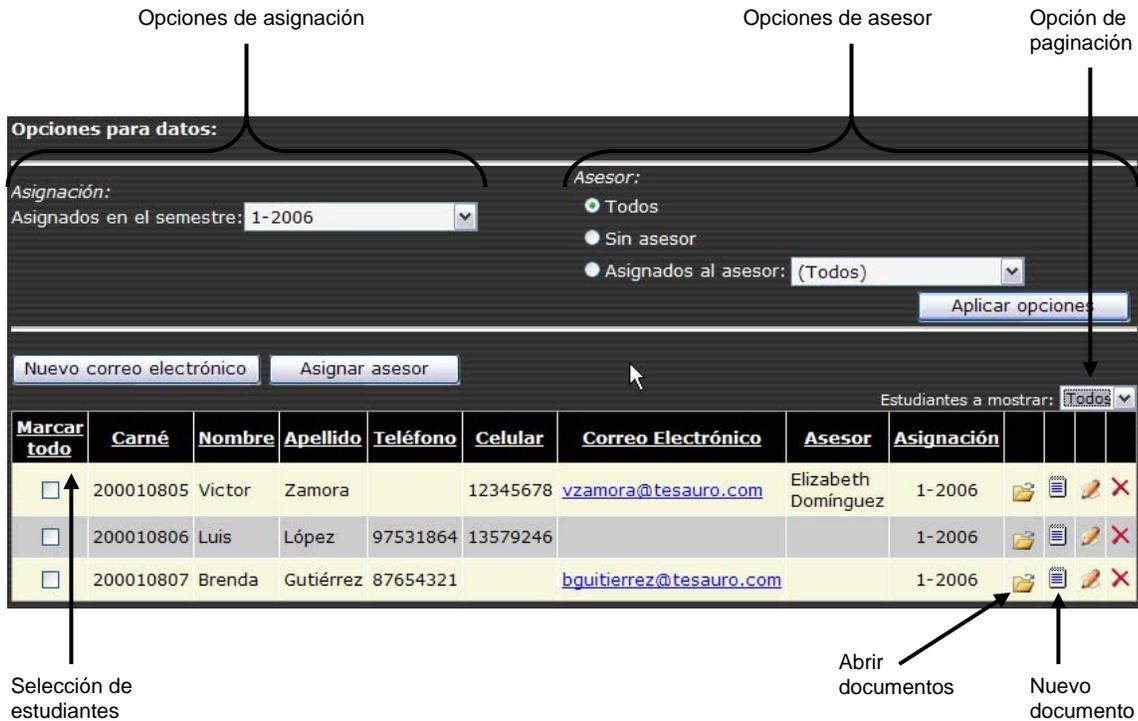


4.1.5.2. Administrar estudiantes

Al seleccionar esta opción aparece una pantalla como la que se puede ver en la figura 65, es el listado de estudiantes que han sido asignados al usuario tutor que ha iniciado sesión en la aplicación.

De forma predeterminada únicamente se muestra el listado de los estudiantes asignados en el semestre actual, este filtro de datos se puede cambiar, la forma de hacerlo se explica más adelante.

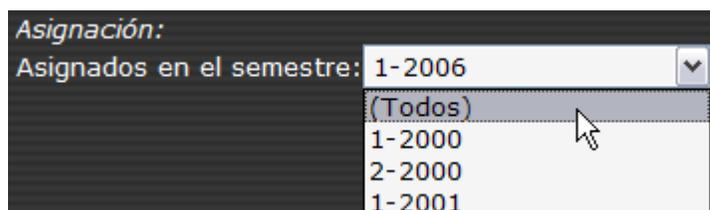
Figura 65. **Administrar estudiantes**



Las opciones de administrar estudiantes son descritas a continuación:

- Opciones de asignación: Como se mencionó anteriormente, esta pantalla muestra de forma predeterminada únicamente los estudiantes que han sido asignados en el semestre actual, pero se puede cambiar seleccionando del listado el semestre deseado o bien seleccionado la opción “(Todos)” como se muestra en la figura 66. Luego de seleccionar su opción debe dar un clic en el botón “Aplicar opciones”.

Figura 66. **Opciones de asignación**



- Opciones de asesor: Funciona de la misma forma que la opciones de asignación, de forma predeterminada se muestran todos los estudiantes, los que tienen asignado un asesor y los que no. Hay tres opciones que se pueden utilizar:
 - Todos: Muestra todos los estudiantes, los que tienen asignado un asesor y los que no.
 - Sin asesor: Se muestran únicamente los estudiantes que no tienen un asesor asignado.
 - Asignados al asesor: Esta opción permite seleccionar si se van a mostrar únicamente los estudiantes que tengan asignado un asesor específico o bien se puede seleccionar que se muestren todos los estudiantes que tengan asignado un asesor.

Luego de seleccionar su opción debe dar un clic en el botón “Aplicar opciones”.

- Selección de estudiantes: Esta columna se utiliza para seleccionar uno o más estudiantes a los cuales se les puede enviar un correo electrónico o asignar un asesor, estas opciones se explican más adelante. También puede hacer un clic en el título de la columna para marcar todos los estudiantes o para desmarcar todos los estudiantes.
- Enviar correo electrónico: Para enviar un correo electrónico debe seleccionar al menos un estudiante; al dar clic en este botón se mostrará la pantalla que se puede ver en la figura 67.

Figura 67. Enviar correo electrónico

The image shows a screenshot of an email composition interface. At the top, there are several input fields: "Para:" containing the email addresses "vzamora@tesauro.com,bguitierrez@tesauro.com", "CC:", "BCC:", and "Asunto:". Below these is an "Adjuntar:" section with a large empty box and three buttons: "Eliminar", "Examinar...", and "Agregar". Underneath is a "Mensaje:" section with a rich text editor toolbar containing icons for undo, redo, bold, italic, underline, text color, background color, bulleted list, numbered list, link, unlink, indent, outdent, and text wrap. Below the toolbar are dropdown menus for "Estilo", "Formato", "Fuente", and "Tamaño". The main body of the email is a large empty text area. At the bottom right, there are "Enviar" and "Cancelar" buttons.

En el campo “Para” se incluyen los correos electrónicos de todos los estudiantes que han sido seleccionados y tienen direcciones de correo electrónico.

Además puede agregar sus propias direcciones en los campos “CC” y “BCC”, los dos campos envían copias del correo con la diferencia que el campo “BCC” envía copias ocultas, es decir, los otros destinatarios no podrán saber que una copia ha sido enviada a las direcciones que se escribieron en ese campo.

También puede adjuntar archivos en el correo electrónico, para esto utilice el botón “Examinar...” para seleccionar el archivo y luego presione el botón “Agregar” para adjuntar el archivo.

Cuando haya escrito su mensaje de un clic en el botón “Enviar” para enviar el correo electrónico o de un clic en “Cancelar” para regresar a la pantalla de administración de estudiantes sin enviar el correo.

- Asignar asesor: Para asignar un asesor debe seleccionar al menos un estudiante; al dar clic en este botón se mostrará la pantalla que se puede ver en la figura 68.

Figura 68. Asignar asesor

Carné	Nombre	Apellido
200010807	Brenda	Gutiérrez

Reemplazar asignación anterior:

Seleccione asesor: Elizabeth Domínguez

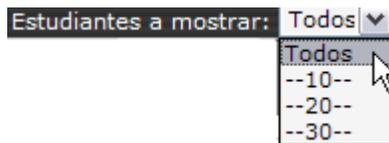
Asignar Cancelar

Si selecciona la opción “Reemplazar asignación anterior” será asignado el asesor que seleccione de la lista y reemplazará al asesor anterior si existía una asignación.

Luego de seleccionar el asesor de un clic en el botón “Asignar” para realizar la asignación o de un clic en el botón “Cancelar” para regresar a la pantalla de administración de estudiantes sin realizar la asignación.

- Opción de paginación: De forma predeterminada se muestran todos los estudiantes en la misma pantalla, si la cantidad es muy grande o quiere que los estudiantes se muestre por páginas puede utilizar esta opción y seleccionar ver 10, 20, 30 o todos los estudiantes, ver figura 69.

Figura 69. Opción de paginación

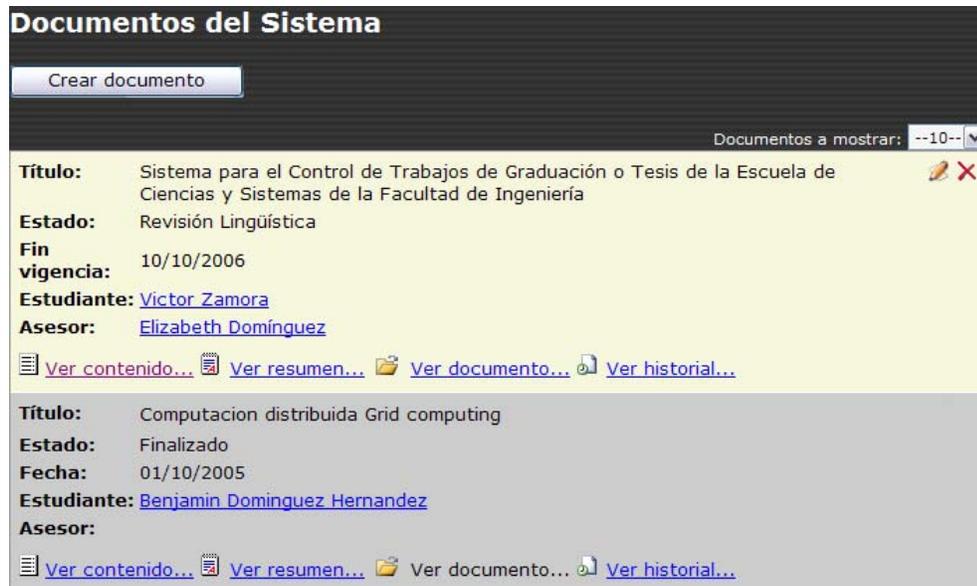


- Abrir documentos: Esta opción funciona de la misma manera que la opción administrar documentos que es explicada más adelante, la única diferencia es que aquí únicamente se muestran los documentos del estudiante seleccionado.
- Nuevo documento: Al dar clic este botón nos envía a la opción de crear documento. Para más información ver el apartado: 4.1.4.4 Crear documentos. La única diferencia es que el documento que será creado puede ser asignado únicamente al estudiante seleccionado.

4.1.5.3. Administrar documentos

Al seleccionar esta opción aparece una pantalla como la que se puede ver en la figura 70. Aquí se muestran todos los documentos que existen el sistema, de forma predeterminada se muestran 10 documentos por página, esta opción se puede cambiar seleccionando 10, 20, 30 o todos en la opción “Documentos a mostrar”.

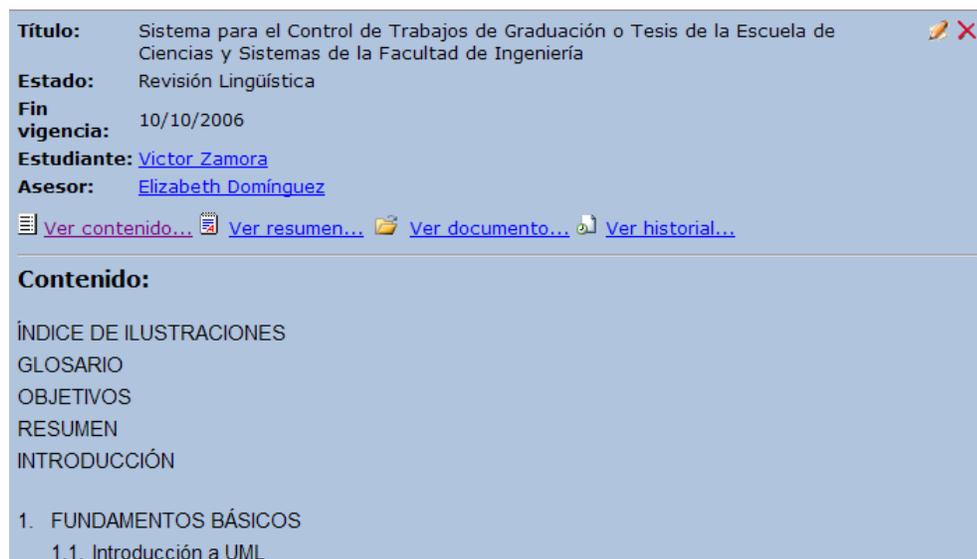
Figura 70. Administrar documentos



Las opciones de administrar documentos son descritas a continuación:

- Ver contenido: Al dar un clic en este vínculo se muestra el contenido del documento como se puede ver en la figura 71.

Figura 71. Ver contenido de un documento



- Ver resumen: Al dar un clic en este vínculo se muestra el resumen del documento.
- Ver documento: Si en el momento de creación o edición de un documento se agregó un archivo se podrá abrir al dar clic en este vínculo.
- Ver historial: Al dar clic en este vínculo se muestra el historial del documento, este se crea cada vez que se modifica un documento. Puede ver un ejemplo de historial en la figura 72.

Figura 72. **Historial de un documento**

Historial:					
Estado	Inicio Vigencia	Fin vigencia	Comentario	Asesor	Estudiante
Revisión Asesor	23/04/2006	24/06/2006	Trabajando con el asesor	Elizabeth Domínguez	200010805
Aprobado	01/04/2006	01/04/2006	Protocolo aprobado	Elizabeth Domínguez	200010805
Pendiente Aprobación	25/03/2006	25/03/2006	Entrega de protocolo	Elizabeth Domínguez	200010805

- Crear documento: Al dar clic este botón nos envía a la opción de crear documento. Para más información ver el apartado: 4.1.4.4 Crear documentos.

Las opciones de edición () y eliminación () únicamente están disponibles para los documentos de los estudiantes que han sido asignados en algún semestre al usuario tutor.

4.1.6. Rol administrador

El rol administrador puede realizar todas las acciones del rol digitador y el rol tutor, además realizar las siguientes acciones:

- Crear usuarios.
- Administrar usuarios.
- Administrar roles.
- Administrar estados.

En las opciones “Administrar usuarios” y “Administrar documentos” se agregan permisos especiales para el rol administrador, estos son explicados más adelante.

A continuación se explica cada una de las opciones disponibles para el rol administrador:

4.1.6.1. Crear usuarios

Aquí puede crear los usuarios que forman parte del sistema y trabajan con la aplicación, los datos requeridos se describen a continuación (todos los campos son obligatorios):

- Nombre de usuario: Debe escribir el nombre que se utilizará para ingresar en el sistema, este nombre es único, es decir, no pueden haber dos usuarios con el mismo nombre.

- **Contraseña:** Debe escribir la contraseña que se utilizará para ingresar al sistema, esta debe tener una longitud mínima de 5 caracteres y tener al menos un carácter no alfanumérico, es decir un símbolo, por ejemplo: !, @, #, \$, %, etc.
- **Confirmar contraseña:** Este campo se utiliza para confirmar la contraseña ingresada en el campo anterior, se debe escribir la misma contraseña.
- **Nombre:** Se utiliza para escribir el nombre propio del usuario al que será asignada la cuenta.
- **Correo electrónico:** Es la dirección de correo electrónico que se utiliza para recordar la contraseña, cambiar la contraseña y cambiar la pregunta y respuesta de seguridad. Este campo también es único en el sistema, es decir, dos usuarios no pueden compartir la misma dirección.
- **Pregunta de seguridad:** Al momento de recuperar su contraseña esta pregunta aparecerá para verificar su identidad.
- **Respuesta de seguridad:** Es la respuesta a la pregunta que ingresó en el campo anterior, nunca debe olvidar esta respuesta, ya que es imposible recuperarla del sistema.
- **Rol:** Debe seleccionar el rol al que el usuario pertenecerá, este define los privilegios que tiene el sistema.

Si la información ingresada es correcta se muestra una confirmación y el usuario es creado. La información ingresada será enviada a la dirección de correo electrónico que utilizó en el formulario de creación.

4.1.6.2. Administrar usuarios

Al seleccionar esta opción aparece una pantalla como la que se puede ver en la figura 73. Aquí se muestran todos los usuarios que existen el sistema.

Figura 73. Administrar usuarios

Usuarios del Sistema						
<input type="button" value="Crear usuario"/>						
Usuario	Nombre	Correo Electrónico	Rol	Activo		
Administrador	Victor Zamora	vzamora@tesauro.com	Administrador	<input checked="" type="checkbox"/>		
ejdigitador	Usuario Digitador	digitador@ejemplo.com	Digitador	<input checked="" type="checkbox"/>		
ejtutor	Usuario Tutor	tutor@ejemplo.com	Tutor	<input checked="" type="checkbox"/>		
fluna	Fred Luna	fluna@tesauro.com	Especial	<input checked="" type="checkbox"/>		
rmunoz	Rogelio Muñoz	rmunoz@tesauro.com	Tutor	<input checked="" type="checkbox"/>		
smorales	Steve Morales	smorales@tesauro.com	Tutor	<input checked="" type="checkbox"/>		

Las opciones de administrar usuarios son descritas a continuación:

- **Crear usuario:** Al dar clic este botón nos envía a la opción de crear usuarios. Para más información ver el apartado: 4.1.6.1 Crear usuarios.
- **Activo:** Esta columna indica el estado de la cuenta de usuario, activo quiere decir que la cuenta puede ser utilizada para ingresar al sistema, si la cuenta esta desactivada el usuario que utilice esta cuenta no podrá ingresar al sistema aunque el nombre de usuario y contraseña sean correctos al intentar iniciar sesión. Esta opción puede ser modificada al dar clic en el botón editar usuario ().

El usuario “Administrador” no puede ser desactivado, no se puede cambiar su rol y no puede ser eliminado.

4.1.6.3. Administrar roles

Al seleccionar esta opción aparece una pantalla como la que se puede ver en la figura 74. Aquí se muestran todos los roles que existen el sistema.

Figura 74. Administrar roles

The screenshot displays a web interface for managing roles. At the top, there is a section titled "Roles del Sistema" containing a table with the following data:

Nombre del Rol	Descripción		
Administrador	Administrador del sitio Web.		
Digitador	Únicamente puede ingresar datos.		
Especial	Tiene permisos especiales en la aplicación.		
Tutor	Puede modificar estudiantes y asesores.		

Below the table is a section titled "Crear Rol" with two input fields: "Nombre del Rol:" and "Descripción:". A "Crear" button is located at the bottom left of this section.

Las opciones de administrar roles son descritas a continuación:

- Crear rol: Para crear un nuevo rol debe ingresar el "Nombre del Rol" en el cuadro correspondiente, este campo es obligatorio y debe ser un nombre único; el campo "Descripción" es opcional y se utiliza para dar una descripción del rol. Cuando haya ingresado los datos necesarios de un clic en el botón "Crear" y se mostrará un cuadro de diálogo confirmando la creación del rol y será agregado en la lista de roles.
- Para modificar y eliminar roles utilice los botones edición () y eliminación () correspondientemente. Los roles Administrador, Tutor y Digitador no pueden ser eliminados.

Para configurar los permisos que tendrán los roles creados será necesario modificar 2 archivos que se encuentran en el directorio virtual del sitio *Web*. El nombre de los archivos a configurar es *web.config*, uno se encuentra en el directorio raíz (*/tesauro/*) y el otro en el directorio *admin* (*/tesauro/admin/*).

Estos archivos contienen varias secciones de configuración, esta sección únicamente se explicará la secciones que se utilizan para configurar los permisos para los roles, si desea información de otras secciones consulte el manual técnico en el apartado: 4.2.3.5.1 Archivo *web.config*.

Las secciones que se utilizan para configurar el acceso se llaman *location* y tienen el formato que se muestra a continuación:

```
<location path="editAdviser.aspx">
  <system.web>
    <authorization>
      <allow roles="administrador, tutor" />
      <deny users="*" />
    </authorization>
  </system.web>
</location>
```

El único parámetro que necesita modificar es “*allow roles*”, el ejemplo anterior permite el acceso a los roles administrador y tutor a la página *editAdviser.aspx*.

Por ejemplo, si usted crea un rol que se llama Especial y desea permitir el acceso a la página *editAdviser.aspx* debe modificar la sección de la siguiente forma:

```

<location path="editAdviser.aspx">
  <system.web>
    <authorization>
      <allow roles="administrador, tutor, especial" />
      <deny users="*" />
    </authorization>
  </system.web>
</location>

```

De igual forma si usted quiere que un rol no tenga acceso a esta página únicamente debe eliminar el nombre del rol del parámetro “allow roles”.

4.1.6.4. Administrar estados

Al seleccionar esta opción aparece una pantalla como la que se puede ver en la figura 75. Aquí se muestran todos los estados que existen el sistema.

Figura 75. Administrar estados

Estados de Documentos

Nombre del Estado	Descripción		
Aprobado	Aprobado por parte del tutor.		
Finalizado	Tesis o trabajo de graduación terminado.		
Pendiente Aprobación	Pendiente de aprobación por parte del tutor.		
Rechazado	Rechazado por parte del tutor.		
Revisión Asesor	Trabajo con el asesor.		
Revisión Director	Revisión del Director de Escuela.		
Revisión Lingüística	Revisión del especialista en lingüística.		
Revisión Revisor	Revisión del coordinador de área o revisor.		

Crear Estado

Nombre del Estado:

Descripción:

Las opciones de administrar roles son descritas a continuación:

- Crear estado: Para crear un nuevo estado debe ingresar el “Nombre del Estado” en el cuadro correspondiente, este campo es obligatorio y debe ser un nombre único; el campo “Descripción” es opcional y se utiliza para dar una descripción del estado. Cuando haya ingresado los datos necesarios de un clic en el botón “Crear” y se mostrará un cuadro de diálogo confirmando la creación del estado y será agregado en la lista de estados.
- Para modificar y eliminar estados utilice los botones edición () y eliminación () correspondientemente.

4.1.6.5. Administrar estudiantes (rol administrador)

Esta opción se utiliza de la misma forma que la opción administrar estudiantes del rol tutor, la diferencia es que el rol administrador tiene acceso a dos opciones más y las opciones de asignación tienen más características, las nuevas opciones se pueden ver en la figura 76.

Figura 76. Administrar estudiantes (rol administrador)

Opciones para datos:

Asignación:

Todos

Sin asignación

Asignados en el semestre:

con el tutor:

Asesor:

Todos

Sin asesor

Asignados al asesor:

Estudiantes a mostrar:

Marcar todo	Carné	Nombre	Apellido	Teléfono	Celular	Correo Electrónico	Asesor	Asignación	Tutor			
<input type="checkbox"/>	36	Orlando Sánchez Mendoza	José Christian Bradna Villanueva							  		

Las opciones de administrar estudiantes del rol administrador son descritas a continuación:

- Opciones de asignación: De igual forma que las opciones para el rol tutor está pantalla muestra de forma predeterminada únicamente los estudiantes que han sido asignados en el semestre actual; el rol administrador tiene tres opciones que puede utilizar para mostrar los datos que son:
 - Todos: Muestra todos los estudiantes, los que ha sido asignados a un usuario en algún semestre.
 - Sin asignación: Se muestran únicamente los estudiantes que no nunca han sido asignados a ningún usuario.
 - Asignados en el semestre: Esta opción permite mostrar los estudiantes que han sido asignados a un usuario específico seleccionándolo de la lista, o bien seleccionando la opción “(Todos)” para mostrar todos los estudiantes que tiene un usuario asignado en algún semestre.

Luego de seleccionar su opción debe dar un clic en el botón “Aplicar opciones”.

- Crear estudiante: Al dar clic este botón nos envía a la opción de crear estudiante. Para más información ver el apartado: 4.1.4.2 Crear estudiantes.

- Asignar tutor: Para asignar un tutor debe seleccionar al menos un estudiante; al dar clic en este botón se mostrará la pantalla que se puede ver en la figura 77.

Figura 77. Asignar tutor

Carné	Nombre	Apellido
200010805	Victor	Zamora

Reemplazar asignación anterior:

Seleccione tutor: Administrador

Año: 2006

Semestre: Primero

Asignar Cancelar

Si selecciona la opción “Reemplazar asignación anterior” será asignado el tutor que seleccione de la listas y reemplazará la asignación anterior en el año y semestre seleccionados.

Luego de seleccionar el tutor, año y semestre de un clic en el botón “Asignar” para realizar la asignación o de un clic en el botón “Cancelar” para regresar a la pantalla de administración de estudiantes sin realizar la asignación.

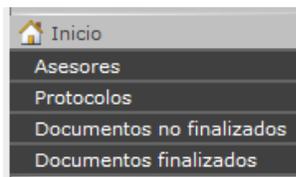
4.1.6.6. Administrar documentos (rol administrador)

La única diferencia que se hay en esta opción es que el rol administrador puede editar y eliminar todos los documentos del sistema, el rol tutor puede editar y eliminar únicamente los documentos de los estudiantes que han sido asignados a él en el algún semestre.

4.1.7. Consultas

Para realizar consultas no es necesario haber iniciado sesión en el sistema. La página de inicio de la aplicación muestra los últimos cinco trabajos de graduación terminados y en el menú de la izquierda (ver figura 78) se muestran cuatro opciones en las que se puede consultar información acerca de los asesores, protocolos, documentos no finalizados y documentos finalizados. A continuación se explica cada uno de los elementos del menú.

Figura 78. Menú de consultas



4.1.7.1. Menú asesores

Al seleccionar este menú podrá consultar todos los asesores que han trabajado o están trabajando con estudiante en el desarrollo de su trabajo de graduación (ver figura 79).

Figura 79. Asesores



The screenshot shows a web application interface. At the top, there is a blue header with the text 'Inicio : Asesores'. Below the header, the title 'Asesores' is displayed in white on a dark background. The main content is a table with the following columns: 'Nombre', 'Apellido', 'Teléfono', 'Celular', 'Correo Electrónico', '¿Vigente?', '¿Es Catedrático?', and 'Curriculum Vitae'. The table contains three rows of data.

	Nombre	Apellido	Teléfono	Celular	Correo Electrónico	¿Vigente?	¿Es Catedrático?	Curriculum Vitae
	Elizabeth	Domínguez			edominguez@tesauro.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Ver ...
	Rodolfo	Zamora	12345678		rzamora@tesauro.com	<input type="checkbox"/>	<input type="checkbox"/>	Ver ...
	David	Esquivel		87654321		<input checked="" type="checkbox"/>	<input type="checkbox"/>	Ver ...

Puede ordenar la lista dando un clic en el encabezado de cada columna y también puede ver los trabajos finalizados y no finalizados del asesor dando un clic en el icono , se mostrará una pantalla como la que se puede ver en la figura 80, en esta pantalla también se muestran los datos personales del asesor.

Figura 80. Datos del asesor

Datos del asesor

Nombre: Elizabeth
Apellido: Domínguez
Teléfono:
Celular:
Correo electrónico: elidominguez@tasawar.com
¿Vigente? Si No
¿Es catedrático? Si No
Curriculum Vitae:

Documentos no finalizados:

Título: Sistema para el Control de Trabajos de Graduación o Tesis de la Escuela de Ciencias y Sistemas de la Facultad de Ingeniería
Estado: Revisión Lingüística
Inicio vigencia: 10/08/2006
Fin vigencia: 10/10/2006
Estudiante: [Víctor Zamora](#)
[Ver contenido...](#) [Ver resumen...](#) [Ver documento...](#)

Documentos finalizados:

Título: Microsoft.Net y sun java enterprise system como herramientas para lograr aplicaciones de servicios por demanda, de hoy y manana
Fecha: 04/2005
Estudiante: [Karla Fabiola Santos Juarez](#)
[Ver contenido...](#) [Ver resumen...](#) [Ver documento...](#)

4.1.7.2. Menú protocolos

Aquí se muestran todos los protocolos que forman parte de la aplicación, se considera protocolo el documento que se encuentra en el estado *Pendiente aprobación*, *Rechazado* o *Aprobado*; sobre los protocolos se pueden realizar búsquedas ingresando la palabra a buscar en el cuadro de texto que se muestra en la figura 81 y luego dando clic en el botón *Búsqueda*.

Figura 81. **Protocolos**



El botón *Ver Todos* se utiliza para borrar cualquier búsqueda que se haya realizado y mostrar todos los documentos. En *Temas* se pueden seleccionar las siguientes opciones:

- Todos: Muestra todos los documentos.
- Disponibles: Muestra únicamente los documentos en los que el periodo de vigencia ha terminado, por lo tanto el tema puede ser solicitado.
- No Disponibles: Muestra únicamente los documentos que aún se encuentran vigentes, es decir, el tema no puede ser solicitado.

Si da clic en el nombre del estudiante o en el nombre de asesor se mostrarán los datos personales de cada uno respectivamente.

4.1.7.3. Menú documentos no finalizados

Este menú muestra los documentos que no han sido finalizados, sobre estos documentos se pueden realizar las mismas operaciones que se realizan con los protocolos. Los documentos no finalizados son los que se encuentran en alguno de los siguientes estados: *Revisión Asesor*, *Revisión Revisor*, *Revisión Lingüística*, *Revisión Director*, o bien cualquier otro estado que haya sido agregado a la aplicación por el Administrador y sea diferente a los estados reservados para un protocolo.

4.1.7.4. Menú documentos finalizados

Aquí se muestran todos los documentos finalizados, sobre estos documentos también se pueden realizar las mismas operaciones que se realizan con los protocolos y los documentos no finalizados con la excepción de la opción *Temas*.

4.2. Manual técnico

A continuación se describen los aspectos necesarios para poder comprender el funcionamiento de la aplicación, como lo son la tecnología utilizada para la construcción, la implementación de la base de datos, y descripción de los métodos y componentes utilizados en la implementación de la aplicación.

4.2.1. Tecnología utilizada

- *Microsoft Visual Studio 2005 Professional Edition*: Esta versión posee un poderoso conjunto de herramientas de desarrollo necesarias para la creación de aplicaciones cliente orientadas al manejo de datos, aplicaciones en n capas utilizando servicios *Web*, y ricas aplicaciones *Web*. Sus principales características son:
 - Crear aplicaciones de línea de negocio usando *Visual Basic*, *C#*, *C++* y *J#*.
 - Construir aplicaciones para *Windows*, la *Web* y dispositivos móviles.
 - Desarrollar aplicaciones cliente/servidor.

El lenguaje utilizado para la construcción de la aplicación *Web* fue *Visual Basic 2005*; al contrario de su antecesor *Visual Basic 6.0* que es un lenguaje manejado por eventos, es decir, los eventos son los que causan que una porción de código se ejecute, este es un lenguaje de programación orientado a objetos.

La programación orientada a objetos proporciona gran flexibilidad y fácil mantenimiento en el código escrito, ya que la implementación de cada objeto es independiente y por lo tanto, si es necesario realizar cambios estos no afectan a otros objetos.

Aunque se utilizó la edición *Professional* de *Visual Studio*, se puede utilizar cualquier otra edición, incluyendo la edición gratis *Visual Web Developer 2005 Express Edition*, que puede ser descargada del sitio *Web de Microsoft* (<http://msdn.microsoft.com/vstudio/express>).

- *Microsoft SQL Server 2000 Developer Edition*: Es un manejador de base de datos relacional y análisis de datos. Sus principales características son:
 - Habilitado para *Web*, se puede utilizar HTTP para enviar consultas, realizar búsquedas de texto en documentos almacenados y ejecutar consultas con el lenguaje natural. También es totalmente compatible con el lenguaje XML.
 - Alta escalabilidad y confiabilidad, soporta hasta 32 procesadores y 64 GB de RAM para aprovechar los sistemas de multiprocesador y además puede distribuir las bases de datos y la carga de datos entre servidores.

Se instaló el *SQL Server 2000 Service Pack 4*, este resuelve varios problemas de seguridad encontrados en la versión original y agrega nuevas características que mejoran la funcionalidad.

Es recomendable instalar como mínimo el *SQL Server 2000 Service Pack 3a*, ambas versiones pueden ser descargadas del sitio *Web de Microsoft* (<http://www.microsoft.com/sql/downloads/2000/default.mspx>).

También se puede utilizar *SQL Server 2005*, que es la nueva versión lanzada por *Microsoft* recientemente, esta versión incluye una edición gratis, *SQL Server 2005 Express*, que puede ser descargada del sitio *Web* de *Microsoft* (<http://msdn.microsoft.com/vstudio/express>), esta edición también está incluida en algunas de las ediciones de *Visual Studio 2005*.

- *Microsoft Windows XP Professional*: Este sistema operativo posee las características necesarias para ser utilizado en conjunto con *Visual Studio 2005* y *SQL Server 2000* para el desarrollo de la aplicación *Web*, aunque pueden ser utilizados cualquiera de los siguientes: *Windows Server 2003*, *Windows 2000 Professional* o *Server*.

- *InstallShield 12*: Esta es una herramienta que se utiliza para crear instaladores para *Windows*, las principales características por la cual fue utilizada son:
 - Se integra completamente con *Visual Studio 2005*, esto permite crear un instalador para la aplicación *Web*, utilizando el mismo entorno de desarrollo.

 - El instalador puede ejecutar *scripts* de creación de bases de datos para *SQL Server*.

Una versión de demostración de esta herramienta puede ser descargada desde el sitio de *Macrovision* (<http://www.macrovision.com>).

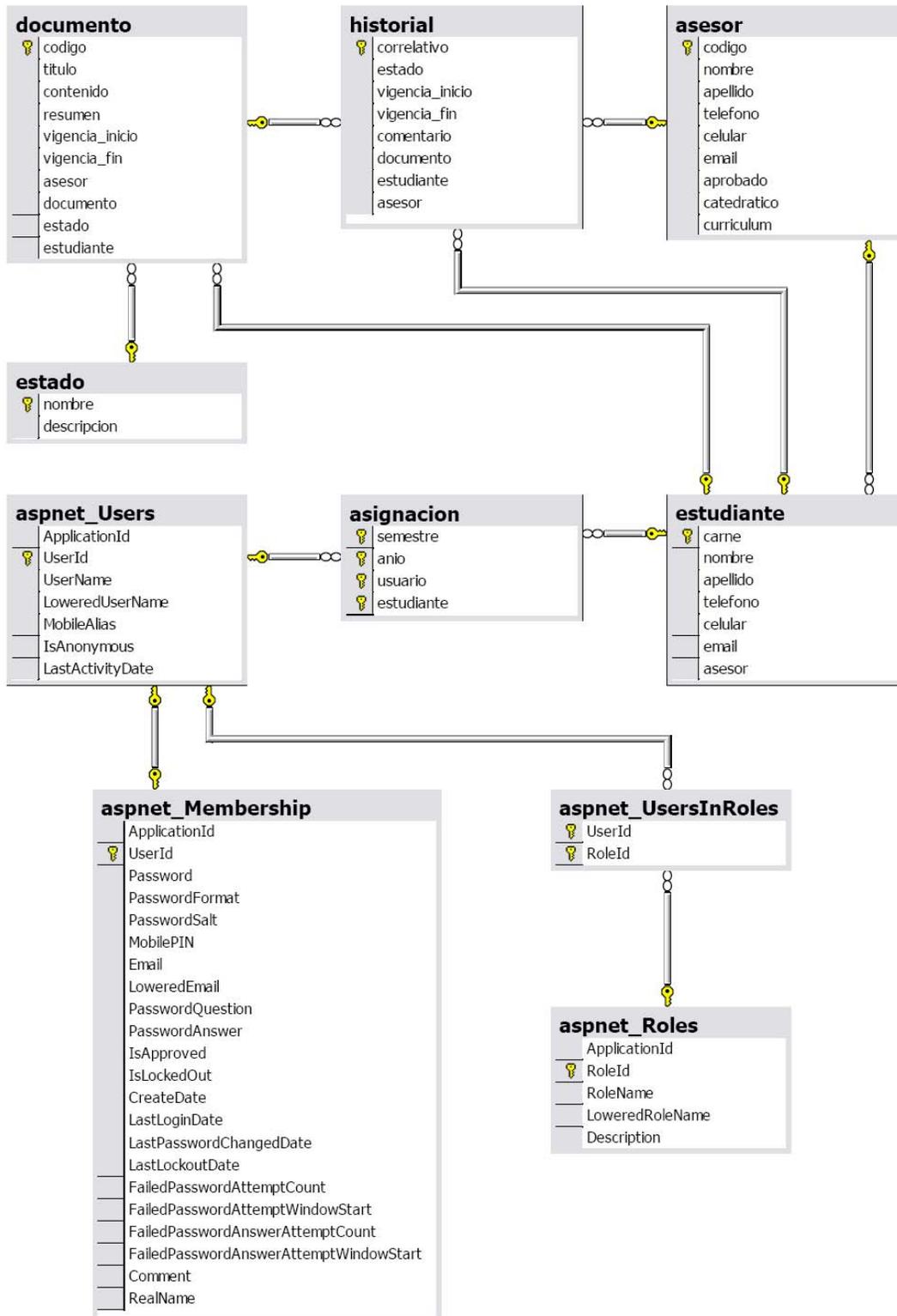
4.2.2. Implementación de la base de datos

En el apartado 3.1 Diseño de la base de datos, se describen la entidades y se presenta el diagrama entidad relación; en este apartado se detalla la implementación y los cambios que son necesarios para acoplarlo al manejador de base de datos *SQL Server 2000* y al lenguaje de programación *Visual Basic 2005*.

En la figura 82 se muestra el diagrama de implementación de la base de datos, como se puede observar las tablas *usuario* y *rol* del diagrama original son sustituidas por las tablas *aspnet_Users*, *aspnet_Membership*, *aspnet_Roles* y *aspnet_UsersInRoles*. Estas tablas forman parte del modelo de seguridad que es implementado en *ASP.NET 2.0*, este modelo agrega un juego de tablas a la base de datos que es utilizado para manejar usuarios, roles, personalización y eventos. Este documento sólo hace referencia a las tablas de usuarios y roles.

Para crear estas tablas es necesario ejecutar la aplicación *aspnet_regsql.exe*; es necesario haber instalado *Microsoft Visual Studio 2005* o bien el *Microsoft .NET Framework 2.0*, este se puede descargar desde el sitio Web de *Microsoft* (<http://msdn.microsoft.com/netframework/>). La aplicación se encuentra en la dirección *C:\WINDOWS\Microsoft.NET\Framework\v2.0.xxxx*. Esta aplicación permite seleccionar una base de datos en un servidor existente y luego crea todas las tablas necesarias para soportar el modelo de seguridad, las tablas creadas tienen el prefijo “*aspnet_*” en el nombre.

Figura 82. Diagrama de implementación de la base de datos



4.2.2.1. Tablas

A continuación se describen las tablas mostradas en la figura 78 y se muestra el *script* utilizado para su creación.

4.2.2.1.1. Tabla *aspnet_Users*

Esta tabla almacena la información relacionada con la identificación del usuario en la aplicación, en la tabla I se muestra el nombre de la columna, su tipo y descripción.

Tabla I. Tabla *aspnet_Users*

Nombre de la columna	Tipo de la columna	Descripción
<i>ApplicationId</i>	<i>uniqueidentifier</i>	Identificador único de la aplicación
<i>UserId</i>	<i>uniqueidentifier</i>	Identificador único del usuario
<i>UserName</i>	<i>nvarchar(256)</i>	Nombre del usuario
<i>LoweredUserName</i>	<i>nvarchar(256)</i>	Nombre del usuario en minúscula
<i>MobileAlias</i>	<i>nvarchar(16)</i>	Alias móvil del usuario (no usado actualmente)
<i>IsAnonymous</i>	<i>bit</i>	1 = Usuario anónimo, 0 = Usuario no anónimo
<i>LastActivityDate</i>	<i>datetime</i>	Fecha y hora de la última actividad del usuario

El *script* utilizado para la creación de esta tabla no se muestra, ya que forma parte de las tablas creadas con la aplicación *aspnet_regsql.exe*.

4.2.2.1.2. Tabla *aspnet_Membership*

Esta tabla almacena la información relacionada con la admisión del usuario en la aplicación, en la tabla II se muestra el nombre de la columna, su tipo y descripción.

Tabla II. Tabla *aspnet_Membership*

Nombre de la columna	Tipo de la columna	Descripción
<i>ApplicationId</i>	<i>uniqueidentifier</i>	Identificador único de la aplicación
<i>UserId</i>	<i>uniqueidentifier</i>	Identificador único del usuario
<i>Password</i>	<i>nvarchar(128)</i>	Contraseña en el formato especificado por <i>PasswordFormat</i>
<i>PasswordFormat</i>	<i>int</i>	Formato de la contraseña, 0 = texto plano, 1 = <i>Hashed</i> , 2 = cifrada
<i>PasswordSalt</i>	<i>nvarchar(128)</i>	Valor aleatorio de 128 bits usado con el formato de contraseña
<i>MobilePIN</i>	<i>nvarchar(16)</i>	<i>PIN</i> móvil del usuario (no usado actualmente)
<i>Email</i>	<i>nvarchar(256)</i>	Dirección de correo electrónico del usuario
<i>LoweredEmail</i>	<i>nvarchar(256)</i>	Dirección de correo electrónico del usuario en minúscula.
<i>PasswordQuestion</i>	<i>nvarchar(256)</i>	Pregunta para recuperar contraseña
<i>PasswordAnswer</i>	<i>nvarchar(128)</i>	Respuesta para <i>PasswordQuestion</i>
<i>IsApproved</i>	<i>bit</i>	1 = aprobado, 0 = no aprobado

Continúa

<i>IsLockedOut</i>	<i>bit</i>	1 = bloqueado, 0 = no bloqueado
<i>CreateDate</i>	<i>datetime</i>	Fecha y hora de la creación del usuario
<i>LastLoginDate</i>	<i>datetime</i>	Fecha y hora del último ingreso del usuario
<i>LastPasswordChangedDate</i>	<i>datetime</i>	Fecha y hora del último cambio de contraseña
<i>LastLockoutDate</i>	<i>datetime</i>	Fecha y hora del último bloqueo del usuario
<i>FailedPasswordAttemptCount</i>	<i>int</i>	Número de fallos consecutivos al intentar ingresar
<i>FailedPasswordAttempt- WindowsStart</i>	<i>datetime</i>	Fecha y hora del primer fallo al ingresar
<i>FailedPasswordAnswer- AttemptCount</i>	<i>int</i>	Número de fallos consecutivos al intentar recuperar la contraseña
<i>FailedPasswordAnswer- AttemptWindowsStart</i>	<i>datetime</i>	Fecha y hora del primer fallo al intentar recuperar la contraseña
<i>Comment</i>	<i>ntext</i>	Comentarios u observaciones adicionales
<i>RealName</i>	<i>nvarchar(100)</i>	Este campo no forma parte de la definición original de la tabla, fue agregado para poder almacenar el nombre real de cada usuario.

El *script* utilizado para la creación de esta tabla no se muestra, ya que forma parte de las tablas creadas con la aplicación *aspnet_regsql.exe*, el siguiente *script* es el que utilizó para agregar la columna *RealName* a la definición de la tabla:

```
ALTER TABLE aspnet_Membership
    ADD RealName NVARCHAR(100) NULL;
```

4.2.2.1.3. Tabla *aspnet_Roles*

Esta tabla almacena los roles que son manejados en la aplicación, el nombre de los roles es único y cada rol puede tener una descripción; en la tabla III se muestra el nombre de la columna, su tipo y descripción.

Tabla III. Tabla *aspnet_Roles*

Nombre de la columna	Tipo de la columna	Descripción
<i>ApplicationId</i>	<i>uniqueidentifier</i>	Identificador único de la aplicación
<i>RoleId</i>	<i>uniqueidentifier</i>	Identificador único del rol
<i>RoleName</i>	<i>nvarchar(256)</i>	Nombre del rol
<i>LoweredRoleName</i>	<i>nvarchar(256)</i>	Nombre del rol en minúscula
<i>Description</i>	<i>nvarchar(256)</i>	Descripción del rol

El *script* utilizado para la creación de esta tabla no se muestra, ya que forma parte de las tablas creadas con la aplicación *aspnet_regsql.exe*.

4.2.2.1.4. Tabla *aspnet_UsersInRoles*

Esta tabla se utiliza para hacer la relación de roles y usuarios, en la tabla IV se muestra el nombre de la columna, su tipo y descripción.

Tabla IV. **Tabla aspnet_UsersInRoles**

Nombre de la columna	Tipo de la columna	Descripción
<i>UserId</i>	<i>uniqueidentifier</i>	Identificador único del usuario
<i>RoleId</i>	<i>uniqueidentifier</i>	Identificador único del rol

El *script* utilizado para la creación de esta tabla no se muestra, ya que forma parte de las tablas creadas con la aplicación *aspnet_regsql.exe*.

4.2.2.1.5. Tabla asesor

Esta tabla almacena la información de cada asesor que forma parte de la aplicación, en la tabla V se muestra el nombre de la columna, su tipo y descripción.

Tabla V. **Tabla asesor**

Nombre de la columna	Tipo de la columna	Descripción
<i>Codigo</i>	<i>int</i>	Identificador único del asesor
<i>Nombre</i>	<i>nvarchar(50)</i>	Nombre del asesor
<i>Apellido</i>	<i>nvarchar(50)</i>	Apellido del asesor
<i>Telefono</i>	<i>nvarchar(12)</i>	Número de teléfono del asesor
<i>Celular</i>	<i>nvarchar(12)</i>	Número de celular o teléfono móvil del asesor
<i>Email</i>	<i>nvarchar(100)</i>	Dirección de correo electrónico del asesor
<i>Aprobado</i>	<i>bit</i>	1 = aprobado, 0 = no aprobado
<i>Catedratico</i>	<i>bit</i>	1 = es catedrático, 0 = no es catedrático (sólo aplica para Escuela de Ciencias y Sistemas)
<i>Curriculum</i>	<i>nvarchar(125)</i>	Dirección del archivo que contiene el <i>Curriculum Vitae</i> del asesor

A continuación se muestra el *script* utilizado para la creación de esta tabla:

```
CREATE TABLE asesor (
    codigo INT IDENTITY(1,1) PRIMARY KEY,
    nombre NVARCHAR(50) NOT NULL,
    apellido NVARCHAR(50) NOT NULL,
    telefono NVARCHAR(12),
    celular NVARCHAR(12),
    email NVARCHAR(100),
    aprobado BIT NOT NULL,
    catedratico BIT NOT NULL,
    curriculum NVARCHAR(125)
);
```

4.2.2.1.6. Tabla *estudiante*

Esta tabla almacena la información de cada estudiante que relacionada con la identificación del usuario en la aplicación, en la tabla VI se muestra el nombre de la columna, su tipo y descripción.

Tabla VI. Tabla *estudiante*

Nombre de la columna	Tipo de la columna	Descripción
<i>Carne</i>	<i>int</i>	Carné del estudiante
<i>Nombre</i>	<i>nvarchar(50)</i>	Nombre del estudiante
<i>Apellido</i>	<i>nvarchar(50)</i>	Apellido del estudiante
<i>Telefono</i>	<i>nvarchar(12)</i>	Número de teléfono del estudiante
<i>Celular</i>	<i>nvarchar(12)</i>	Número de celular o teléfono móvil del estudiante
<i>Email</i>	<i>nvarchar(100)</i>	Dirección de correo electrónico del estudiante

Continúa

<i>Asesor</i>	<i>int</i>	Código del asesor asignado actualmente al estudiante
---------------	------------	--

A continuación se muestra el *script* utilizado para la creación de esta tabla:

```
CREATE TABLE estudiante (
    carne INT PRIMARY KEY,
    nombre NVARCHAR(50) NOT NULL,
    apellido NVARCHAR(50) NOT NULL,
    telefono NVARCHAR(12),
    celular NVARCHAR(12),
    email NVARCHAR(100),
    asesor INT,
    CONSTRAINT fk_asesor FOREIGN KEY(asesor) REFERENCES asesor(codigo)
);
```

4.2.2.1.7. Tabla *asignación*

Esta tabla se utiliza para hacer la relación de usuarios y estudiantes en un semestre específico, en la tabla VII se muestra el nombre de la columna, su tipo y descripción.

Tabla VII. **Tabla *asignación***

Nombre de la columna	Tipo de la columna	Descripción
<i>Semestre</i>	<i>tinyint</i>	Identificador único del asesor
<i>Anio</i>	<i>smallint</i>	Nombre del asesor
<i>Usuario</i>	<i>uniqueidentifier</i>	Apellido del asesor
<i>Estudiante</i>	<i>int</i>	Número de teléfono del asesor

A continuación se muestra el *script* utilizado para la creación de esta tabla:

```
CREATE TABLE asignacion (  
    semestre TINYINT NOT NULL,  
    anio SMALLINT NOT NULL,  
    usuario UNIQUEIDENTIFIER NOT NULL,  
    estudiante INT NOT NULL,  
    PRIMARY KEY(semestre, anio, usuario, estudiante),  
    CONSTRAINT fk_usuario FOREIGN KEY(usuario) REFERENCES  
        aspnet_Users(UserId),  
    CONSTRAINT fk_estudiante_asignacion FOREIGN KEY(estudiante)  
        REFERENCES estudiante(carne)  
);
```

4.2.2.1.8. Tabla estado

Esta tabla almacena los estados en que se pueden encontrar los documentos que forman parte de la aplicación, en la tabla VIII se muestra el nombre de la columna, su tipo y descripción.

Tabla VIII. Tabla *estado*

Nombre de la columna	Tipo de la columna	Descripción
<i>Nombre</i>	<i>nvarchar(50)</i>	Nombre del estado, debe ser único
<i>Descripción</i>	<i>nvarchar(150)</i>	Descripción del estado

A continuación se muestra el *script* utilizado para la creación de esta tabla:

```
CREATE TABLE estado (  
    nombre NVARCHAR(50) PRIMARY KEY,  
    descripcion NVARCHAR(150)  
);
```

4.2.2.1.9. Tabla *documento*

Esta tabla almacena la información de cada documento que forma parte de la aplicación, en la tabla IX se muestra el nombre de la columna, su tipo y descripción.

Tabla IX. Tabla *documento*

Nombre de la columna	Tipo de la columna	Descripción
<i>Codigo</i>	<i>int</i>	Identificador único del documento
<i>Titulo</i>	<i>nvarchar(256)</i>	Título del documento
<i>Contenido</i>	<i>ntext</i>	Contenido o índice del documento
<i>Resumen</i>	<i>ntext</i>	Resumen del documento
<i>Vigencia_inicio</i>	<i>datetime</i>	Indica la fecha en la que inicia la validez de un documento
<i>Vigencia_fin</i>	<i>datetime</i>	Indica la fecha en la que finaliza la validez de un documento
<i>Asesor</i>	<i>int</i>	Asesor asignado al estudiante en el momento de la creación o modificación del documento
<i>Documento</i>	<i>nvarchar(125)</i>	Dirección del archivo que contiene el documento completo
<i>Estado</i>	<i>nvarchar(50)</i>	Nombre del estado en el que se encuentra actualmente el documento
<i>Estudiante</i>	<i>int</i>	Carné del estudiante al que pertenece el documento

A continuación se muestra el *script* utilizado para la creación de esta tabla:

```
CREATE TABLE documento (  
    codigo INT IDENTITY(1,1) PRIMARY KEY,  
    titulo NVARCHAR(256) NOT NULL,
```

```

    contenido NTEXT,
    resumen NTEXT,
    vigencia_inicio DATETIME NOT NULL,
    vigencia_fin DATETIME NOT NULL,
    asesor INT,
    documento NVARCHAR(125),
    estado NVARCHAR(50) NOT NULL,
    estudiante INT NOT NULL,
    CONSTRAINT fk_estado FOREIGN KEY(estado) REFERENCES estado(nombre),
    CONSTRAINT fk_estudiante FOREIGN KEY(estudiante) REFERENCES
        estudiante(carne)
);

```

4.2.2.1.10. Tabla *historial*

Esta tabla almacena el historial de los cambios realizados en un documento, también se incluye la creación del mismo; en la tabla X se muestra el nombre de la columna, su tipo y descripción.

Tabla X. Tabla *historial*

Nombre de la columna	Tipo de la columna	Descripción
<i>Correlativo</i>	<i>bigint</i>	Número correlativo para cada entrada en el historial
<i>Estado</i>	<i>nvarchar(50)</i>	Estado del documento en el momento de la creación de la entrada en el historial
<i>Vigencia_inicio</i>	<i>datetime</i>	Fecha de inicio de la validez de un documento en el momento de la creación de la entrada en el historial
<i>Vigencia_fin</i>	<i>datetime</i>	Fecha de fin de la validez de un documento en el momento de la creación de la entrada en el historial

Continúa

<i>Comentario</i>	<i>nvarchar(256)</i>	Observaciones de la entrada en el historial
<i>Documento</i>	<i>int</i>	Código del documento al que pertenece la entrada en el historial
<i>Estudiante</i>	<i>int</i>	Carné del estudiante al que pertenece la entrada en el historial
<i>Asesor</i>	<i>int</i>	Asesor asignado al estudiante en el momento de la creación de la entrada en el historial

A continuación se muestra el *script* utilizado para la creación de esta tabla:

```
CREATE TABLE historial (  
    correlativo BIGINT IDENTITY(1,1) PRIMARY KEY,  
    estado NVARCHAR(50) NOT NULL,  
    vigencia_inicio DATETIME NOT NULL,  
    vigencia_fin DATETIME NOT NULL,  
    comentario NVARCHAR(256),  
    documento INT NOT NULL,  
    estudiante INT NOT NULL,  
    asesor INT,  
    CONSTRAINT fk_documento FOREIGN KEY(documento) REFERENCES  
        documento(codigo),  
    CONSTRAINT fk_estudiante_historial FOREIGN KEY(estudiante) REFERENCES  
        estudiante(carne),  
    CONSTRAINT fk_asesor_historial FOREIGN KEY(asesor) REFERENCES  
        asesor(codigo)  
);
```

4.2.2.2. Procedimientos almacenados (*stored procedures*)

El acceso a la base de datos se realiza a través de procedimientos almacenados, estos son rutinas SQL precompiladas que son almacenadas en la base de datos.

La ventaja de utilizar procedimientos almacenados es que proporcionan mejor rendimiento ya que se pueden ejecutar varias consultas en un solo procedimiento y a la vez proporcionan independencia del código implementado en la aplicación, esto quiere decir que un procedimiento almacenado puede ser modificado sin afectar el funcionamiento o estructura de la aplicación.

También reducen el tráfico de la red porque únicamente es necesario hacer una llamada con el nombre del procedimiento y no todo el texto de la consulta.

A continuación en la tabla XI se describen los procedimientos almacenados utilizados en la aplicación.

Tabla XI. **Procedimientos almacenados**

Procedimiento almacenado	Descripción
<i>usp_UpdateRealName</i>	Agrega el Nombre Real del usuario a la tabla <i>aspnet_Membership</i>
<i>usp_GetRealName</i>	Obtiene el Nombre Real de un usuario en la tabla <i>aspnet_Membership</i>
<i>usp_GetUsers</i>	Retorna todos los usuarios de la aplicación
<i>usp_DeleteUser</i>	Elimina un usuario por medio de <i>UserName</i>
<i>usp_GetRoles</i>	Retorna todos los roles de la aplicación
<i>usp_CreateRole</i>	Crea un nuevo rol en la aplicación
<i>usp_DeleteRole</i>	Elimina un rol por medio de <i>RoleName</i>

Continúa

<i>usp_UpdateRole</i>	Modifica un rol por medio de RoleName
<i>usp_GetAdvisers</i>	Retorna todos los asesores de la aplicación
<i>usp_CreateAdviser</i>	Crea un nuevo asesor en la aplicación
<i>usp_UpdateCurriculum</i>	Agrega la dirección del archivo que contiene el Curriculum de un asesor
<i>usp_DeleteAdviser</i>	Elimina un asesor por medio de su código
<i>usp_GetAdviser</i>	Obtiene un asesor por medio de su código
<i>usp_UpdateAdviser</i>	Modifica un asesor por medio de su código
<i>usp_CreateStudent</i>	Crea un nuevo estudiante en la aplicación
<i>usp_CreateAssignment</i>	Crea una asignación para un usuario y un estudiante en un semestre específico
<i>usp_GetStudents</i>	Retorna todos los estudiantes de la aplicación
<i>usp_DeleteStudent</i>	Elimina un estudiante por medio de su carné
<i>usp_AssignAdviser</i>	Asigna un asesor a un estudiante
<i>usp_GetStudent</i>	Obtiene un estudiante por medio de su carné
<i>usp_UpdateStudent</i>	Modifica un estudiante por medio de su carné
<i>usp_GetStates</i>	Retorna todos los estados de documentos de la aplicación
<i>usp_CreateState</i>	Crea un nuevo estado en la aplicación
<i>usp_DeleteState</i>	Elimina un estado por medio de su nombre
<i>usp_UpdateState</i>	Modifica un estado por medio de su nombre
<i>usp_GetDocuments</i>	Retorna todos los documentos de la aplicación
<i>usp_CreateDocument</i>	Crea un nuevo documento en la aplicación
<i>usp_UpdateDocumentFile</i>	Agrega la dirección del archivo que contiene el documento completo en la tabla <i>documento</i>
<i>usp_GetDocument</i>	Obtiene un documento por medio de su código
<i>usp_GetContent</i>	Obtiene el contenido o índice de un documento por su código

Continúa

<i>usp_GetSummary</i>	Obtiene el resumen de un documento por medio de su código
<i>usp_UpdateDocument</i>	Modifica un documento por medio de su código
<i>usp_DeleteDocument</i>	Elimina un documento por medio de su código
<i>usp_GetHistory</i>	Obtiene los registros de la tabla historial
<i>usp_VerifyAssign</i>	Verifica si existe alguna asignación para un estudiante y un usuario
<i>usp_GetLastDocuments</i>	Obtiene los últimos 5 trabajos finalizados
<i>usp_GetAdviserUnfinishedDocuments</i>	Obtiene todos los documentos no finalizados de un asesor
<i>usp_GetAdviserFinishedDocuments</i>	Obtiene todos los documentos finalizados de un asesor
<i>usp_GetProtocols</i>	Obtiene todos los protocolos
<i>usp_GetUnfinishedDocuments</i>	Obtiene todos los documentos no finalizados
<i>usp_GetFinishedDocuments</i>	Obtiene todos los documentos finalizados

Para ver el *script* de creación de cada procedimiento almacenado busque el archivo *sp.sql* en el disco de instalación de la aplicación o bien consulte la base de datos cuando ya esté instalada.

4.2.2.3. Información predeterminada de la base de datos

Al momento de crear la base de datos se incluyen de forma predeterminada roles, usuarios y estados que son necesarios para el funcionamiento inicial de la aplicación, a continuación se describen cada uno de ellos.

4.2.2.3.1. Roles predeterminados

Se incluyen tres roles de forma predeterminada, estos son creados con el procedimiento almacenado *usp_CreateRole* que incluye los parámetros: nombre del rol, descripción del rol, nombre de la aplicación y código de error (este último es un parámetro de salida por lo tanto se escribe la palabra *NULL* que significa que ningún valor ha sido enviado). A continuación se describe cada rol:

- Administrador: Administrador del sitio *Web*.
- Tutor: Puede modificar estudiantes, asesores y documentos.
- Digitador: Únicamente puede ingresar datos.

A continuación se muestra el *script* utilizado para la creación de los roles predeterminados:

```
EXEC usp_CreateRole 'Administrador','Administrador del sitio Web.','TESAURO',NULL
EXEC usp_CreateRole 'Tutor',' Puede modificar estudiantes, asesores y
documentos.','TESAURO',NULL
EXEC usp_CreateRole 'Digitador','Únicamente puede ingresar datos.','TESAURO',NULL
```

4.2.2.3.2. Usuarios predeterminados

Únicamente se incluye un usuario de forma predeterminada, este es el usuario Administrador que posee todos los permisos sobre la aplicación, a continuación se describe el usuario:

- Nombre de usuario: Administrador.
- Nombre real: Administrador.
- Contraseña: *@dm!n*.
- Pregunta de seguridad: *Hola*.

- Respuesta de seguridad: *Mund0*.
- Correo electrónico: En blanco.
- Rol: Administrador.

Se recomienda cambiar inmediatamente la contraseña, pregunta de seguridad y respuesta de seguridad luego de instalar la aplicación.

A continuación se muestra el *script* utilizado para la creación del usuario predeterminado:

```

DECLARE @Date DATETIME
SET @Date = GETDATE()
EXEC aspnet_Membership_CreateUser 'TESAURO','Administrador',
    '6tWqm8Z4M6boFrs1DIS1AOdAj9j1H1OTsbPJqVI2pu8=',
    'vI8axbwbMotOgRdqGP4XHQ==','','hola',
    '6tWqm8Z4M6boFrs1DIS1AHsS7Z4jQ4ig4PiTWrZFtJI=',
    1, @Date, @Date, 1,2, NULL
EXEC usp_updateRealName 'Administrador','Administrador',NULL
EXEC aspnet_UsersInRoles_AddUsersToRoles 'TESAURO','Administrador',
    'Administrador', @Date

```

4.2.2.3.3. Estados predeterminados

Se incluyen ocho estados de forma predeterminada, estos son creados con el procedimiento almacenado *usp_CreateState* que incluye los parámetros: nombre del estado, descripción del estado y código de error (este último es un parámetro de salida por lo tanto se escribe la palabra *NULL* que significa que ningún valor ha sido enviado). A continuación se describe cada estado:

- Pendiente aprobación: Pendiente de aprobación por parte del tutor.
- Rechazado: Rechazado por parte del tutor.

- Aprobado: Aprobado por parte del tutor.
- Revisión asesor: Trabajo con el asesor.
- Revisión revisor: Revisión del coordinador de área o revisor.
- Revisión lingüística: Revisión del especialista en lingüística.
- Revisión director: Revisión del Director de Escuela.
- Finalizado: Trabajo de graduación terminado.

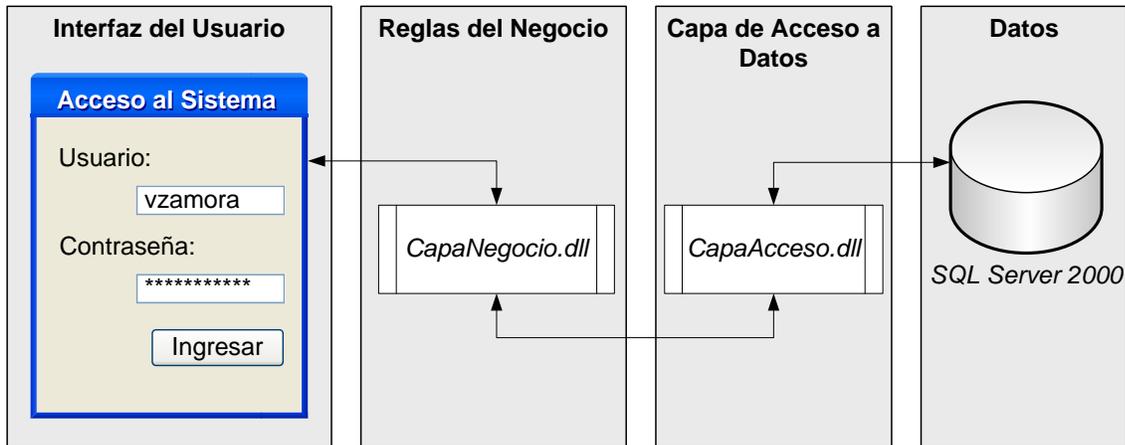
A continuación se muestra el *script* utilizado para la creación de los estados predeterminados:

```
EXEC usp_CreateState 'Pendiente Aprobación','Pendiente de aprobación por parte del
tutor.', NULL
EXEC usp_CreateState 'Rechazado','Rechazado por parte del tutor.',NULL
EXEC usp_CreateState 'Aprobado','Aprobado por parte del tutor.', NULL
EXEC usp_CreateState 'Revisión Asesor','Trabajo con el asesor.', NULL
EXEC usp_CreateState 'Revisión Revisor','Revisión del coordinador de área o revisor.',
NULL
EXEC usp_CreateState 'Revisión Lingüística','Revisión del especialista en lingüística.',
NULL
EXEC usp_CreateState 'Revisión Director','Revisión del Director de Escuela.', NULL
EXEC usp_CreateState 'Finalizado','Trabajo de graduación terminado.', NULL
```

4.2.3. Implementación de la aplicación

La arquitectura seleccionada para la implementación de la aplicación es la de arquitectura flexible de n capas, con esto se logra una independencia mayor entre los datos, las reglas del negocio y la interfaz del usuario. El diagrama de implementación se puede observar en la figura 83.

Figura 83. **Arquitectura de la aplicación**



Como se puede observar en la figura 79, la aplicación está dividida en 4 capas: Interfaz del usuario, reglas del negocio, acceso a datos y los datos. En los siguientes apartados se detalla cada una de estas capas.

4.2.3.1. Capa de datos

Esta capa hace referencia al uso de *SQL Server 2000* como manejador de datos, para obtener más detalles de las características e implementación diríjase a los apartados: *4.2.1 Tecnología utilizada* y *4.2.2 Implementación de la base de datos*, respectivamente.

4.2.3.2. Capa de acceso a datos

La capa de acceso de datos se encuentra implementada en un proyecto llamado *CapaAcceso* dentro de la solución *Tesouro* de *Visual Studio 2005*, esta puede ser encontrada en el disco de instalación de la aplicación.

En la tabla XII se describen los principales métodos y funciones que forman parte de la capa de acceso de datos.

Tabla XII. **Métodos y funciones de la capa de acceso a datos**

Método o función	Descripción
<i>New</i>	Constructor de la clase, recibe como parámetro la cadena de conexión para la base de datos
<i>runSPDataSet</i>	Esta función retorna un objeto <i>DataSet</i> , toma como parámetro el nombre de un procedimiento almacenado
<i>runSPDataReader</i>	Esta función retorna un objeto <i>DataReader</i> , toma como parámetro el nombre de un procedimiento almacenado
<i>runSPOutput</i>	Esta función retorna un <i>array list</i> (lista de cadenas), toma como parámetro el nombre de un procedimiento almacenado, esta función se utiliza para retornar los parámetros de salida del procedimiento almacenado

Como se pudo observar en la tabla XII las funciones reciben como parámetro el nombre de un procedimiento almacenado, a continuación se explica con un ejemplo como hacer uso de la función *runSPOutput*, las otras funciones se usan de la misma forma.

Se usará como ejemplo la función *updateRealName* de la capa de reglas del negocio para el ejemplo.

Lo primero es hacer una instancia de la clase *SQLServer* del proyecto *CapaAcceso* de la siguiente forma:

```
Dim localOutPutServer As New SQLServer(ConnectionString)
```

Luego se agregan los parámetros que el procedimiento almacenado necesita. Se debe indicar el nombre del parámetro, valor, tipo, tamaño e indicar si es de salida o de entrada. A continuación se muestra el código para agregar parámetros.

```

localOutPutServer.AddParameter("@UserName", UserName,
    SQLServer.SqlDataType.SqlNVarChar, 256, ParameterDirection.Input)
localOutPutServer.AddParameter("@RealName", RealName,
    SQLServer.SqlDataType.SqlNVarChar, 100, ParameterDirection.Input)
localOutPutServer.AddParameter("@ErrorCode", ,
    SQLServer.SqlDataType.SqlInteger, , ParameterDirection.Output)

```

Finalmente se indica el nombre del procedimiento almacenado y se hace el llamado a la función *runSPOOutput* como se muestra a continuación:

```

ParamsStoredProcedure = "usp_updateRealName"
localDSOutput = localOutPutServer.runSPOOutput(ParamsStoredProcedure)

```

La variable *localDSOutput* es de tipo *ArrayList* y se utiliza para retornar los parámetros de salida si existen.

4.2.3.3. Capa de reglas del negocio

De la misma forma que la capa acceso de datos, la capa de reglas del negocio se encuentra implementada en la solución *Tesouro* de *Visual Studio 2005* en el proyecto llamado *CapaNegocio*.

Esta capa tiene varias clases que manejan los usuarios, asesores, estudiantes y documentos. En los siguientes apartados se explican los métodos y funciones de cada clase.

4.2.3.3.1. Clase *AppUser*

Esta clase maneja todas las reglas del negocio relacionadas con los usuarios de la aplicación.

En la tabla XIII se describen los principales métodos y funciones que forman parte de la clase *AppUser*.

Tabla XIII. **Métodos y funciones de la clase *AppUser***

Método o función	Descripción
<i>New</i>	Constructor de la clase, recibe como parámetro la cadena de conexión para la base de datos
<i>updateRealName</i>	Esta función modifica el nombre del real de un usuario, toma como parámetros el nombre del usuario y el nombre real del usuario
<i>getUsers</i>	Esta función retorna todos los usuarios que forman parte de la aplicación en un <i>DataSet</i> , no tiene parámetros
<i>getRealName</i>	Esta función retorna el nombre real de un usuario tomando como parámetro el nombre de usuario

4.2.3.3.2. Clase *AppAdviser*

Esta clase maneja todas las reglas del negocio relacionadas con los asesores de la aplicación. En la tabla XIV se describen los principales métodos y funciones que forman parte de la clase *AppAdviser*.

Tabla XIV. **Métodos y funciones de la clase *AppAdviser***

Método o función	Descripción
<i>New</i>	Constructor de la clase, recibe como parámetro la cadena de conexión para la base de datos
<i>createAdviser</i>	Esta función crea un asesor en la aplicación, toma como parámetros el nombre, apellido, teléfono, celular, correo electrónico, si esta aprobado, si es catedrático y la dirección del archivo que contiene el <i>curriculum vitae</i> del asesor

Continúa

<i>updateCurriculum</i>	Esta función modifica la dirección del archivo que contiene el <i>curriculum vitae</i> de un asesor por medio de código
<i>getAdviser</i>	Esta función retorna los datos de un asesor en un <i>DataSet</i> por medio de su código
<i>updateAdviser</i>	Esta función modifica un asesor en la aplicación, toma como parámetros el código, nombre, apellido, teléfono, celular, correo electrónico, si esta aprobado, si es catedrático y la dirección del archivo que contiene el <i>curriculum vitae</i> del asesor

4.2.3.3.3. Clase *AppStudent*

Esta clase maneja todas las reglas del negocio relacionadas con los estudiantes de la aplicación. En la tabla XV se describen los principales métodos y funciones que forman parte de la clase *AppStudent*.

Tabla XV. **Métodos y funciones de la clase *AppStudent***

Método o función	Descripción
<i>New</i>	Constructor de la clase, recibe como parámetro la cadena de conexión para la base de datos
<i>createStudent</i>	Esta función crea un estudiante en la aplicación, toma como parámetros el carné, nombre, apellido, teléfono, celular y correo electrónico, del estudiante
<i>createAssignment</i>	Esta función crea una asignación entre un usuario y un estudiante en un semestre específico, toma como parámetros el número de semestre, el año, el nombre de usuario, el carné del estudiante y una variable que indica si la asignación debe ser actualizada o no

Continúa

<i>createAdviserAssignment</i>	Esta función crea una asignación entre un asesor y un estudiante, toma como parámetros el carné del estudiante, el código del asesor y una variable que indica si la asignación debe ser actualizada o no
<i>getStudent</i>	Esta función retorna los datos de un estudiante en un <i>DataSet</i> por medio de su carné
<i>updateStudent</i>	Esta función modifica un estudiante en la aplicación, toma como parámetros el carné, nombre, apellido, teléfono, celular, correo electrónico y código del asesor que ha sido asignado al estudiante

4.2.3.3.4. Clase *AppDocument*

Esta clase maneja todas las reglas del negocio relacionadas con los documentos de la aplicación. En la tabla XVI se describen los principales métodos y funciones que forman parte de la clase *AppDocument*.

Tabla XVI. **Métodos y funciones de la clase *AppDocument***

Método o función	Descripción
<i>New</i>	Constructor de la clase, recibe como parámetro la cadena de conexión para la base de datos
<i>createDocument</i>	Esta función crea un documento en la aplicación, toma como parámetros el carné del estudiante, título, contenido o índice, resumen, fecha de inicio de validez, fecha de finalización de validez, estado y comentario del documento
<i>updateDocumentFile</i>	Esta función modifica la dirección del archivo que contiene el documento completo por medio del código de documento
<i>getDocument</i>	Esta función retorna los datos de un documento en un <i>DataSet</i> por medio de su código

Continúa

<i>updateDocument</i>	Esta función modifica un documento en la aplicación, toma como parámetros el código del documento, carné del estudiante, título, contenido o índice, resumen, fecha de inicio de validez, fecha de finalización de validez, estado, dirección del archivo que contiene el documento completo, y comentario del documento
<i>getContent</i>	Esta función retorna el contenido o índice de un documento por medio de su código
<i>getSummary</i>	Esta función retorna el resumen de un documento por medio de su código
<i>verifyAssign</i>	Esta función verifica si existe alguna asignación entre un usuario y un estudiante, toma como parámetros el carné del estudiante y el nombre de usuario
<i>getDocumentCode</i>	Retorna el código del último documento de un estudiante, toma como parámetro el carné del estudiante

4.2.3.4. Capa de interfaz del usuario

En esta capa se maneja la comunicación de los usuarios con la aplicación y para hacer que esta sea lo más fácil posible se utilizan componentes adicionales y controles de usuario, estos se describen a continuación.

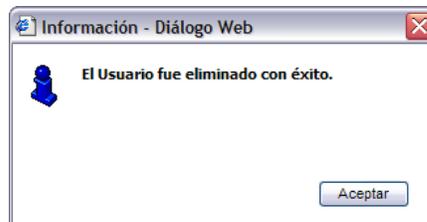
También se proporciona una pequeña descripción de lo que hace cada página que forma parte de la aplicación *Web*.

4.2.3.4.1. Componentes adicionales

Para agregar un componente únicamente debe copiar el archivo en la carpeta *bin* que se encuentra en el directorio de la aplicación. A continuación se describen los componentes utilizados:

- *Cooperator.Framework.Web.dll*: Este es un control *MsgBox* para *ASP.NET 2.0* que puede ser descargado desde el *blog* de Eugenio Serrano (<http://eugenioserrano.spaces.msn.com/PersonalSpace.aspx>), se encuentra ubicado en los archivos de diciembre de 2005; la versión utilizada en el proyecto fue la 1.2. Es útil ya que *ASP.NET 2.0* no incluye una forma para poder mostrar mensajes de información, advertencia o error al usuario, al utilizar este componente se pueden mostrar mensajes como el que se muestra en la figura 84. Las instrucciones de uso y ejemplos forman parte del archivo que se descarga desde el *blog*.

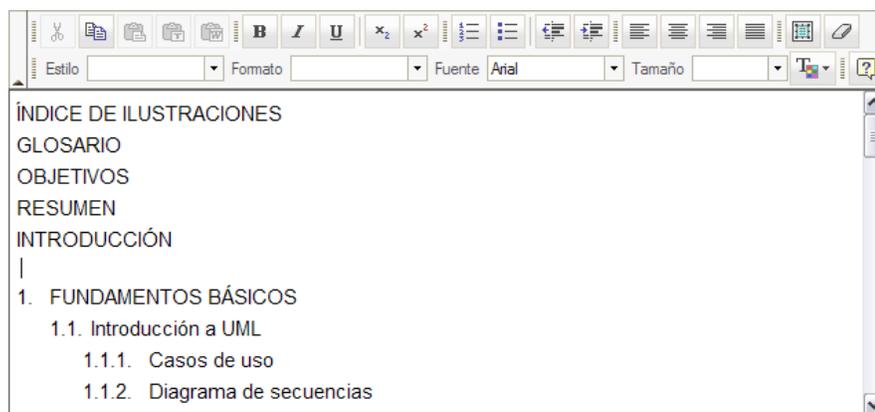
Figura 84. **Cooperator.Framework.Web.dll**



- *FredCK.FCKeditorV2.dll*: Para el envío de correos electrónicos y la edición del contenido y resumen de los documentos se utilizó un editor que permite escribir texto con formato como colores, tipos de fuentes, tamaño, etc. El editor "*FCK Editor*" permite crear texto con formato sin tener la necesidad de conocer *HTML*, esto es útil para los usuarios finales que no tienen conocimientos técnicos y deben usar la aplicación.

El editor puede ser descargado del sitio *FCKeditor* (<http://www.fckeditor.net/>), aquí encontrará la última versión y la documentación necesaria para su uso; la versión utilizada en el proyecto fue la 2.3. Un ejemplo del uso del editor para crear texto con formato puede observarse en la figura 85.

Figura 85. **FredCK.FCKeditorV2.dll**



- *GMDDatePicker.dll*: Este componente se utiliza para facilitar a los usuarios el ingreso de fechas en la aplicación, es un control gráfico (ver figura 86) que permite seleccionar la fecha de una forma fácil y sin la posibilidad de cometer un error en el formato o rango de validez. Puede descárgalo del sitio *GrayMatterSoft* (<http://www.graymattersoft.com/>), donde encontrará la última versión y la documentación necesaria para su uso; la versión utilizada en el proyecto fue la 0.8.3.

Figura 86. **GMDDatePicker.dll**



- *FlashControl.dll*: Este es un control que permite el uso de película *Flash* en aplicaciones *ASP.NET 2.0*. Puede descárgalo del sitio *Flash Control for ASP.NET 2.0* (<http://www.flash-control.net/>), donde encontrará la última versión y la documentación necesaria para su uso; la versión utilizada en el proyecto fue la 2.4.

4.2.3.4.2. Controles de usuario

Adicionalmente a los controles que vienen incluidos con *Visual Studio 2005* y que son llamados controles de servidor, se puede crear controles propios que son llamados controles de usuario; un control de usuario es un grupo de uno o más controles de servidor o elementos *HTML* que encapsulan una pieza de funcionalidad. A continuación en la tabla XII se describen los controles de usuario utilizados en la aplicación:

Tabla XVII. **Controles de usuario**

Control de usuario	Descripción
<i>assignAdviser.ascx</i>	Se utiliza para crear o modificar la asignación de un asesor a un estudiante
<i>assignTutor.ascx</i>	Se utiliza para crear o modificar la asignación de un tutor a un estudiante en un semestre específico
<i>changePassword.ascx</i>	Se utiliza para cambiar la contraseña de un usuario
<i>changeSecurity.ascx</i>	Se utiliza para cambiar la pregunta y respuesta de seguridad de un usuario
<i>editUser.ascx</i>	Se utiliza para cambiar los datos personales de un usuario
<i>lastDocuments.ascx</i>	Se utiliza para mostrar los últimos cinco trabajos finalizados
<i>sendMail.ascx</i>	Se utiliza para crear un correo electrónico con datos adjuntos y enviarlo a los destinatarios ingresados

4.2.3.4.3. Páginas Web

También llamadas formularios *Web* (*Web forms*), son documentos que utilizan código *HTML* y controles de servidor para crear páginas que pueden ser vistas en un explorador *Web*.

En la tabla XVIII se describen las páginas *Web* utilizadas en la aplicación:

Tabla XVIII. Páginas *Web*

Página <i>Web</i>	Descripción
Páginas de administración	
<i>advisers.aspx</i>	En esta página se administran todos los asesores del sistema, se muestra un listado con los datos personales y se dan las opciones de agregar, modificar y eliminar asesores
<i>createAdviser.aspx</i>	Aquí se ingresan y verifican los datos necesarios para la creación de un asesor
<i>createDocument.aspx</i>	Aquí se ingresan y verifican los datos necesarios para la creación de un documentos
<i>createStudent.aspx</i>	Aquí se ingresan y verifican los datos necesarios para la creación de un estudiante
<i>createUser.aspx</i>	Aquí se ingresan y verifican los datos necesarios para la creación de un usuario
<i>default.aspx</i>	Es la página predeterminada de la sección de administración, se muestran las acciones que pueden realizar los diferentes roles del sistema

Continúa

<i>documents.aspx</i>	En esta página se administran todos los documentos del sistema, se muestra un listado con los datos principales y se dan las opciones de agregar, modificar y eliminar. También se puede ver el contenido o índice, el resumen, documento completo y el historial de cada documento
<i>editAdviser.aspx</i>	Aquí se modifican y verifican los datos de un asesor
<i>editDocument.aspx</i>	Aquí se modifican y verifican los datos de un documento
<i>editStudent.aspx</i>	Aquí se modifican y verifican los datos de un estudiante
<i>editUser.aspx</i>	Aquí se modifican y verifican los datos de un usuario
<i>loadStudents.aspx</i>	En esta página se realiza una carga de estudiantes al sistema por medio de un archivo delimitado por comas
<i>roles.aspx</i>	En esta página se administran todos los roles del sistema, se muestra un listado con los datos del rol y se dan las opciones de agregar, modificar y eliminar roles
<i>states.aspx</i>	En esta página se administran todos los estados del sistema, se muestra un listado con los datos del estado y se dan las opciones de agregar, modificar y eliminar estados

Continúa

<i>students.aspx</i>	En esta página se administran todos los estudiantes del sistema, se muestra un listado con los datos personales y se dan las opciones de agregar, modificar y eliminar estudiantes. También se puede aplicar un filtro para cambiar el contenido a mostrar y existe la opción de enviar correo electrónico, asignar tutor o asesor a un estudiante
<i>users.aspx</i>	Aquí se administran todos los usuarios del sistema, se muestra un listado con los datos personales y se dan las opciones de agregar, modificar y eliminar usuarios
<i>viewDocuments.aspx</i>	Se utiliza para mostrar los documentos de un estudiante, empezando por el último.
Páginas de usuario final	
<i>accessDenied.aspx</i>	Esta página se utiliza para mostrar un mensaje de error en el caso de que un usuario no haya escrito bien su usuario o contraseña, o no tenga permisos para ver una página
<i>default.aspx</i>	Es la página predeterminada de la aplicación, se muestran las opciones para los usuarios finales como los últimos cinco trabajos finalizados y un menú con las opciones de búsqueda
<i>editProfile.aspx</i>	En esta página se muestran las opciones para modificar los datos personales, cambiar la contraseña y cambiar la pregunta y respuesta de seguridad del usuario que ha iniciado sesión en el sistema

Continúa

<i>error.aspx</i>	Aquí se muestran los errores producidos al ingresar una página o recurso que no existe, también se muestran los errores que puedan ocurrir en el servidor pero de una manera que no compromete la seguridad ni el código fuente de la aplicación
<i>logOut.aspx</i>	Aquí se muestra un mensaje indicando que la sesión ha sido finalizada con éxito
<i>recoverPassword.aspx</i>	Esta página se utiliza para poder recuperar la contraseña de un usuario por medio de su pregunta y respuesta de seguridad
<i>searchAdvisers.aspx</i>	Aquí se muestra un listado de los asesores con sus datos personales y la opción de ver los documentos finalizados y no finalizados del asesor
<i>searchFinishedDocuments.aspx</i>	En esta página se muestran todos los documentos finalizados, se muestra un listado con los datos principales y se puede ver el contenido o índice, el resumen y el documento completo. Se pueden realizar consultas que buscan en el título, contenido o índice y en el resumen
<i>searchProtocols.aspx</i>	En esta página se muestran todos los protocolos, se muestra un listado con los datos principales y se puede ver el contenido o índice, el resumen y el documento completo. Se pueden realizar consultas que buscan en el título, contenido o índice y en el resumen. También se puede seleccionar mostrar los temas no disponibles, es decir lo que su periodo de vigencia ha terminado

Continúa

<i>searchUnfinishedDocuments.aspx</i>	En esta página se muestran todos los no documentos finalizados, se muestra un listado con los datos principales y se puede ver el contenido o índice, el resumen y el documento completo. Se pueden realizar consultas que buscan en el título, contenido o índice y en el resumen. También se puede seleccionar mostrar los temas no disponibles, es decir lo que su periodo de vigencia ha terminado
<i>viewAdviser.aspx</i>	Se muestran los datos personales de un asesor y sus documentos finalizados y no finalizados
<i>viewStudent.aspx</i>	Se muestran los datos personales de un estudiante

4.2.3.5. Configuración

Visual Studio 2005 utiliza archivos para configurar las aplicaciones Web que se desarrollan con *ASP.NET 2.0*, para la aplicación desarrollada se utilizaron únicamente dos archivos: el archivo *web.config* y el archivo *web.sitemap*, estos se explican en los siguientes apartados.

4.2.3.5.1. Archivo *web.config*

Es un archivo en *XML* que define la configuración de una aplicación *Web* en *ASP.NET 2.0*, una de las ventajas de esta forma de configuración es que el archivo puede ser modificado mientras la aplicación se encuentra en ejecución y los cambios se actualizan de forma automática y sin necesidad de reiniciar el servidor.

El archivo es protegido automáticamente prohibiendo su acceso desde Internet, únicamente se puede configurar desde el servidor local o bien desde la herramienta de configuración del servidor *Web*.

A continuación se detallan las secciones utilizadas en la aplicación desarrollada:

- Sección *connectionStrings*: Se utiliza para especificar la colección de cadenas de conexión hacia la base de datos. A continuación se muestra un ejemplo de su utilización con una pequeña descripción de cada elemento:

```
<connectionStrings>
  <clear/>
  <add name="TESAUROConnectionString" connectionString="Data
    Source=localhost;Initial Catalog=TESAURO;Persist Security
    Info=True;User ID=user;Password=password"
    providerName="System.Data.SqlClient"/>
</connectionStrings>
```

- El elemento *clear* se utiliza para remover todas las cadenas de conexión que puedan ser heredadas del archivo de configuración global.
- El elemento *add* se utiliza para agregar una nueva cadena de conexión; a continuación una descripción de sus propiedades:
 - *name*: Nombre de la cadena de conexión.
 - *connectionString*: La cadena de conexión.

- *providerName*: El proveedor utilizado, en este caso *SQL Server*.
- Sección *location*: La funcionalidad de esta sección puede ser consultada en el apartado: 4.1.6.3 Administrar roles.
- Sección *compilation*: Se utiliza para especificar la configuración utilizada en la compilación por *ASP.NET 2.0*. A continuación se muestra un ejemplo de su utilización con una pequeña descripción de cada elemento:

```
<compilation debug="true">
  <assemblies>
    <add assembly="System.Design, Version=2.0.0.0, Culture=neutral,
      PublicKeyToken=B03F5F7F11D50A3A"/>
  </assemblies>
</compilation>
```

- La propiedad *debug* de *compilation* se utiliza para poder depurar la aplicación cuando su valor es *true*, esta opción afecta el rendimiento por lo que debe utilizarse únicamente den la fase de desarrollo.
- El elemento *assemblies* define los nombres de los ensamblados que son incluidos durante la compilación de la aplicación, estos a son agregados automáticamente por el compilador por lo que no se recomienda su modificación.

- Sección *membership*: En esta sección se configuran los parámetros para el manejo y autenticación de las cuentas de usuario. A continuación se muestra un ejemplo de su utilización con una pequeña descripción de cada elemento:

```

<membership>
  <providers>
    <clear/>
    <add name="AspNetSqlMembershipProvider"
      type="System.Web.Security.SqlMembershipProvider, System.Web,
        Version=2.0.0.0, Culture=neutral,
        PublicKeyToken=b03f5f7f11d50a3a"
      connectionStringName="TESAUROConnectionString"
      enablePasswordRetrieval="true"
      enablePasswordReset="true"
      requiresQuestionAndAnswer="true"
      applicationName="TESAURO"
      requiresUniqueEmail="true"
      passwordFormat="Encrypted"
      maxInvalidPasswordAttempts="20"
      minRequiredPasswordLength="5"
      minRequiredNonalphanumericCharacters="0"
      passwordAttemptWindow="10"
      passwordStrengthRegularExpression=""/>
  </providers>
</membership>

```

- El elemento *providers* se utiliza para definir la colección de proveedores de membresía.
- El elemento *clear* se utiliza para remover todos los proveedores que puedan ser heredados del archivo de configuración global.

- El elemento *add* se utiliza para agregar un nuevo proveedor de membresía; a continuación una descripción de sus propiedades:
 - *name*: Nombre del proveedor de membresía.
 - *type*: Especifica el tipo de proveedor que se utiliza, en este caso *SQL Server*.
 - *connectionStringName*: Especifica el nombre de una conexión definida en la sección *connectionStrings*.
 - *enablePasswordRetrieval*: Especifica si es posible recuperar la contraseña de un usuario.
 - *enablePasswordReset*: Especifica si es posible restablecer la contraseña de un usuario.
 - *requiresQuestionAndAnswer*: Especifica si es necesario el uso de una pregunta y respuesta de seguridad para recuperar o restablecer una contraseña.
 - *applicationName*: Nombre de la aplicación con la que es almacenada la información en la base de datos.
 - *requiresUniqueEmail*: Especifica si el correo electrónico de cada usuario debe ser único.
 - *passwordFormat*: Especifica la forma en que la contraseña es almacenada en la base de datos.
 - *maxInvalidPasswordAttempts*: Especifica el número máximo de intentos inválidos al usar la contraseña o al contestar la pregunta de seguridad.
 - *minRequiredPasswordLength*: Especifica el mínimo de caracteres que debe llevar una contraseña, el máximo es 128.

- *minRequiredNonalphanumericCharacters*: Especifica el mínimo de caracteres no alfanuméricos que debe llevar una contraseña, este valor no puede ser mayor que *minRequiredPasswordLength*.
 - *passwordAttemptWindow*: Es el número de minutos durante los que los intentos inválidos de contraseña o respuesta de seguridad son registrados.
 - *passwordStrengthRegularExpression*: Especifica una expresión regular para evaluar la contraseña.
- Sección *roleManager*: En esta sección se configuran los parámetros para el manejo de los roles. A continuación se muestra un ejemplo de su utilización con una pequeña descripción de cada elemento:

```

<roleManager enabled="true" cacheRolesInCookie="true"
cookieName="TESAURO_ROLES">
  <providers>
    <clear/>
    <add name="AspNetSqlRoleProvider"
      connectionStringName="TESAUROConnectionString"
      applicationName="TESAURO"
      type="System.Web.Security.SqlRoleProvider,
        System.Web, Version=2.0.0.0, Culture=neutral,
        PublicKeyToken=b03f5f7f11d50a3a"/>
  </providers>
</roleManager>

```

- La propiedad *enabled* de *roleManager* especifica si el manejo de roles está habilitado o no.

- La propiedad *cacheRolesInCookie* de *roleManager* especifica si el listado de roles es almacenado en una *cookie* para mejorar el rendimiento de la aplicación.
- La propiedad *cookieName* de *roleManager* especifica el nombre de la *cookie* utilizada para almacenar los roles.
- El elemento *providers* se utiliza para definir la colección de proveedores de manejo de roles.
- El elemento *clear* se utiliza para remover todos los proveedores que puedan ser heredados del archivo de configuración global.
- El elemento *add* se utiliza para agregar un nuevo proveedor de manejo de roles; a continuación una descripción de sus propiedades:
 - *name*: Nombre del proveedor de manejo de roles.
 - *type*: Especifica el tipo de proveedor que se utiliza, en este caso *SQL Server*.
 - *connectionStringName*: Especifica el nombre de una conexión definida en la sección *connectionStrings*.
 - *applicationName*: Nombre de la aplicación con la que es almacenada la información en la base de datos.
- Sección *siteMap*: En esta sección se configura el soporte de infraestructura de navegación para configurar, guardar, e interpretar la navegación de sitio. A continuación se muestra un ejemplo de su utilización con una pequeña descripción de cada elemento:

```

<siteMap enabled="true">
  <providers>
    <clear/>
    <add name="XmlSiteMapProvider"
      type="System.Web.XmlSiteMapProvider, System.Web,
        Version=2.0.0.0,Culture=neutral,
        PublicKeyToken=b03f5f7f11d50a3a"
      siteMapFile="web.sitemap"
      securityTrimmingEnabled="true"/>
  </providers>
</siteMap>

```

- El elemento *providers* se utiliza para definir la colección de proveedores de navegación.
- El elemento *clear* se utiliza para remover todos los proveedores que puedan ser heredados del archivo de configuración global.
- El elemento *add* se utiliza para agregar un nuevo proveedor de navegación; a continuación una descripción de sus propiedades:
 - *name*: Nombre del proveedor de navegación.
 - *type*: Especifica el tipo de proveedor que se utiliza, en este caso es de tipo *XML*.
 - *siteMapFile*: Especifica el archivo que contiene los datos de mapa de sitio, este archivo es explicado en el siguiente apartado.

- *securityTrimmingEnabled*: Permite el uso de roles para restringir la navegación del sitio.
- Sección *authentication*: En esta sección se configura el soporte de autenticación. A continuación se muestra un ejemplo de su utilización con una pequeña descripción de cada elemento:

```
<authentication mode="Forms">  
  <forms cookieless="AutoDetect"  
    loginUrl="~/accessDenied.aspx"  
    name="TESAURO_AUTH"/>  
</authentication>
```

- La propiedad *mode* especifica el tipo de autenticación utilizado por la aplicación. En este caso se utiliza *Forms* que indica que la autenticación se hará por medio de *ASP.NET*.
 - La propiedad *cookieless* especifica si se usarán *cookies* y su comportamiento. En este caso se utiliza *AutoDetect* que indica que la *cookies* son utilizadas únicamente si es posible.
 - La propiedad *loginUrl* indica la página que utilizada para ingresar al sistema en el caso de que un usuario no esté autenticado.
 - La propiedad *name* especifica el nombre de la *cookie* utilizada para la autenticación.
- Sección *sessionState*: En esta sección se configura el estado de la sesión para la aplicación. A continuación se muestra un ejemplo de su utilización con una pequeña descripción de cada elemento:

```
<sessionState cookieless="AutoDetect"/>
```

- La propiedad *cookieless* especifica si se usarán *cookies* y su comportamiento. En este caso se utiliza *AutoDetect* que indica que la *cookies* son utilizadas únicamente si es posible.
- Sección *machineKey*: En esta sección se configuran las claves utilizadas para encriptación y descifrado de las *cookies* utilizadas en la aplicación. A continuación se muestra un ejemplo de su utilización con una pequeña descripción de cada elemento:

```
<machineKey validationKey="2859D0DD0B56A9C9BA4DF3EC1B8466515B8E70AF01  
DF78068775B8BA43B74282FFAC88F72A33B34718962C  
BF46C22C2BB1707404A8D44FAF7D3B7F1AF6949CBC"  
decryptionKey="0B6C6260C59720D47B5947AE9E9E9A9DA49912DC50  
D1BF3DF"  
validation="SHA1" />
```

- La propiedad *validationKey* especifica la clave utilizada para validar los datos cifrados.
- La propiedad *decryptionKey* especifica la clave utilizada para cifrar y descifrar los datos.
- La propiedad *validation* especifica el tipo de encriptación utilizado para validar los datos.

Si necesita ayuda para crear las claves utilice la siguiente herramienta: <http://www.aspnetresources.com/tools/keycreator.aspx>.

- Sección *pages*: Define ajustes de configuración globales para todas las páginas del sitio. A continuación se muestra un ejemplo de su utilización con una pequeña descripción de cada elemento:

```
<pages theme="Default" maintainScrollPositionOnPostBack="true"/>
```

- La propiedad *theme* especifica el tema utilizado de forma predeterminada.
 - La propiedad *maintainScrollPositionOnPostBack* especifica si la posición de la página se mantiene entre llamadas al servidor.
- Sección *httpRuntime*: Configura los ajustes utilizados durante la ejecución del sitio, los ajustes definidos en esta sección afecta de forma directa en la carga de archivos hacia el servidor. A continuación se muestra un ejemplo de su utilización con una pequeña descripción de cada elemento:

```
<httpRuntime executionTimeout="110" maxRequestLength="4096" />
```

- La propiedad *executionTimeout* especifica el número máximo de segundos que una petición puede ser ejecutada sin que sea detenida de forma automática. Esta propiedad define el tiempo máximo que un archivo puede tardarse en ser cargado al servidor.
- La propiedad *maxRequestLength* especifica el tamaño máximo en *kilo bytes (KB)* que una petición puede tener. Esta propiedad define el tamaño máximo que un archivo puede tener para ser cargado en servidor.

- Sección *customErrors*: Esta sección se utiliza para sustituir los errores predeterminados del servidor. A continuación se muestra un ejemplo de su utilización con una pequeña descripción de cada elemento:

```
<customErrors mode="RemoteOnly" defaultRedirect="~/error.aspx">
  <error statusCode="404" redirect="~/error.aspx?code=404"/>
  <error statusCode="408" redirect="~/error.aspx?code=408"/>
  <error statusCode="505" redirect="~/error.aspx?code=505"/>
</customErrors>
```

- La propiedad *mode* especifica si los errores personalizados están habilitados, deshabilitados o son mostrados únicamente a los usuarios remotos.
- La propiedad *defaultRedirect* especifica la página que se mostrará en caso de que un error ocurra.
- El elemento *error* especifica una página personalizada para un código de estado *HTTP* determinado, a continuación una descripción de sus propiedades:
 - *statusCode*: Especifica el código de estado *HTTP*.
 - *redirect*: La dirección de la página que es mostrada.
- Sección *mailSettings*: Esta sección se utiliza para configurar el envío de correos electrónicos desde la aplicación *Web*. A continuación se muestra un ejemplo de su utilización con una pequeña descripción de cada elemento:

```
<mailSettings>
  <smtp deliveryMethod="network" from="administrador@usac.edu.gt">
    <network host="mail.usac.edu.gt " port="25" password="" userName=""/>
  </smtp>
</mailSettings>
```

- El elemento *smtp* configura el método de entrega y la dirección desde la que se envían los correos electrónicos, a continuación una descripción de sus propiedades:
 - *deliveryMethod*: Especifica método utilizado para la entrega de correo electrónico. En este caso se utiliza *network* que indica que se utilizará un servidor *SMTP* (transporte de correo simple externo).
 - *from*: Especifica la dirección de correo electrónico desde la cual es enviado el correo saliente.
- El elemento *network* configura las opciones para utilizar el servidor *SMTP*, a continuación una descripción de sus propiedades:
 - *host*: Especifica la dirección del servidor de correo externo.
 - *port*: Especifica el puerto utilizado para enviar correo electrónico.
 - *password*: Especifica la contraseña para poder enviar correo electrónico, esta propiedad es opcional.

- *userName*: Especifica el nombre de usuario necesario para poder enviar correo electrónico, esta propiedad es opcional.

4.2.3.5.2. Archivo *web.sitemap*

Es un archivo en *XML* que define la estructura de navegación de un sitio *Web*. A continuación se muestra un ejemplo de su utilización con una pequeña descripción de cada elemento:

```
<siteMapNode url="~/default.aspx" title="Inicio" description="">
  <siteMapNode url="~/admin/default.aspx" title="Administración" description="" >
    <siteMapNode url="~/admin/users.aspx" title=" Administrar Usuarios" description="">
      <siteMapNode url="~/admin/createUser.aspx" title="Crear Usuarios" description=" " />
      <siteMapNode url="~/admin/roles.aspx" title="Administrar Roles" description="" />
    </siteMapNode>
    <siteMapNode url="~/admin/advisers.aspx" title="Administrar Asesores" description="">
      <siteMapNode url="~/admin/createAdviser.aspx" title="Crear Asesores" description="" />
    </siteMapNode>
  </siteMapNode>
</siteMapNode>
```

Debe existir al menos un elemento *siteMapNode*, y siempre debe existir una raíz única en la que todos los elementos están incluidos. La raíz puede contener todos los elementos *siteMapNode* que sean necesarios y a la vez estos pueden contener más elementos; se pueden anidar elementos *siteMapNode* a cualquier profundidad.

A continuación se describen las propiedades contenidas en el elemento *siteMapNode*:

- La propiedad *url* indica la ruta virtual que corresponden a una página en la aplicación. También puede contener rutas a páginas en otras aplicaciones o apuntar sitios Web completamente diferentes.
- La propiedad *title* se utiliza para desplegar el nombre asociado a una página de la aplicación.
- La propiedad *description* se utiliza para desplegar información en forma flotante que describe el elemento seleccionado.

En la aplicación se utiliza este archivo para definir la estructura que tiene el menú, cualquier cambio que se quiera hacer en el menú se debe efectuar en el archivo *web.sitemap*.

4.3. Manual de instalación

A continuación se listan los requisitos mínimos para la instalación de la aplicación; de igual forma se describen los pasos para poder instalarla, se puede realizar de forma manual o por medio de un asistente de instalación. Por convención se utilizará “C:\” para la letra de unidad del disco duro y “D:\” para la letra de unidad del lector de *DVD’s* o *CD’s*.

4.3.1. Requisitos mínimos

Los requisitos listados a continuación son recomendados para la correcta ejecución de la aplicación en el ambiente final de distribución:

- Hardware:
 - Procesador *Pentium* III de 1.0 *GHz* o compatible.

- 512 MB de RAM
 - Disco duro de 80 GB.
 - Unidad de lectora de DVD's o CD's.
 - Conexión a Internet de al menos 512 Kbps.
- Software:
 - Microsoft Windows 2000 Server Service Pack 3
 - Microsoft SQL Server 2000 Service Pack 3a
 - Microsoft .NET Framework 2.0
 - Adobe Acrobat Reader

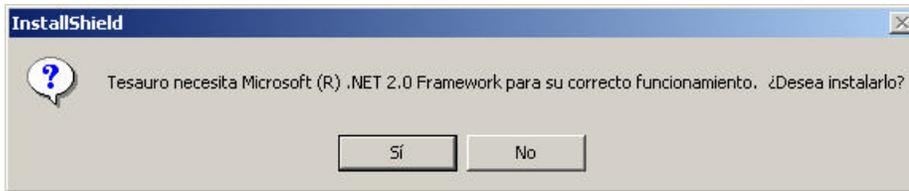
4.3.2. Instalación con asistente

El uso de este asistente de instalación esta optimizado para su uso en el sistema operativo *Microsoft Windows 2000* o *Microsoft Windows Server 2003*, y el manejador de bases de datos *Microsoft SQL Server 2000* o *2005*, si desea realizar una instalación con diferentes condiciones a las mencionadas o desea cambiar los parámetros predeterminados realice una instalación manual, para más información vea el apartado: 4.3.3 Instalación manual.

Al insertar el *CD* se ejecuta automáticamente el asistente de instalación, si no es así, debe iniciarlo de forma manual dando doble clic en el archivo *setup.exe* que se encuentra ubicado en el *CD*.

Al iniciar el asistente comprueba de forma automática si los requisitos necesarios están instalados en el sistema, sino es así se muestra una pantalla (ver figura 87) indicando el componente faltante con la opción de instalarlo.

Figura 87. **Instalación – prerequisite**



Luego de comprobar si el sistema cumple con los requisitos se muestra una pantalla de bienvenida (ver figura 88), en esta pantalla únicamente es necesario dar clic en el botón *Siguiente* para continuar o en el botón *Cancelar* para salir de la instalación.

Figura 88. **Instalación – bienvenida**



La siguiente pantalla (ver figura 89) solicita los datos para realizar la conexión al manejador de base de datos (este asistente es únicamente compatible con *SQL Server*), debe ingresar el nombre del servidor de base de datos y debe seleccionar la forma en que se autenticará, si selecciona autenticación a través del servidor debe indicar el usuario y contraseña. Debe dar clic en el botón *Siguiente* para continuar con la instalación.

Figura 89. Instalación – servidor de base de datos



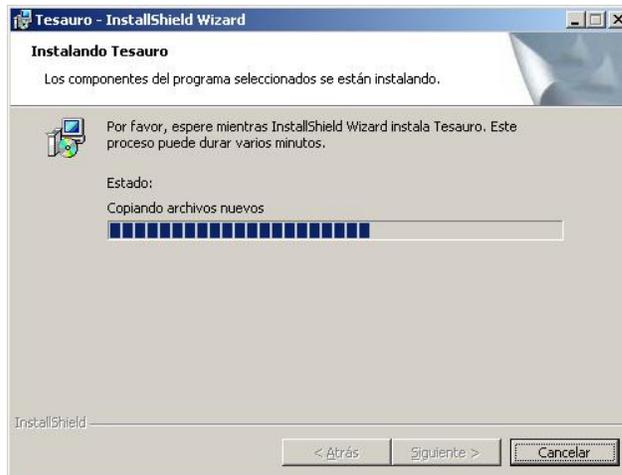
Si los datos proporcionados para la conexión al servidor de base de datos son correctos se muestra una pantalla (ver figura 90) indicando que el asistente está preparado para comenzar la instalación. Para continuar debe dar un clic en el botón *Instalar*.

Figura 90. Instalación – preparado para instalar



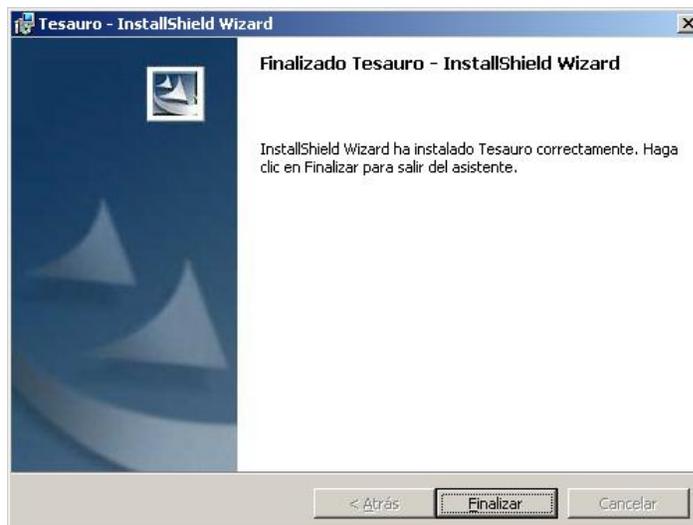
A continuación se muestra una pantalla (ver figura 91) en la que se puede observar el progreso de la instalación, si desea detener la instalación debe dar clic en el botón *Cancelar*.

Figura 91. **Instalación – progreso**



Si no ocurre ningún error durante la instalación se mostrará una pantalla (ver figura 92) indicando que la aplicación ha sido instalada correctamente. Para cerrar el asistente de un clic en el botón *Finalizar*.

Figura 92. **Instalación – finalizada**

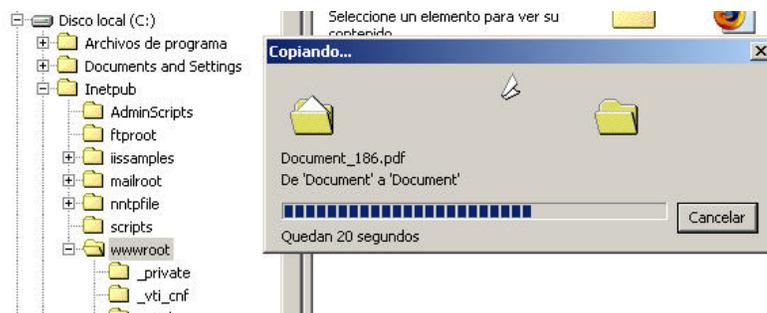


Luego de que ha finalizado el asistente de instalación vea el apartado: 4.3.4 Configuración de la aplicación, para realizar los últimos pasos necesarios de configuración y poder ejecutar la aplicación.

4.3.3. Instalación manual

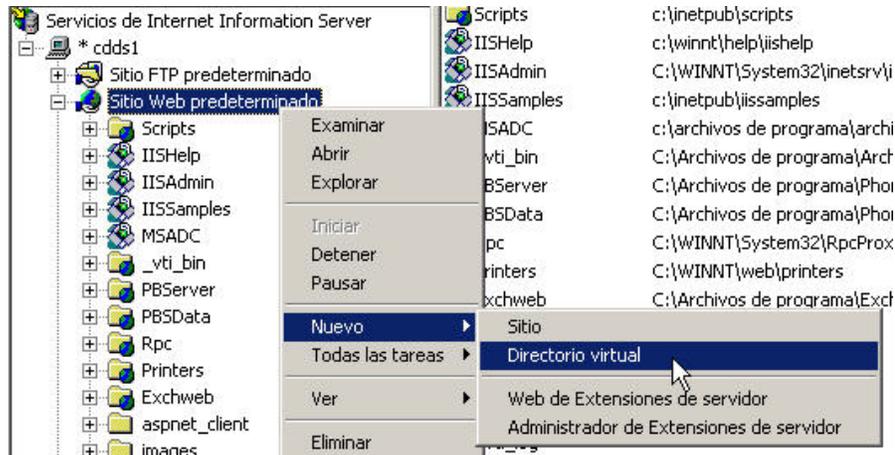
Si por alguna razón no es posible utilizar el asistente o bien desea llevar a cabo la instalación de forma manual siga los pasos que se describen a continuación. El primer paso es copiar el directorio *TESAURO* que se encuentra en la ubicación *D:\ISRRootFolder* del disco de instalación a la ubicación *C:\inetpub\wwwroot* que es de forma predeterminada el directorio utilizado para almacenar aplicaciones *Web*, en la figura 93 se muestra el proceso de copiado.

Figura 93. Copiar directorio de la aplicación



Luego debe abrir la herramienta “Servicios de *Internet Information Server*”, que se encuentra en “Herramientas administrativas” dentro de “Panel de control”. Aquí debe crear un directorio virtual, para hacerlo de un clic derecho en “Sitio *Web* predeterminado” y luego seleccione la opción “Nuevo” y finalmente de un clic en “Directorio Virtual”. En la figura 94 puede observar cómo crear un directorio virtual.

Figura 94. Crear directorio virtual



Después se iniciará el asistente para crear un directorio virtual, en la primera pantalla del asistente únicamente es necesario dar un clic en el botón *Siguiente* para continuar con la creación.

A continuación se debe indicar el alias o nombre del directorio virtual, este alias se utiliza para acceder a la aplicación desde el explorador *Web*; el nombre que se debe utilizar es: TESAURO, de esta forma para iniciar la aplicación se debe escribir `http://<nombre del servidor>/tesauro` en el explorador *Web*.

Luego de ingresar el alias debe dar un clic en el botón *Siguiente* para continuar con la creación del directorio virtual. En la figura 95 se muestra la pantalla de creación del alias.

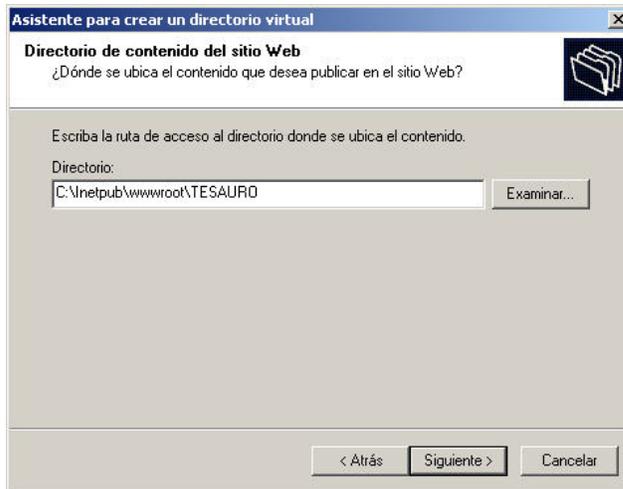
Figura 95. Alias del directorio virtual



En el siguiente paso del asistente se debe indicar donde se encuentran los archivos de la aplicación, estos archivos son los que fueron copiados del disco de instalación al disco duro, si se utilizó el directorio predeterminado la dirección será: `C:\inetpub\wwwroot\TESAURO`,

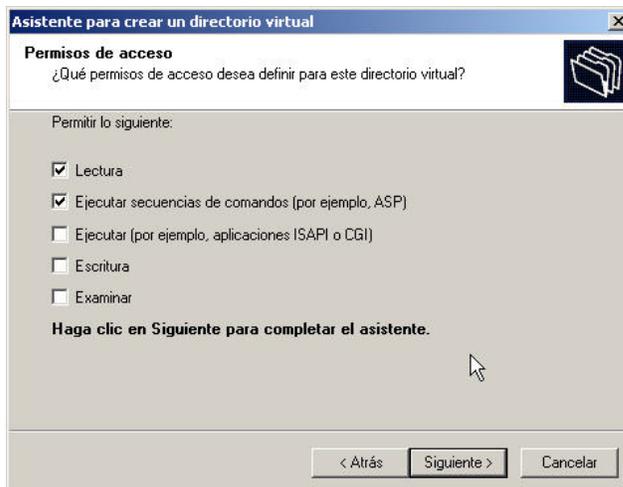
Luego de ingresar el directorio debe dar un clic en el botón *Siguiete* para continuar con la creación del directorio virtual. En la figura 96 se muestra la pantalla del directorio de contenido.

Figura 96. Directorio de contenido del sitio *Web*



A continuación deberá indicar los permisos del directorio virtual (ver figura 97), únicamente es necesario seleccionar las opciones *Lectura* y *Ejecutar secuencias de comandos* (por ejemplo, ASP). Para completar el asistente de un clic en el botón *Siguiete*.

Figura 97. Permisos de acceso

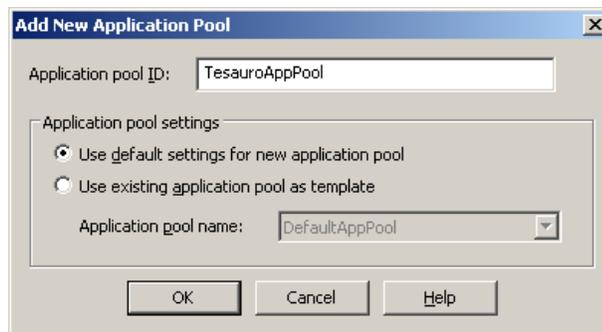


La última pantalla del asistente indica que el asistente ha finalizado, para continuar de un clic en el botón *Finalizar*.

El siguiente paso únicamente aplica a una instalación en *Microsoft Windows Server 2003*. Se debe crear un *Application Pool*, para esto es necesario dar un clic derecho en la carpeta *Application Pools* de la herramienta “Servicios de *Internet Information Server*”, luego seleccionar *New*, y finalmente dar un clic en *Application Pool...*

A continuación se muestra una pantalla (ver figura 98) donde se ingresa el nombre del *Application Pool*, el nombre a utilizar es: *TesauroAppPool*. Cuando haya ingresado el nombre de un clic en el botón *OK*.

Figura 98. **Crear un *Application Pool***

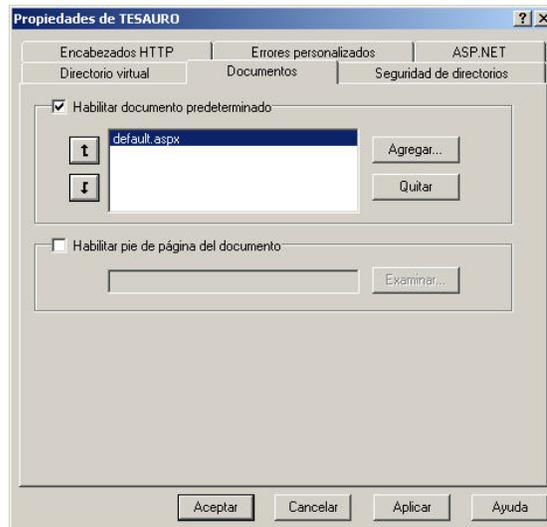


El siguiente paso es configurar las propiedades del directorio virtual, para esto debemos dar un clic derecho sobre el nombre del directorio, en este caso *TESAURO*, y luego seleccionar la opción *Propiedades*.

A continuación se mostrarán un conjunto de fichas con varias opciones de configuración, solo es necesario modificar la ficha de *Documentos*, *ASP.NET* y Directorio Virtual (esta última solo en el caso de *Microsoft Windows Server 2003*).

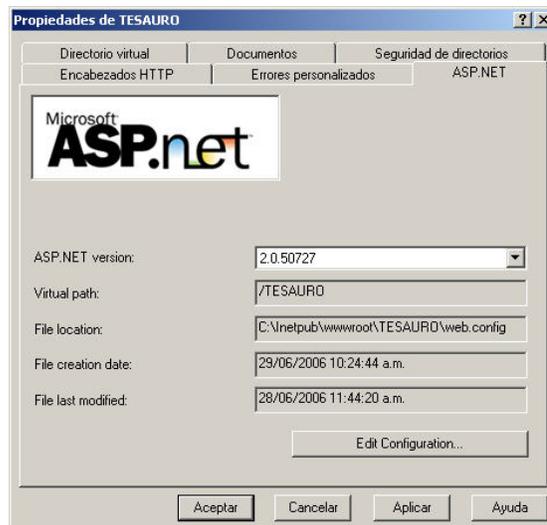
En la ficha de *Documentos* (ver figura 99) se debe seleccionar la opción “*Habilitar documento predeterminado*”, luego se quitan todos los documentos a excepción de *default.aspx*.

Figura 99. **Ficha documentos**



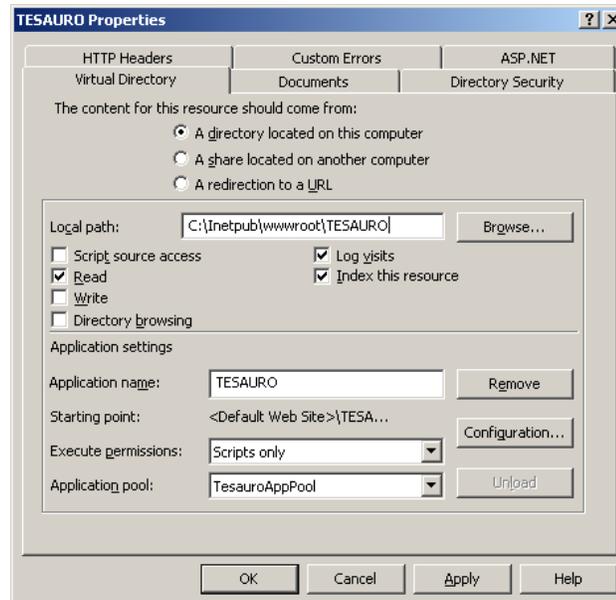
En la ficha *ASP.NET* (ver figura 100) debe seleccionarse la versión 2.0.50727.

Figura 100. **Ficha ASP.NET**



En la ficha *Directorio Virtual* (ver figura 101) debe seleccionarse el *Application Pool* que se creó anteriormente.

Figura 101. **Ficha directorio virtual**



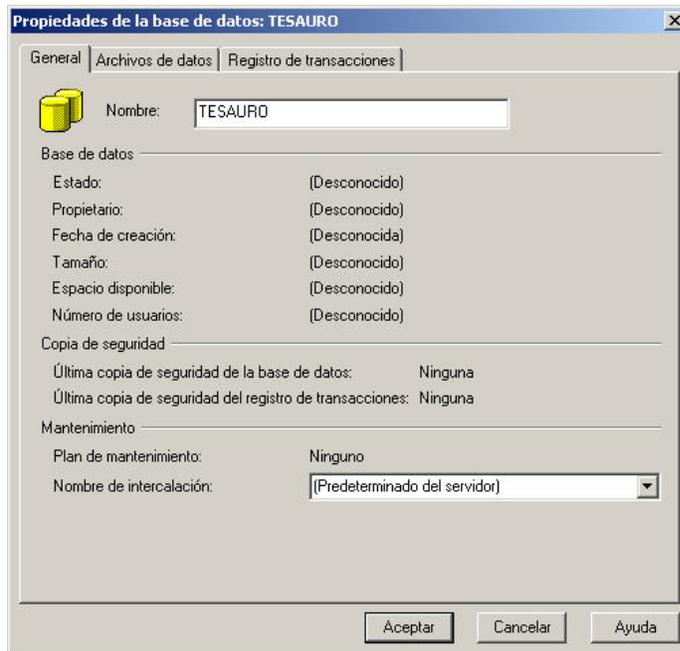
Al terminar de configurar todas la propiedades del directorio virtual debe dar un clic en el botón *Aceptar* y habrá finalizado la creación del directorio virtual.

A continuación se debe crear la base de datos, para esta tarea se utiliza el *Administrador corporativo* de *Microsoft SQL Server*, este se puede encontrar en la sección *Programas* del menú de inicio de *Microsoft Windows*.

En el *Administrador corporativo* debe seleccionar el servidor de bases de datos y luego dar un clic derecho sobre la carpeta *Bases de datos* y luego seleccionar la opción *Nueva base de datos...*

En la pantalla de creación (ver figura 102) únicamente debe ingresar el nombre del base de datos, en este caso debe ser: *TESAURO* y luego dar un clic en el botón *Aceptar* para crearla.

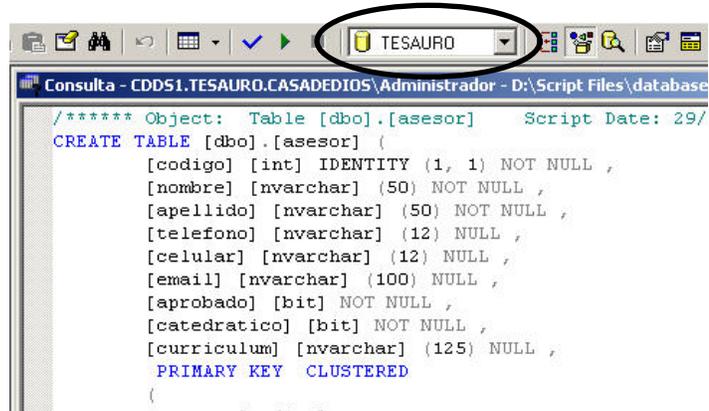
Figura 102. **Crear base de datos**



El último paso de la instalación manual es crear las tablas, procedimientos almacenados y los datos predeterminados en la base de datos, para esto se deben utilizar los *scripts* que se encuentran en el directorio "*D:\Código fuente\Scripts*" del disco de instalación.

Debe utilizar la herramienta "*Analizador de consultas*" que se encuentra en la misma ubicación que el *Administrador corporativo* para ejecutarlos. Debe tener cuidado de seleccionar la base de datos TESAURO como se muestra en la figura 103. Los scripts a ejecutar son: *DatabaseScript.sql* y *UserScript.sql*.

Figura 103. Carga de *scripts*



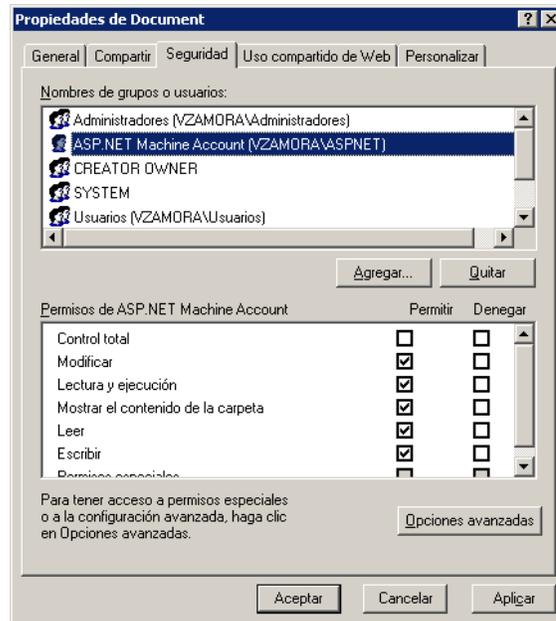
Luego de finalizar la instalación manual vea el apartado siguiente, para realizar los últimos pasos necesarios de configuración y poder ejecutar la aplicación.

4.3.4. Configuración de la aplicación

Luego de finalizar la instalación debe modificar el archivo *web.config* que se encuentra el directorio en el que fue instalada la aplicación, el directorio predeterminado es: *C:\inetpub\wwwroot\TESAURO*. Las secciones a modificar son *connectionStrings* y *mailSettings*, información sobre cómo realizar las modificaciones se encuentran en el apartado: 4.2.3.5.1 Archivo *web.config*.

También se debe agregar la cuenta de usuario *aspnet* (*network service* en el caso de *Windows Server 2003*) a los directorios *Curriculum* y *Document* que se encuentran en el directorio donde fue instalada la aplicación y se deben permitir los siguientes permisos: modificar, lectura y ejecución, mostrar el contenido de la carpeta, leer y escribir. Puede ver un ejemplo en la figura 104.

Figura 104. Permisos de carpeta



Para iniciar la aplicación debe escribir en un explorador *Web* la siguiente dirección: *http://<nombre del servidor>/tesauro*, se mostrará la pantalla de inicio de la aplicación y podrá comenzar a utilizarla; para información de cómo utilizar la aplicación consulte el apartado: 4.1 Manual de usuario.

Si desea ingresar a la sección de administración de la aplicación debe utilizar los siguientes datos:

- Usuario: administrador
- Contraseña: @dm!n

Se recomienda que cambie la contraseña al iniciar sesión por primera vez para que nadie pueda utilizar la cuenta de administración, para esto consulte el apartado: 4.1.2.2 Cambiar mi contraseña.

CONCLUSIONES

1. La forma actual de almacenamiento de información en cuanto a los trabajos de graduación de la Escuela de Ciencias y Sistemas, de la Facultad de Ingeniería, carece de organización, lo que dificulta la búsqueda y obtención de los datos deseados.
2. Para optimizar la utilización de recursos, es necesaria la implementación de un sistema de control, que permita conocer los proyectos terminados y los protocolos aprobados para los trabajos de graduación de la Escuela de Ciencias y Sistemas, de la Facultad de Ingeniería.
3. El uso de la arquitectura de n capas proporciona la flexibilidad y posibilidad de modificar las capas de forma independiente, y a la vez con transparencia para el usuario final, por lo que se usa en esta aplicación.
4. Al ser una aplicación *Web* es posible el acceso de múltiples usuarios en diferentes lugares, a través de un equipo con servicio de *Internet*.
5. La aplicación tiene la capacidad de almacenar información concerniente a trabajos de graduación, terminados o en proceso, cada uno con detalles de estudiante, asesor, tutor y estado, asimismo puede hacer filtros para la obtención de datos específicos.
6. La aplicación proporciona una interfaz práctica y sencilla tanto para el usuario como para el administrador.

RECOMENDACIONES

1. Asignar un responsable de la carga y actualización de datos, para contar siempre con una aplicación funcional y efectiva.
2. Se deberá cambiar de versión a la aplicación para aprovechar las características y estándares de nuevas tecnologías y esté siempre vigente.
3. Debe existir acceso restringido al servidor en el cual se encuentra la aplicación para evitar el robo, alteración o uso indebido de la misma.
4. Es necesario que el servidor que contenga la aplicación cuente con un *firewall*, antivirus y sea actualizado constantemente con los últimos parches de seguridad para mantener la integridad del sistema operativo y la aplicación.
5. Hacer del conocimiento público la existencia de la aplicación y su apropiada utilización.

BIBLIOGRAFÍA

1. Bell, Donald. "UML basics: An introduction to the Unified Modeling Language" **The Rational Edge**. <http://www.rational.com>. 2003.
2. Bell, Donald. "UML basics: The activity diagram" **The Rational Edge**. <http://www.rational.com>. 2003.
3. Bell, Donald. "UML's sequence diagram" **The Rational Edge**. <http://www.rational.com>. 2004.
4. Date, C. J. **Introducción a los sistemas de bases de datos**. 7ª. Ed. México: Pearson Educación, 2001. 960pp.
5. Senn, James A. **Análisis y diseño de sistemas de información**. 2ª. Ed. México: Editorial McGraw Hill, 1992. 225pp.
6. Spencer, Ken *et. al.* **OOP: Building reusable components with Microsoft Visual Basic .NET**. Estados Unidos: Microsoft Press, 2003. 491pp.
7. Bellinaso, Marco. **ASP.NET 2.0 Website programming: problem – design – solution**. Estados Unidos: Wiley Publishing, 2006. 576pp.