

CREACIÓN DE UNA HERRAMIENTA PARA GESTIÓN Y CONSULTA DE NORMAS DE
SEGURIDAD DE LA INFORMACIÓN APLICABLES A COMPAÑÍAS DE DESARROLLO
DE SOFTWARE FINANCIERO COLOMBIANO

DARÍO HERNANDO ACOSTA AMAYA
JUAN NICOLÁS AYALA RODRÍGUEZ

UNIVERSIDAD PILOTO DE COLOMBIA
FACULTAD DE INGENIERÍA
DIRECCIÓN DE POSTGRADOS
ESPECIALIZACIÓN DE SEGURIDAD INFORMÁTICA
BOGOTÁ D.C.
2014

CREACIÓN DE UNA HERRAMIENTA PARA GESTIÓN Y CONSULTA DE NORMAS DE
SEGURIDAD DE LA INFORMACIÓN APLICABLES A COMPAÑÍAS DE DESARROLLO
DE SOFTWARE FINANCIERO COLOMBIANO

DARÍO HERNANDO ACOSTA AMAYA
JUAN NICOLÁS AYALA RODRÍGUEZ

PROYECTO DE GRADO PARA OPTAR AL TÍTULO DE ESPECIALISTA EN
SEGURIDAD INFORMÁTICA

DIRECTOR

ALVARO ESCOBAR ESCOBAR

UNIVERSIDAD PILOTO DE COLOMBIA
FACULTAD DE INGENIERÍA
DIRECCIÓN DE POSTGRADOS
ESPECIALIZACIÓN DE SEGURIDAD INFORMÁTICA
BOGOTÁ D.C.

2014

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Bogotá D.C. 24 de Abril de 2014

CONTENIDO

	pág.
0. INTRODUCCIÓN	26
1. JUSTIFICACIÓN	27
2. PLANTEAMIENTO DEL PROBLEMA	28
2.1. PRESENTACIÓN.....	28
2.2. DESCRIPCIÓN DEL PROBLEMA	28
2.3. FORMULACIÓN DEL PROBLEMA.....	29
2.4. ALCANCE	29
2.5. LÍMITES	29
3. OBJETIVOS.....	30
3.1. OBJETIVO GENERAL	30
3.2. OBJETIVOS ESPECÍFICOS.....	30
4. MARCO TEÓRICO	31
4.1. ESTADO ACTUAL	31
4.2. MARCO CONCEPTUAL	31
4.2.1. Norma ISO 27000.....	31
4.2.2. Norma ISO 27001.....	32
4.2.3. Norma ISO 27002.....	32
4.2.4. SGSI (Sistema de gestión de seguridad de la información).....	32
4.2.5. Ley 1273 de 2009	33
4.2.6. Ley 1266 de 2008	33
4.2.7. Circular Externa 052 de 2007.....	33
4.2.8. Circular Externa 042 de 2012.....	34
4.2.9. Circular Externa 007 de 1996.....	34
4.2.10. Circular Externa 022 de 2010.....	35
4.3. MARCO TECNOLÓGICO	35
4.3.1. C# (C Sharp).....	35

4.3.2.	Microsoft SQL Server Express.....	36
4.3.3.	NHibernate.....	36
4.3.4.	Pentaho Reporting.....	37
4.3.5.	Subversion.....	38
4.3.6.	TortoiseSVN.....	38
5.	DISEÑO METODOLÓGICO.....	39
5.1.	PLANIFICACIÓN DE SISTEMAS DE INFORMACIÓN (PSI).....	39
5.1.1.	Inicio del plan de sistemas de información.....	39
5.1.1.1.	Análisis de la necesidad.....	39
5.1.1.2.	Identificación del alcance.....	39
5.1.1.3.	Determinación de responsables.....	40
5.1.2.	Definición y organización.....	40
5.1.2.1.	Especificación del ámbito y alcance.....	40
5.1.2.2.	Organización.....	41
5.1.3.	Estudio de la información relevante.....	42
5.1.3.1.	Selección y análisis de antecedentes.....	42
5.1.3.2.	Valoración de antecedentes.....	43
5.1.4.	Identificación de requisitos.....	43
5.1.4.1.	Estudio de los procesos.....	43
5.1.4.2.	Análisis de las necesidades de información.....	45
5.1.5.	Diseño del modelo de sistemas de información.....	47
5.1.5.1.	Diagnóstico de la situación actual.....	47
5.1.5.2.	Definición del modelo de sistemas de información.....	49
5.1.6.	Definición de la arquitectura tecnológica.....	50
5.1.6.1.	Selección de la arquitectura tecnológica.....	50
5.1.7.	Definición del plan de acción.....	52
5.2.	DESARROLLO DE SISTEMAS DE INFORMACIÓN.....	53
5.2.1.	Estudio de viabilidad del sistema (EVS).....	53
5.2.1.1.	Establecimiento del alcance del sistema.....	54

5.2.1.2.	Estudio de la situación Actual..	58
5.2.1.3.	Definición de requisitos del sistema.	58
5.2.1.4.	Estudio de alternativas de solución	61
5.2.1.5.	Valoración de las alternativas	63
5.2.1.6.	Selección de la Solución	69
5.2.2.	Análisis del sistema de información (ASI).	73
5.2.2.1.	Definición del sistema.	73
5.2.2.2.	Establecimiento de requisitos.	82
5.2.2.3.	Identificación de subsistemas de análisis	100
5.2.2.4.	Análisis de los casos de uso	105
5.2.2.1.	Descripción de la Interacción de Objetos.	108
5.2.2.2.	Análisis de clases.	110
5.2.2.3.	Definición de interfaces de usuario.	120
5.2.2.4.	Análisis de consistencia y especificación de requisitos.	126
5.2.2.5.	Especificación del plan de pruebas.	138
5.2.2.6.	Aprobación del análisis del sistema de información	151
5.2.3.	Diseño del sistema de información (DSI)	151
5.2.3.1.	Definición de la arquitectura del sistema.	151
5.2.3.2.	Diseño de la arquitectura de soporte	162
5.2.3.3.	Diseño de casos de uso reales.	164
5.2.3.4.	Diseño de clases.	172
5.2.3.5.	Diseño físico de datos.	173
5.2.3.6.	Verificación y aceptación de la arquitectura del sistema	180
5.2.3.7.	Generación de especificaciones de construcción	183
5.2.3.8.	Especificación técnica del plan de pruebas	188
5.2.3.9.	Aprobación del diseño del sistema de información	197
5.3.	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI).	197
5.3.1.1.	Preparación del entorno de generación y construcción	197
5.3.1.2.	Generación del código de los componentes y procedimientos.	199

5.3.1.3.	Ejecución de las pruebas unitarias.....	200
5.3.1.4.	Ejecución de las pruebas del sistema.	201
5.3.1.5.	Elaboración de los manuales de usuario.	203
6.	DESARROLLO, RESULTADOS Y APORTES	204
7.	CONCLUSIONES	206
8.	BIBLIOGRAFÍA	208
9.	ANEXOS	211

LISTA DE TABLAS

	pág.
Tabla 1. Retorno de la inversión	66

LISTA DE FIGURAS

	pág.
Figura 1. Diagrama de actividades	45
Figura 2. Diagrama de clases inicial	46
Figura 3. Modelo entidad relación inicial.....	46
Figura 4. Modelo del sistema de información.	50
Figura 5. Diagrama de representación.....	52
Figura 6. Estructura organizativa	55
Figura 7. Planificación de las alternativas.....	69
Figura 8. Modelo de descomposición de subsistemas.....	70
Figura 9. Gestión de Usuarios	84
Figura 10. Generación de reportes	85
Figura 11. Gestión de plan de Acción.....	85
Figura 12. Gestión de Normas.....	86
Figura 13. Gestión de Controles y comentarios	86
Figura 14. Interfaz inicial propuesta para la aplicación.	100
Figura 15. Diagrama de flujo de datos – Subsistemas de análisis	102
Figura 16. Diagrama de paquetes - Interfaces entre subsistemas.....	103
Figura 17. Diagrama de paquetes – Integración de subsistemas.....	104
Figura 18. Identificación de clases – caso de uso Gestión de normas	105

Figura 19. Identificación de clases – caso de uso Gestión de usuarios.....	106
Figura 20. Identificación de clases – caso de uso Gestión Reportes.....	106
Figura 21. Identificación de clases – caso de uso Gestión Plan de Acción	107
Figura 22. Identificación de clases – caso de uso Gestión Controles, Comentarios y Observaciones	107
Figura 23. Diagrama de interacción - caso de uso Gestión de Normas.....	108
Figura 24. Diagrama de interacción - caso de uso Gestión de usuarios.....	109
Figura 25. Diagrama de interacción - caso de uso Gestión Reportes.....	109
Figura 26. Diagrama de interacción - caso de uso Gestión Plan de Acción	110
Figura 27. Diagrama de interacción - caso de uso Gestión Controles, Comentarios y Observaciones	110
Figura 28. Asociaciones y agregaciones de clases	120
Figura 29. Interfaz generación de reportes	121
Figura 30. Interfaz plan de acción.....	122
Figura 31. Interfaz observación.....	122
Figura 32. Interfaz comentarios	123
Figura 33. Interfaz controles	123
Figura 34. Interfaz normas	124
Figura 35. Interfaz usuarios	124
Figura 36. Comportamiento dinámico de interfaz de consultor	125

Figura 37. Comportamiento dinámico de interfaz de auditor	125
Figura 38. Comportamiento dinámico de interfaz de administrador.....	126
Figura 39. Prototipo interfaz- Implementación de normas en el sistema.	129
Figura 40. Prototipo interfaz - Implementación de controles en el sistema.....	130
Figura 41. Prototipo interfaz - Porcentaje de cumplimiento establecido por analista de seguridad.	130
Figura 42. Prototipo interfaz - Porcentaje de implementación general de la norma.	131
Figura 43. Prototipo interfaz – Consulta de normas.....	131
Figura 44. Prototipo interfaz – licenciamiento de norma.....	132
Figura 45. Prototipo interfaz – Perfiles de acceso a la aplicación.....	132
Figura 46. Prototipo interfaz – Controles y comentarios en modo lectura.	133
Figura 47. Niveles de arquitectura.	152
Figura 48. Subsistemas de diseño de aplicación – ADComun	156
Figura 49. Subsistemas de diseño de aplicación – VSComun	157
Figura 50. Subsistemas de diseño de aplicación – ADNucleo.....	158
Figura 51. Especificación del Entorno Tecnológico – Comunicaciones.....	160

Figura 52. Patrón Singleton	162
Figura 53. Implementación Singleton.....	163
Figura 54. Patrón Fachada	164
Figura 55. Diagrama de clases asociadas a los casos de uso implementados ...	165
Figura 56. Interfaz de usuario – página de inicio de sesión.....	166
Figura 57. Interfaz de usuario – página principal.....	167
Figura 58. Interfaz de usuario – página Control.....	167
Figura 59. Interfaz de usuario – página Plan de Acción	168
Figura 60. Interfaz de usuario – página Comentarios.....	168
Figura 61. Interfaz de usuario – página Observación	169
Figura 62. Interfaz de usuario – página Usuarios	169
Figura 63. Interfaz de usuario – página Reporte	170
Figura 64. Modelo de navegación de interfaz de pantalla gráfica.....	171
Figura 65. Tabla base de datos – Clase Norma	173
Figura 66. Tabla base de datos – Clase Control	173

Figura 67. Tabla base de datos – Clase Comentario.....	174
Figura 68. Tabla base de datos – Clase PlanDeAccion.....	174
Figura 69. Tabla base de datos – Clase Rol.....	175
Figura 70. Tabla base de datos – Observacion	175
Figura 71. Tabla base de datos – Usuario.....	176
Figura 72. Diagrama entidad relación, distribución de datos.	180

LISTA DE ANEXOS

	pág.
Anexo 1. Lista de pruebas.	211
Anexo 2. Base de datos inicial de la aplicación.	227
Anexo 3. Código fuente de la aplicación.....	232
Anexo 4. Manuales técnicos de operación, seguridad y administración del sistema.	385
Anexo 5. Manual de usuario de la aplicación Information Security Status.....	412
Anexo 6. Código Patron Fachada.....	415

LISTA DE CUADROS

	pág.
Cuadro 1. Responsables	40
Cuadro 2. Participantes	41
Cuadro 3. Catálogo de requisitos.....	43
Cuadro 4. Modelo de los procesos de implementación de un SGSI	44
Cuadro 5. Comparación de requisitos vs sistemas de información.....	49
Cuadro 6. Catálogo de requisitos – Estudio de Viabilidad del Sistema	56
Cuadro 7. Plan de trabajo	57
Cuadro 8. Políticas de seguridad	59
Cuadro 9. Catálogo de requisitos.....	61
Cuadro 10. Preselección de alternativas de solución	62
Cuadro 11. Descripción de alternativas de solución	63
Cuadro 12. Estimación de costes	64
Cuadro 13. Estimación de beneficios tangibles	65
Cuadro 14. Estimación de beneficios intangibles.....	66
Cuadro 15. Identificación de las alternativas críticas	67
Cuadro 16. Inventario de alternativas críticas	68
Cuadro 17. Matriz Procesos / Localización Geográfica	71
Cuadro 18. Matriz Datos / Localización Geográfica	71
Cuadro 19. Elementos de la implantación	72

Cuadro 20. Requisito 1 – Implementar nuevas normas.....	73
Cuadro 21. Requisito 2 – Controlar las normas	74
Cuadro 22. Requisito 3 – Crear planes de acción sobre los controles	75
Cuadro 23. Requisito 4 – Medir implementación general	76
Cuadro 24. Requisito 5 – Medir implementación general	76
Cuadro 25. Requisito 6 – Consultar las normas.....	77
Cuadro 26. Modelo de negocio - Lienzo de modelo de negocios	79
Cuadro 27. Catálogo de normas	81
Cuadro 28. Catálogo de usuarios	82
Cuadro 29. Especificación de Casos de uso UC-001	87
Cuadro 30. Especificación de Casos de uso UC-002	89
Cuadro 31. Especificación de Casos de uso UC-003	91
Cuadro 32. Especificación de Casos de uso UC-004	93
Cuadro 33. Especificación de Casos de uso UC-005	95
Cuadro 34. Análisis de requisitos.....	97
Cuadro 35. Responsabilidades y atributos de la clase FormularioAgregarNormas	111
Cuadro 36. Responsabilidades y atributos de la clase FormularioActualizarNormas	111

Cuadro 37. Responsabilidades y atributos de la clase FormularioEliminarNormas	112
Cuadro 38. Responsabilidades y atributos de la clase FormularioConsultarNormas	112
Cuadro 39. Responsabilidades y atributos de la clase FormularioAgregarUsuario	113
Cuadro 40. Responsabilidades y atributos de la clase FormularioModificarUsuario	113
Cuadro 41. Responsabilidades y atributos de la clase FormularioEliminarUsuario	114
Cuadro 42. Responsabilidades y atributos de la clase FormularioGenerarReporte	114
Cuadro 43. Responsabilidades y atributos clase FormularioAgregarPlandeAccion	115
Cuadro 44. Responsabilidades y atributos clase FormularioModificarPlandeAccion	115
Cuadro 45. Responsabilidades y atributos clase FormularioEliminarPlandeAccion	116
Cuadro 46. Responsabilidades y atributos clase FormularioAgregarObservación	116
Cuadro 47. Responsabilidades y atributos de la clase FormularioVerObservación	117

Cuadro 48. Responsabilidades y atributos clase FormularioAgregarComentario	117
Cuadro 49. Responsabilidades y atributos clase FormularioVerComentario	118
Cuadro 50. Responsabilidades y atributos de la clase FormularioImplementarControl	118
Cuadro 51. Responsabilidades y atributos clase FormularioVerObservación	119
Cuadro 52. Verificación de los modelos definidos	127
Cuadro 53. Consistencia Modelo de clases – Diagramas dinámicos	128
Cuadro 54. Consistencia Modelo de clases – Interfaces de usuario	128
Cuadro 55. Especificación de requisitos de software	134
Cuadro 56. Criterios de severidad de las pruebas	138
Cuadro 57. Prueba funcional – Agregar Usuario	139
Cuadro 58. Prueba funcional – Modificar Usuario	139
Cuadro 59. Prueba funcional – Eliminar Usuario	140
Cuadro 60. Prueba funcional – Generar reporte	140
Cuadro 61. Prueba funcional – Agregar plan de acción	141
Cuadro 62. Prueba funcional – Modificar plan de acción	141

Cuadro 63. Prueba funcional – Eliminar plan de acción	142
Cuadro 64. Prueba funcional – Agregar Norma.....	142
Cuadro 65. Prueba funcional – Actualizar Norma.....	143
Cuadro 66. Prueba funcional – Eliminar Norma	143
Cuadro 67. Prueba funcional – Consultar Norma	144
Cuadro 68. Prueba funcional – Implementar Control.....	144
Cuadro 69. Prueba funcional – Actualizar Control.....	145
Cuadro 70. Prueba funcional – Ver Control.....	145
Cuadro 71. Prueba funcional – Agregar Comentario.....	146
Cuadro 72. Prueba funcional – Ver Comentario	146
Cuadro 73. Prueba funcional – Agregar Observación	147
Cuadro 74. Prueba funcional – Ver Observación	147
Cuadro 75. Requisitos del entorno de pruebas.....	148
Cuadro 76. Definición de pruebas unitarias	149

Cuadro 77. Definición pruebas del sistema.	149
Cuadro 78. Definición pruebas de integración	150
Cuadro 79. Definición pruebas de seguridad	150
Cuadro 80. Requisitos de diseño y construcción (Actualización).....	152
Cuadro 81. Catálogo de normas de diseño y construcción.....	154
Cuadro 82. Vista general de subsistemas de diseño de aplicación	155
Cuadro 83. Especificación del Entorno Tecnológico – Hardware y Software	159
Cuadro 84. Controles y elementos de diseño de la interfaz gráfica.	171
Cuadro 85. Descripción del camino de acceso a los datos.....	177
Cuadro 86. Matriz de especificación de acceso a los datos, relación Tablas – Casos.....	178
Cuadro 87. Análisis de Clases vs Casos de uso.....	182
Cuadro 88. Especificaciones del entorno de construcción.....	183
Cuadro 89. Relación de clases con los componentes asociados.	184
Cuadro 90. Base de datos Gestión, tabla Comentario.....	185
Cuadro 91. Base de datos Gestión, tabla Control.....	185
Cuadro 92. Base de datos Gestión, tabla Norma.....	186
Cuadro 93. Base de datos Gestión, tabla Observación	187
Cuadro 94. Base de datos Gestión, tabla PlanDeAccion.....	187
Cuadro 95. Base de datos Gestión, tabla Rol.....	187
Cuadro 96. Base de datos Gestión, tabla Usuario	188

Cuadro 97. Pruebas unitarias del sistema	189
Cuadro 98. Detalle prueba unitaria PU-1.....	189
Cuadro 99. Detalle prueba unitaria PU-2.....	190
Cuadro 100. Detalle prueba unitaria PU-3.....	191
Cuadro 101. Pruebas del sistema.....	192
Cuadro 102. Detalle prueba de sistema PS-1.....	193
Cuadro 103. Detalle prueba de sistema PS-2.....	193
Cuadro 104. Detalle prueba de sistema PS-3.....	194
Cuadro 105. Detalle prueba de sistema PS-4.....	195
Cuadro 106. Detalle prueba de sistema PS-5.....	195
Cuadro 107. Detalle prueba de sistema PS-6.....	196
Cuadro 108. Elementos de implementación de la base de datos física.....	198
Cuadro 109. Componentes del entorno de construcción.....	198
Cuadro 110. Código de componentes	199
Cuadro 111. Resultado de la realización de pruebas unitarias.....	201
Cuadro 112. Resultado de la realización de pruebas unitarias.....	202

GLOSARIO

C SHARP (C#): es un lenguaje de programación que se ha diseñado para generar diversas aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Las numerosas innovaciones de C# permiten desarrollar aplicaciones rápidamente y mantener la expresividad y elegancia de los lenguajes de estilo de C¹.

CSS: Hojas de Estilo en Cascada (Cascading Style Sheets) es un lenguaje de hojas de estilo utilizado para describir el aspecto y el formato de un documento escrito en un lenguaje de marcas, esto incluye varios lenguajes basados en XML como son XHTML o SVG².

CVS: Concurrent Versions System o simplemente CVS, también conocido como Concurrent Versioning System, es una aplicación informática que implementa un sistema de control de versiones: mantiene el registro de todo el trabajo y los cambios en los ficheros (código fuente principalmente) que forman un proyecto (de programa) y permite que distintos desarrolladores (potencialmente situados a gran distancia) colaboren. CVS se ha hecho popular en el mundo del software libre. Sus desarrolladores difunden el sistema bajo la licencia GPL³.

¹ Microsoft. Visual C#. En: Developer Network. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <<http://msdn.microsoft.com/es-es/library/kx37x362.aspx>>.

² Comercioelectronicoufv. Lenguaje CSS. En: Wiki. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <<http://comercioelectronicoufv.wikispaces.com/Lenguaje+CSS>>

³ Wikipedia. CVS. En: Wiki. [en línea]. No 1 (2014) [consultado 04 sep. 2014]. Disponible en <<http://es.wikipedia.org/wiki/ CVS> >

HQL: Hibernate utiliza un lenguaje de consulta potente (HQL) que se parece a SQL. Sin embargo, comparado con SQL, HQL es completamente orientado a objetos y comprende nociones como herencia, polimorfismo y asociación⁴.

HTML: siglas de HyperText Markup Language («lenguaje de marcas de hipertexto»), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML)⁵.

JAVASCRIPT: (abreviado comúnmente "JS") es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico⁶.

IEEE 830 DE 1998: especificación de requisitos de software (ERS), es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software⁷.

⁴ Hibernate. Capítulo 15. HQL: El lenguaje de consulta de Hibernate. En: Hibernate. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <<http://docs.jboss.org/hibernate/core/3.5/reference/es-ES/html/queryhql.html>>.

⁵ Wikipedia. HTML. En: Wikipedia. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <<http://es.wikipedia.org/wiki/HTML>>.

⁶ Wikipedia. JavaScript. En: Wikipedia. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <<http://es.wikipedia.org/wiki/JavaScript>>.

⁷ Wikipedia. Especificación de requisitos de software. En: Wikipedia. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <http://es.wikipedia.org/wiki/Especificaci%C3%B3n_de_requisitos_de_software>

METRICA V3: La metodología MÉTRICA Versión 3 ofrece a las Organizaciones un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software⁸.

SEGURIDAD DE LA INFORMACIÓN⁹: de acuerdo al concepto definido por la ISO 27001, la seguridad de la información busca preservar la confidencialidad, integridad y confidencialidad, así como otras propiedades como la autenticidad, responsabilidad, no-repudio y confiabilidad.

SGBD¹⁰: (Sistema de gestión de base de datos) o en inglés Database management system (DBMS), es una agrupación de programas que sirven para definir, construir y manipular una base de datos.

⁸ administracionelectronica. Métrica v.3. En: administracionelectronica. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html#.VAkaAWPvauJ>

⁹ ISO 27000.ES. Seguridad de la información. En: ISO 27000.ES. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <<http://www.iso27000.es>>

¹⁰ Universidad don Bosco. Base de datos II. En: <http://www.udb.edu.sv>. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <<http://www.udb.edu.sv/udb/archivo/guia/informatica-ingenieria/base-de-datos-ii/2014/ii/guia-1.pdf>>

RESUMEN

El presente documento pretende establecer los lineamientos necesarios para la creación de un software dirigido a las empresas colombianas de desarrollo de software financiero, que requieran la implementación de normas nacionales e internacionales de seguridad de la información, tomando como eje fundamental la confidencialidad, integridad y disponibilidad de la información. El sistema planteado pretende dar una solución al problema presentado al momento de implementar un sistema de gestión de seguridad de la información y los inconvenientes presentados en su medición mientras se realiza esta tarea.

0. INTRODUCCIÓN

La necesidad de asegurar de forma consciente la información en las empresas, ha llamado la atención no sólo de los dueños del negocio, sino también, de instituciones a nivel nacional e internacional que necesitan adaptarse a las nuevas amenazas que surgen cada día, en torno a la información y a la mitigación de las posibles causas que puedan afectar la disponibilidad, integridad y confidencialidad de la misma.

Tanto es así, que hoy en día instituciones gubernamentales de diferentes estados han definido como parte de sus prioridades, implementar mecanismos que permitan proteger la información de sus ciudadanos y más aún, de aquellas instituciones que son parte del funcionamiento del país como es el caso de las entidades financieras.

El caso a tratar en esta investigación, está dirigido a como las normas de seguridad de la información exigidas a las entidades financieras colombianas, y específicamente a las empresas que les prestan servicios tercerizados de desarrollo de software financiero, presentan algunos inconvenientes en la implementación de Sistemas de Gestión de Seguridad de la Información, a causa de no tener una herramienta que permita realizar esta tarea de forma centralizada, permitiendo conocer en cualquier momento su estado de implementación, a través de la generación de informes que permitan mejorar la toma de decisiones de la alta gerencia.

1. JUSTIFICACIÓN

En busca de asegurar el activo considerado más importante para todo tipo de organización, la protección de la información ha sido el punto de investigación de diferentes organizaciones que buscan definir los mecanismos necesarios para garantizar su confidencialidad, integridad y disponibilidad.

El desarrollo de esta investigación se apoya en la necesidad que tienen las empresas desarrolladoras de software financiero en Colombia, que por su naturaleza requieren manejar en muchos casos información de diferentes entidades financieras, y que de forma directa o indirecta, se ven obligadas a cumplir ciertos parámetros de seguridad de la información como los requeridos por la Superintendencia Financiera de Colombia.

Entre las diferentes exigencias promulgadas por esta entidad gubernamental, se encuentran algunos requerimientos mínimos definidos en la Circular Externa 22 de 2010, dirigidas a empresas "intermediarias" que no están sujetas actualmente a la vigilancia de la superintendencia financiera, pero que en desarrollo de sus actividades, tienen acceso a información confidencial de las entidades financieras y de sus clientes.

Además, el objetivo de esta investigación también está dirigido a las normas internacionales, que aunque muchas veces están relacionadas con las normas emitidas y exigidas por el estado, también ofrecen algunas ventajas al realizar su implementación, como es el caso de mayor competitividad, mejor reputación de las empresas, etc. Entre las normas mencionadas se encuentran el conjunto de estándares ISO 27000, enfocados a proteger la información, los cuales están basados principalmente en un sistema de gestión de seguridad de la información.

Con base en lo anterior, se tiene conciencia de que las empresas colombianas deben conocer las diferentes reglamentaciones emitidas que aplican a su negocio, dónde y cómo las implementan. Es así, que el desarrollo de un software para este fin ayudará en su implementación y toma de decisiones.

2. PLANTEAMIENTO DEL PROBLEMA

2.1. PRESENTACIÓN

En el mundo de hoy donde la información de la compañía es uno de los pilares de su funcionamiento, es necesario pensar de forma obligatoria en su aseguramiento, tal como se observa actualmente en algunos sectores económicos.

A raíz de esta obligación, las empresas buscan medir el esfuerzo dedicado a esta tarea buscando la forma más eficaz y eficiente de lograrlo, y es aquí donde se fundamenta el presente trabajo de investigación, dando como resultado una herramienta capaz de suplir la necesidad de llevar el control en la implementación de un Sistema de Gestión de la Seguridad de la Información (SGSI), siendo este una pieza fundamental en la tarea de salvaguardar el activo Información.

2.2. DESCRIPCIÓN DEL PROBLEMA

Actualmente muchas de las empresas que están obligadas, o que por iniciativa propia pretenden implementar un sistema de gestión de seguridad de la información o norma asociada, se ven llevadas muchas veces a registrar su gestión a través de herramientas poco eficientes y no orientadas a esta tarea como archivos sueltos, carpetas compartidas, etc., dando como resultado un escenario bastante complejo a la hora de medir su implementación, generando confusión en los interesados en el proyecto, y a su vez, no evidenciar los resultados esperados ante la alta gerencia.

Teniendo en cuenta lo anterior, este proyecto busca dar una solución centralizada a este problema, ofreciendo una herramienta de software que permite registrar los avances en la implementación de un Sistema de Gestión de Seguridad de la Información, permitiendo incluir la norma en que estará basado el proyecto planteado (v.gr ISO 27001, Circular 052, etc.), y de esta forma ser adaptable a las necesidades de la organización.

2.3. FORMULACIÓN DEL PROBLEMA

¿Cómo gestionar de forma eficiente y centralizada la implementación de normas enfocadas a la seguridad de la información de organizaciones colombianas dedicadas a prestar servicios tercerizados de desarrollo de software financiero?

2.4. ALCANCE

El sistema será creado para la empresa Colombiana ALFCOM S.A., dedicada a la creación de software financiero. Lo creación del proyecto ofrecerá una herramienta para la medición en la implementación de un sistema de gestión de seguridad información basado principalmente en los requerimientos establecidos en la Norma Circular 052, emitida por la superintendencia financiera de Colombia.

2.5. LÍMITES

La aplicación está dirigida a la empresa ALFCOM S.A., dedicada a la creación de software financiero en Colombia.

La implantación del sistema se hace localmente en un ambiente de pruebas, así como la gestión de usuarios y tareas asociadas, comprobando de esta manera su funcionamiento.

3. OBJETIVOS

3.1.OBJETIVO GENERAL

Crear una herramienta que permita la gestión y consulta de normas de seguridad de la información que aplican a empresas colombianas que prestar servicios tercerizados de desarrollo de software financiero.

3.2.OBJETIVOS ESPECÍFICOS

- Ofrecer una base de conocimiento de las normas que aplican a la gestión de seguridad de la información a la empresa colombiana ALFCOM S.A. de desarrollo de software financiero.
- Identificar las normas de seguridad de la información aplicables a organizaciones colombianas dedicadas al desarrollo de software financiero.
- Comparar el estado de la implementación de las normas que aplican a la organización ALFCOM S.A., para mantenerlas identificadas, documentadas y cuantificadas.
- Implementar un mecanismo de generación de informes detallados y gerenciales que reflejen el estado de implementación actual de las normas de seguridad de información requeridas por la organización.

4. MARCO TEÓRICO

4.1. ESTADO ACTUAL

La creación de herramientas para la gestión en la implementación de un sistema de gestión de seguridad de la información (SGSI), se viene manejando desde hace algunos años por medio de herramientas de software especializadas inicialmente en normas internacionales, entre las que se pueden resaltar están: SECURIA SGSI, GlobalSGSI y Inmuno Suite, las cuales son descritas en el Estudio de la Situación Actual de la sección 5. Diseño metodológico.

4.2. MARCO CONCEPTUAL

4.2.1. Norma ISO 27000. ISO/IEC 27000 es un conjunto de estándares desarrollados -o en fase de desarrollo- por ISO (International Organization for Standardization) e IEC (International Electrotechnical Commission), que proporcionan un marco de gestión de la seguridad de la información utilizable por cualquier tipo de organización, pública o privada, grande o pequeña¹¹.

¹¹ ISO 27000.ES. Seguridad de la información. En: ISO 27000.ES. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <<http://www.iso27000.es>>

4.2.2. Norma ISO 27001. Publicada el 15 de Octubre de 2005. Es la norma principal de la serie y contiene los requisitos del sistema de gestión de seguridad de la información. Tiene su origen en la BS 7799-2:2002 y es la norma con arreglo a la cual se certifican por auditores externos los SGSI de las organizaciones. Sustituye a la BS 7799-2, habiéndose establecido unas condiciones de transición para aquellas empresas certificadas en esta última. En su Anexo A, enumera en forma de resumen los objetivos de control y controles que desarrolla la ISO 27002:2005 (nueva numeración de ISO 17799:2005 desde el 1 de Julio de 2007), para que sean seleccionados por las organizaciones en el desarrollo de sus SGSI; a pesar de no ser obligatoria la implementación de todos los controles enumerados en dicho anexo, la organización deberá argumentar sólidamente la no aplicabilidad de los controles no implementados. Desde el 28 de Noviembre de 2007, esta norma está publicada en España como UNE-ISO/IEC 27001:2007 y puede adquirirse online en AENOR. Otros países donde también está publicada en español son, por ejemplo, Colombia, Venezuela y Argentina. El original en inglés y la traducción al francés pueden adquirirse en ISO.org.

4.2.3. Norma ISO 27002. Desde el 1 de Julio de 2007, es el nuevo nombre de ISO 17799:2005, manteniendo 2005 como año de edición. Es una guía de buenas prácticas que describe los objetivos de control y controles recomendables en cuanto a seguridad de la información. No es certificable. Contiene 39 objetivos de control y 133 controles, agrupados en 11 dominios. Como se ha mencionado en su apartado correspondiente, la norma ISO27001 contiene un anexo que resume los controles de ISO 27002:2005. En España, aún no está traducida (previsiblemente, a lo largo de 2008). Desde 2006, sí está traducida en Colombia (como ISO 17799) y, desde 2007, en Perú (como ISO 17799; descarga gratuita). El original en inglés y su traducción al francés pueden adquirirse en ISO.org.

4.2.4. SGSI (Sistema de gestión de seguridad de la información). Sistema de Gestión de Seguridad de la Información, abreviado como SGSI, es un enfoque que busca planear, hacer, verificar y actuar sobre la tarea de aseguramiento de la información.

De acuerdo a lo definido bajo el estándar ISO 27001, el Sistema de Gestión de Seguridad de la Información, es parte del sistema gerencial general, basado en un enfoque de riesgo comercial; para establecer, implementar, operar, monitorear, revisar, mantener y mejorar la seguridad de la información.

4.2.5. Ley 1273 de 2009. Por medio de la cual se modifica el código penal, se crea un nuevo bien jurídico tutelado - denominado "De la protección de la información y de los datos" - y se preservan integralmente los sistemas que utilicen las tecnologías de la información y las comunicaciones, entre otras disposiciones.

4.2.6. Ley 1266 de 2008. Por la cual se dictan las disposiciones generales del Habeas Data y se regula el manejo de la información contenida en bases de datos personales, en especial la financiera, crediticia, comercial, de servicios y la proveniente de terceros países y se dictan otras disposiciones

4.2.7. Circular Externa 052 de 2007. En Colombia se presenta el cumplimiento obligatorio emitido por la Superintendencia Financiera de Colombia (SFC) bajo su circular 052 del año 2007, enfocada a todas las entidades sometidas a la inspección y vigilancia de la Superintendencia Financiera de Colombia (SFC), con la que se busca garantizar la confidencialidad de la información atendiendo lo dispuesto en el artículo 15 de la Constitución Política de Colombia considerando confidencial para efectos de la aplicación toda aquella información amparada por la reserva bancaria V. gr. número de cuenta; número de identificación personal (PIN); número de tarjeta física; información sobre depósitos o inversiones de cualquier tipo, créditos, saldos, cupos y movimientos de cuenta, siempre que vayan acompañados del nombre o número de identificación del cliente.

La Superfinanciera obliga a las entidades vigiladas tener en cuenta en la contratación de outsourcing¹²:

- SLA (Acuerdo de niveles de servicio).
- Acuerdo de confidencialidad.
- Restricciones sobre software empleado.
- Normas de seguridad informática y física.

¹² COLOMBIA, SUPERINTENDENCIA FINANCIERA DE COLOMBIA. Circular Externa 052 de 2007 (25, octubre, 2007). Requerimientos mínimos de seguridad y calidad en el manejo de información a través de medios y canales de distribución de productos y servicios para clientes y usuarios. 2007. P 1-3.

- Procedimientos a seguir cuando se evidencia alteración o manipulación de equipos o información.
- Planes de contingencia y continuidad
- Identificación física de los funcionarios
- Cifrado fuerte para envío y recepción de información (3DES o AES)

4.2.8. Circular Externa 042 de 2012. Por medio de la cual se incorporan algunas modificaciones al Capítulo Décimo Segundo del Título Primero de la Circular Básica Jurídica, en materia de requerimientos mínimos de seguridad y calidad para la realización de operaciones de las entidades vigiladas por la superintendencia financiera¹³.

El objetivo de esta circular es otorgar herramientas más robustas para las operaciones que realizan clientes y usuarios del sector financiero por medio de medios electrónicos, lo cual obliga a las entidades vigiladas a cumplir unas exigencias mínimas con el fin de proteger y asegurar sus operaciones.

4.2.9. Circular Externa 007 de 1996, Circular Básica Jurídica. Esta circular reúne las diferentes instrucciones en materia jurídica emitidas por la Superintendencia Bancaria y que a la fecha se encuentran vigentes, donde se destacan los siguientes propósitos¹⁴:

Compendiar en un solo volumen las normas e instructivos vigentes. Con este propósito, su presentación se elaboró en hojas intercambiables y su consulta podrá efectuarse vía módem, con el fin de facilitar su difusión y consulta.

Disminuir la carga operativa de las entidades vigiladas en materia de conceptualización jurídica.

¹³ COLOMBIA, SUPERINTENDENCIA FINANCIERA DE COLOMBIA. Circular Externa 042 de 2012 (24, octubre, 2012). Por medio de la cual se incorporan algunas modificaciones al Capítulo Décimo Segundo del Título Primero de la Circular Básica Jurídica, en materia de requerimientos mínimos de seguridad y calidad para la realización de operaciones. 2012. P 1-3.

¹⁴ COLOMBIA, SUPERINTENDENCIA FINANCIERA DE COLOMBIA. Circular Externa 007 de 1996 (19, enero, 1996). Esta Circular reúne las diferentes instrucciones en materia jurídica emitidas por la Superintendencia Bancaria y que a la fecha se encuentran vigentes. 1996. P 1-3.

4.2.10. Circular Externa 022 de 2010. Teniendo en cuenta que las empresas “intermediarias” a que referidas a la comunicación de la presente comunicación no se encuentran actualmente sujetas a la vigilancia de esta Superintendencia, pero que las mismas, en desarrollo de su actividad, tienen acceso a información confidencial de las entidades financieras y de sus clientes; independientemente del esquema jurídico adoptado por las entidades vigiladas con terceros que intervengan en el proceso, las que contraten o interactúen con este tipo de “intermediarios” deben cumplir los requerimientos mínimos previstos en el numeral 3.2. de la referida Circular Externa 022 de 2010, en particular los numerales 3.2.2., literales b) acuerdos de confidencialidad sobre información manejada y sobre las actividades desarrolladas, c) propiedad de la información y g) Procedimientos y controles para la entrega de la información manejada y la destrucción de la misma por parte del tercero una vez finalizado el servicio, y 3.2.5., mecanismos de cifrado fuerte para el envío y recepción de información confidencial con los terceros contratados. Dichas exigencias deben incluirse en los acuerdos o definirse en desarrollo de los contratos que celebren las entidades vigiladas con terceros. Por ende, en todo caso, el responsable de administrar la información de sus clientes en las condiciones antes señaladas es el establecimiento de crédito emisor de la tarjeta¹⁵.

4.3. MARCO TECNOLÓGICO

4.3.1. C# (C Sharp). C# es un lenguaje de programación que se ha diseñado para generar diversas aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Las numerosas innovaciones de C# permiten desarrollar aplicaciones rápidamente y mantener la expresividad y elegancia de los lenguajes de estilo de C.

Visual C# es una implementación del lenguaje de C# de Microsoft. Visual Studio ofrece compatibilidad con Visual C# con un completo editor de código, un compilador, plantillas de proyecto, diseñadores, asistentes para código, un

¹⁵ SUPERINTENDENCIA FINANCIERA DE COLOMBIA [Online]. Superintendencia Financiera de Colombia, 2013 [citado 15 Noviembre de 2014]. Disponible en internet: <<http://www.superfinanciera.gov.co/Normativa/PrincipalesPublicaciones/boletinej/boletin3311/seguridadadycalidad.html>>.

depurador eficaz y de fácil uso y otras herramientas. La biblioteca de clases de .NET Framework ofrece acceso a numerosos servicios de sistema operativo y a otras clases útiles y adecuadamente diseñadas que aceleran el ciclo de desarrollo de manera significativa¹⁶.

4.3.2. Microsoft SQL Server Express. Microsoft® SQL Server™ es un sistema de administración y análisis de bases de datos relacionales de Microsoft para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos¹⁷.

Microsoft® SQL Server® Express es un sistema de administración de datos gratuito, eficaz y confiable que ofrece un almacén de datos completo y confiable para sitios web ligeros y aplicaciones de escritorio. Diseñada para una implementación sencilla y una creación de prototipos rápida¹⁸.

4.3.3. NHibernate. Es un Framework de código abierto para mapear objetos a tablas relacionales para .Net, usando HQL o SQL nativo. Hibernate utiliza un lenguaje de consulta potente (HQL) que se parece a SQL. Sin embargo, comparado con SQL, HQL es completamente orientado a objetos y comprende nociones como herencia, polimorfismo y asociación¹⁹.

Como todas las herramientas de su tipo, Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos en la

¹⁶ Microsoft. Visual C#. En: Developer Network. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <<http://msdn.microsoft.com/es-es/library/kx37x362.aspx>>.

¹⁷ Microsoft. SQL Server. En: technet. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <Microsoft SQL server, technet.microsoft.com/es-es/library/bb545450.aspx>

¹⁸ Microsoft. SQL Server 2012. En: Microsoft.com. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <<http://www.microsoft.com/es-ar/download/details.aspx?id=29062>>

¹⁹ Hibernate. Capítulo 15. HQL: El lenguaje de consulta de Hibernate. En: Hibernate. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <<http://docs.jboss.org/hibernate/core/3.5/reference/es-ES/html/queryhql.html>>.

base de datos operando sobre objetos, con todas las características de la Programación Orientada a Objetos. Hibernate convertirá los datos entre los tipos utilizados por Java (Para el caso del presente proyecto .Net) y los definidos por SQL. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución²⁰.

Esta herramienta pretende ser utilizada para manejar la información entre la aplicación desarrollada a través de Visual Studio 2012 y la base de datos Microsoft SQL Server Express 2012.

4.3.4. Pentaho Reporting. Es una suite open-source que permite crear reportes relacionales y analíticos de un amplio rango de fuentes de datos y salidas de datos como: PDF, Excel, HTML, Texto, RTF (Rich-Text-File), XML y CSV.

Pentaho se define a sí mismo como una plataforma de inteligencia de negocios “orientada a la solución” y “centrada en procesos” que incluye todos los principales componentes requeridos para implementar soluciones basados en procesos y ha sido concebido desde el principio para estar basada en procesos. Las soluciones que Pentaho pretende ofrecer se componen fundamentalmente de una infraestructura de herramientas de análisis e informes integrados con un motor de workflow de procesos de negocio. La plataforma será capaz de ejecutar las reglas de negocio necesarias, expresadas en forma de procesos y actividades y de presentar y entregar la información adecuada en el momento adecuado²¹.

²⁰ Hibernate. Capítulo 15. HQL: El lenguaje de consulta de Hibernate. En: Hibernate. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <<http://docs.jboss.org/hibernate/core/3.5/reference/es-ES/html/queryhql.html>>.

²¹ Wikipedia. Pentaho. En: wikipedia. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en < <https://es.wikipedia.org/wiki/Pentaho> >

4.3.5. Subversion. Programa servidor para el control de versiones de productos de software el cual reemplazó al CVS, se debe usar con un cliente, para usarlo en el visual studio se debe instalar el AnkhSVN y para el explorador de windows el TortoiseSVN.

Subversion (SVN) es una herramienta de control de versiones open source basada en un repositorio cuyo funcionamiento se asemeja enormemente al de un sistema de ficheros. Es software libre bajo una licencia de tipo Apache/BSD.

Utiliza el concepto de revisión para guardar los cambios producidos en el repositorio. Entre dos revisiones sólo guarda el conjunto de modificaciones (delta), optimizando así al máximo el uso de espacio en disco. SVN permite al usuario crear, copiar y borrar carpetas con la misma flexibilidad con la que lo haría si estuviese en su disco duro local. Dada su flexibilidad, es necesaria la aplicación de buenas prácticas para llevar a cabo una correcta gestión de las versiones del software generado.

Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintas computadoras. A cierto nivel, la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones²².

4.3.6. TortoiseSVN. Es un cliente “Subversion”, que funciona como una extensión del shell de Windows, liberado bajo la licencia GNU GPL permite el fácil uso de control de revisiones, control de versiones y control de código fuente de software. Entre sus características están: fácil uso, todos los protocolos de subversión están soportados, configuraciones por proyecto, disponible en varios lenguajes, etc²³.

²² Wikipedia. Subversion. En: wikipedia. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <https://es.wikipedia.org/wiki/Subversion_%28software%29 >

²³ Tortoise SVN. Subversion. En: Tortoise. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <<http://tortoisesvn.net/about.html>>.

5. DISEÑO METODOLÓGICO

5.1. PLANIFICACIÓN DE SISTEMAS DE INFORMACIÓN (PSI)

Esta etapa tiene como objetivo la obtención de un marco de referencia para el desarrollo de sistemas de información que responda a los objetivos estratégicos de la organización.

5.1.1. Inicio del plan de sistemas de información. A través de esta actividad se determina la necesidad del Plan de Sistemas de Información y se lleva a cabo el arranque formal del mismo, con el apoyo del nivel más alto de la organización.

5.1.1.1. Análisis de la necesidad. El sistema de información planteado pretende suplir la necesidad que tienen las compañías de desarrollo de software financiero, para dar cumplimiento a las normas exigidas por entidades como la Superintendencia Financiera de Colombia, bajo su circular 052, otorgando una herramienta que permita medir en todo momento el estado de implementación de un sistema de gestión de seguridad de la información.

5.1.1.2. Identificación del alcance

5.1.1.2.1 Descripción general. El sistema de información planteado está dirigido la empresa ALFCOM S.A. dedicada al desarrollo de software financiero.

5.1.1.2.2 Factores críticos de éxito

- Permite adaptarse a cualquier compañía implementando cualquier tipo de norma.
- Implementación controlada y centralizada de un sistema de gestión de seguridad de la información.

5.1.1.3. Determinación de responsables

5.1.1.3.1 Responsables. Los creadores del proyecto se encargaran de la planeación, realización, implementación y mantenimiento del sistema de información planteado como se evidencia en el *cuadro 1*.

Cuadro 1. Responsables

Responsable	Descripción
Juan Nicolás Ayala Rodríguez	Ingeniero de sistemas
Darío Hernando Acosta Amaya	Ingeniero Telemático

Fuente: Autor

5.1.2. Definición y organización

5.1.2.1. Especificación del ámbito y alcance. A través de esta tarea se describe el ámbito de los procesos de la organización a considerar. Así como también se define el alcance, a través de los objetivos específicos del Plan de Sistemas de Información. Procesos de la organización afectados:

- Implementación y control de un Sistema de Gestión de Seguridad de la Información.
- Cumplimiento de normas exigidas por el estado Colombiano para la empresa ALFCOM dedicadas al desarrollo de software financiero en Colombia.

5.1.2.1.1 Objetivos

- Medir de forma automática a través de una aplicación, el estado de implementación de normas como ISO27001 y CE-052 SFC (Superintendencia Financiera de Colombia).
- Garantizar el cumplimiento de normas exigidas por la ley Colombiana.

- Emisión de reportes gerenciales que apoyen la toma de decisiones de la alta gerencia respecto a la implementación de un sistema de gestión de seguridad de la información, basado en la norma ISO 27001.

5.1.2.2. Organización

5.1.2.2.1 Participantes

A continuación en el *cuadro 2* se detallan los participantes.

Cuadro 2. Participantes

Personal	Cantidad	Funciones
Ingeniero de Sistemas y telecomunicaciones	2	Planeación, desarrollo e implementación del sistema de información planteado
Tutor de tesis	1	Coordinar el desarrollo general del proyecto Soportar el avance del proyecto mediante recomendaciones y puntos de mejora en el desarrollo del proyecto.

Fuente: Autor

5.1.2.2.2 Lugares de reunión. Se determina como lugar de reunión la sede de postgrados de la Universidad Piloto de Colombia, así como reuniones periódicas online con servicios como Google Drive, Skype, etc. para el desarrollo continuo del proyecto.

5.1.2.2.3 Estándares de documentación utilizados. El desarrollo del presente sistema de información estará basado en el estándar de documentación Métrica V3, descrita como una metodología de planificación, desarrollo y mantenimiento de sistemas de información, promovida por el Ministerio de Hacienda y Administraciones Públicas (antiguo Ministerio de Administraciones Públicas) del Gobierno de España para la sistematización de actividades del ciclo de vida de los proyectos software en el ámbito de las administraciones públicas²⁴.

5.1.3. Estudio de la información relevante

5.1.3.1. Selección y análisis de antecedentes

5.1.3.1.1 Análisis de antecedentes. La creación de herramientas para la gestión en la implementación de un sistema de gestión de seguridad de la información (SGSI), se viene manejando desde hace algunos años por medio de herramientas de software especializadas tanto gratuitas como de pago, enfocadas inicialmente en normas internacionales, que buscan suplir la necesidades de las empresas con la necesidad de implementar un SGSI, y más aún después de la emisión de la norma ISO 27001 del año 2005 publicada por la Organización Internacional de Normalización (ISO).

Del mismo modo, con base en lo expuesto en el marco conceptual, las empresas “intermediarias” no se encuentran actualmente sujetas a la vigilancia de la Superintendencia Financiera, pero por el hecho de tener acceso a información confidencial de las entidades financieras y de sus clientes; deben cumplir los requerimientos mínimos previstos en el numeral 3.2. de la referida Circular Externa 022 de 2010; para el presente proyecto se pretende enfatizar en el requerimiento asociado a la definición de “Procedimientos y controles para la entrega de la información manejada y la destrucción de la misma por parte del tercero una vez finalizado el servicio”. De esta manera la herramienta propuesta permitirá

²⁴ Wikipedia. Métrica. En: wikipedia. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <<https://es.wikipedia.org/wiki/M%C3%89TRICA>>

funcionar como soporte ante los acuerdos definidos en desarrollo de los contratos que celebren las entidades vigiladas con terceros.

5.1.3.2. Valoración de antecedentes

5.1.3.2.1 Catálogo de requisitos. A continuación en el *cuadro 3*, se determina el catálogo de requisitos inicial para la realización del proyecto, estableciendo los términos de referencia en función de los antecedentes descritos anteriormente.

Cuadro 3. Catálogo de requisitos

No.	Requisito
1	El sistema estará orientado a la implementación de controles de diferentes normas de seguridad de la información.
2	El sistema evaluará de forma general el estado de implementación del SGSI teniendo en cuenta el porcentaje establecido por la persona encargada de esta tarea.
3	El sistema reportará en todo momento el estado de implementación a través de un valor (0-100%) en la esquina superior derecha de la interfaz gráfica.
4.	El sistema permitirá consultar las diferentes normas implementadas a modo consulta
5.	El sistema permitirá agregar la licencia o comprobante de la norma comprada legalmente (Si aplica).
6.	El sistema permitirá acceder bajo 3 perfiles: auditor, consultor, y administrador.
7	Bajo ninguno de los perfiles se permitirá modificar los comentarios y observaciones realizados a los controles (Garantiza integridad)

Fuente: Autor

5.1.4. Identificación de requisitos

5.1.4.1. Estudio de los procesos. El objetivo final de esta actividad va a ser la especificación de los requisitos de información de la organización, así como obtener un modelo de información que los complementa.

5.1.4.1.1 Modelo de procesos de la organización. Se estudia el proceso de la organización ALFCOM S.A. implicado en el ámbito de la solución propuesta, identificando las actividades y/o funciones, información implicada y unidades organizativas que participan en el desarrollo.

A continuación en el *cuadro 4*, se describe una visión global objetiva del proceso de la organización identificado.

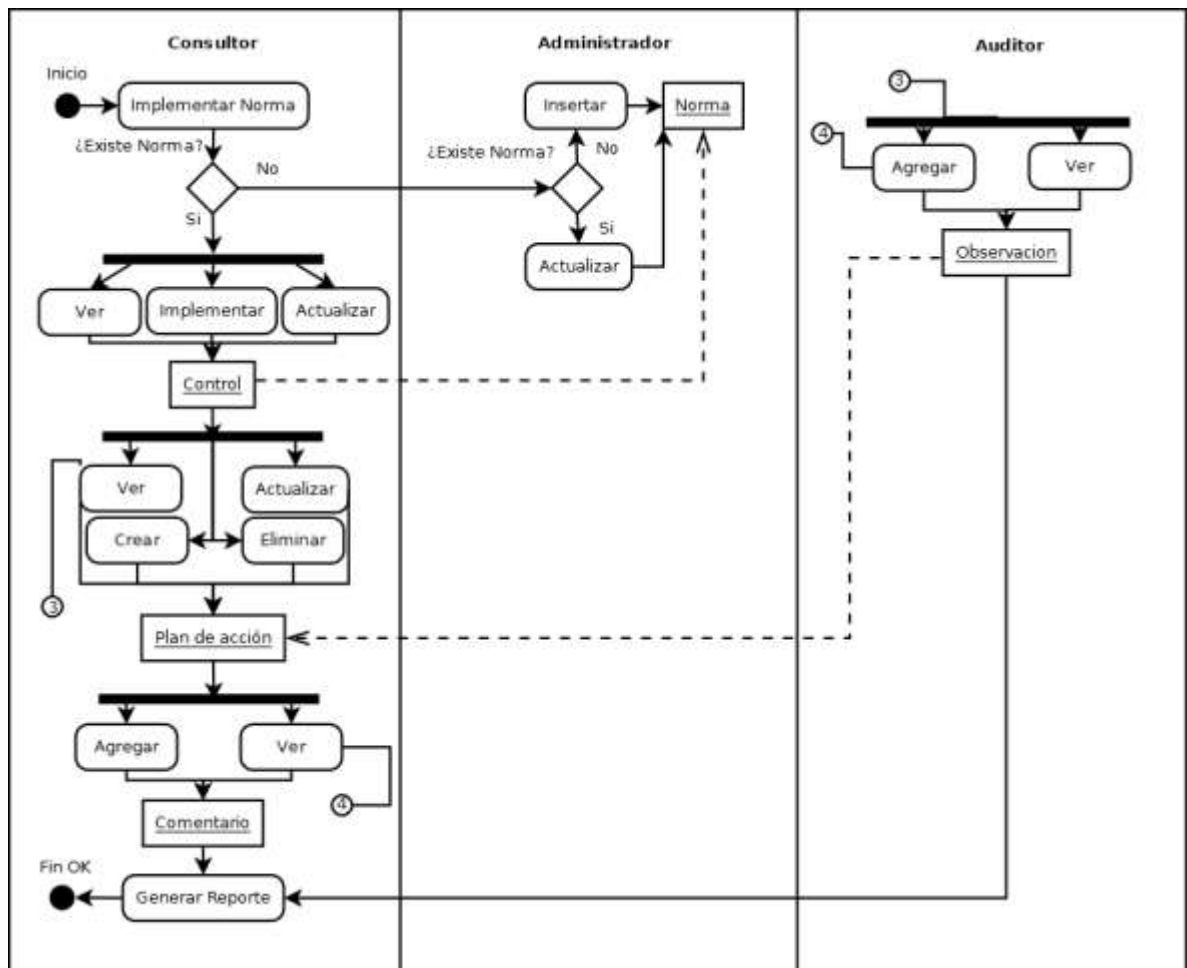
Cuadro 4. Modelo de los procesos de implementación de un SGSI

Nombre del proceso: Implementación de un SGSI		
LÍDER: Oficial de seguridad	OBJETO: Implementar el sistema de gestión de seguridad de la información	ALCANCE: Proceso de desarrollo de software financiero
ENTRADAS: Ley, Normas, Requerimientos, controles.	ACTIVIDADES DE SEGUIMIENTO Y CONTROL: Consulta de leyes y normas emitidas por el estado, normas internacionales.	SALIDAS: Implementación del sistema de gestión de seguridad de la información basado en normas internacionales y nacionales.
PROVEEDORES: ISO 27001, Superintendencia Financiera de Colombia, Estado Colombiano.	INDICADOR: Porcentaje de implementación de la norma a utilizar (Pueden ser varias).	CLIENTES: Auditores internos, externos, oficial de seguridad, alta gerencia.
REQUISITOS: Norma ISO 27001:2015 - 2013, Requerimientos Circulares emitidos por la SFC (v.gr. Circular 052, 042, etc).	DOCUMENTOS: Procedimiento de implementación de SGSI.	REGISTROS: Comentarios, observaciones y plan de acción definidos por cada control a través de la herramienta de implementación del SGSI.

Fuente: Autor

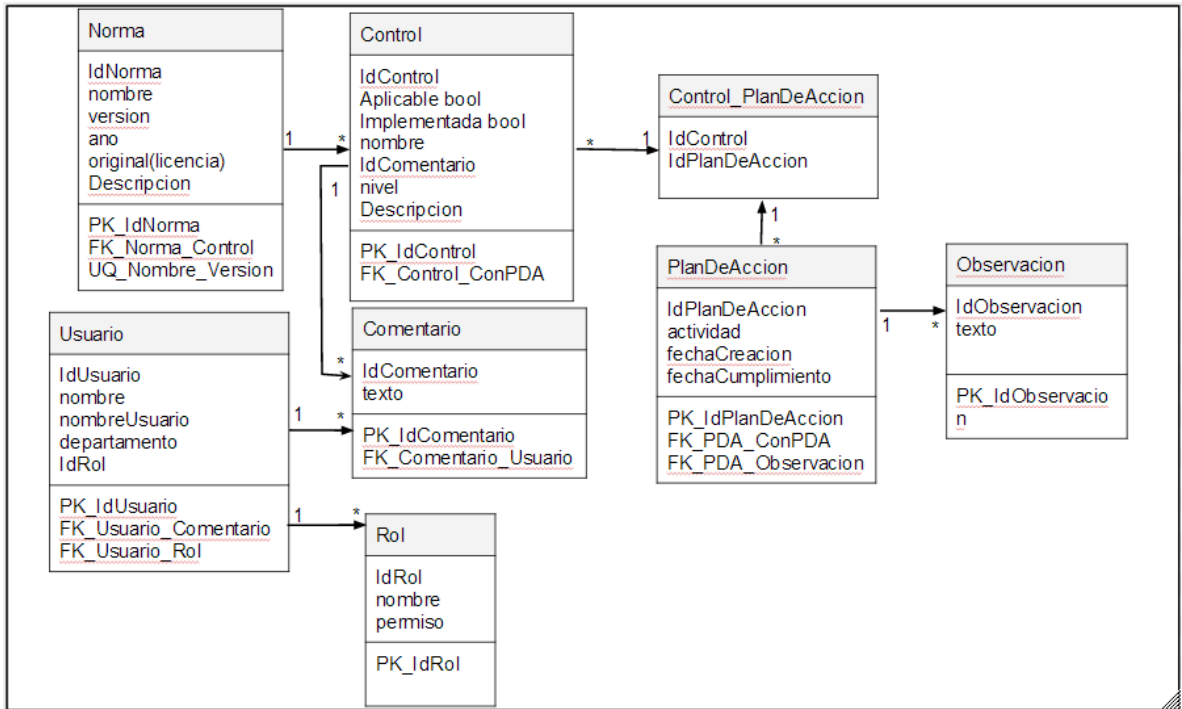
5.1.4.2. Análisis de las necesidades de información. Se identifican las necesidades de información del proceso analizado anteriormente, elaborando un modelo de información que refleja las principales entidades y relaciones que existente entre ellas, a través de una perspectiva de lo que debe ser el proceso en cuanto a sus actividades y funciones, así como a la información de entrada y salida para cada una de ellas. Esta actividad es descrita a través del diagrama de actividades, *figura 1*, diagrama de clases, *figura 2* y diagrama entidad relación descrito a continuación en la *figura 3*, los cuales cubren los aspectos principales de los casos de uso definido más adelante.

Figura 1. Diagrama de actividades



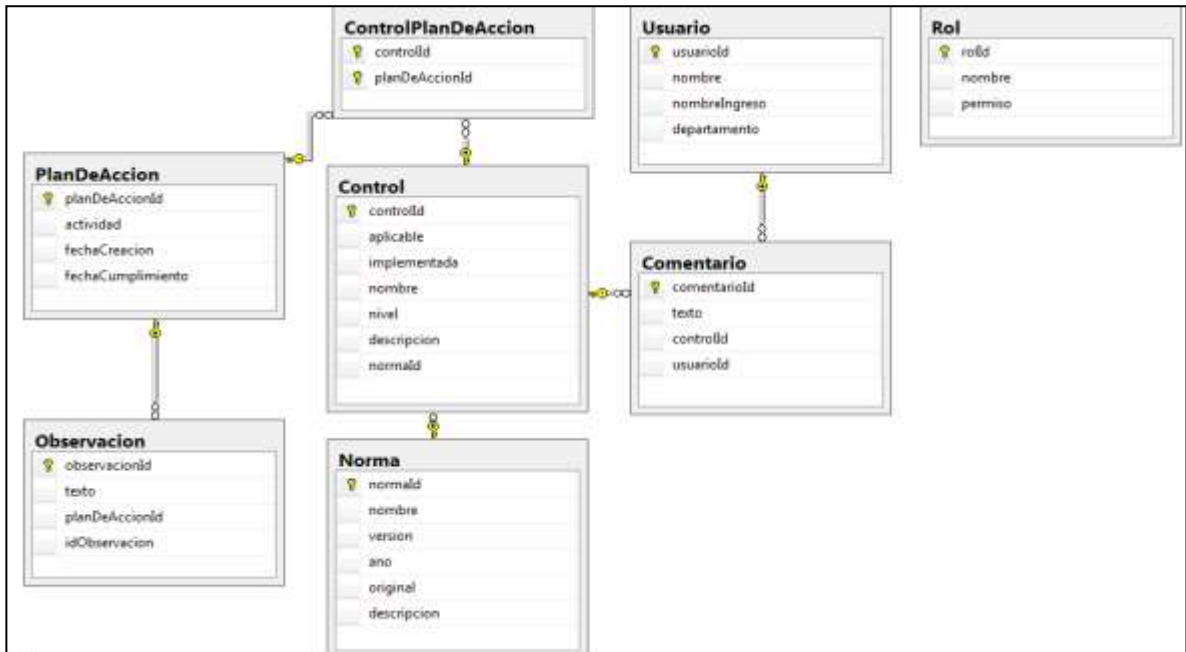
Fuente: Autor

Figura 2. Diagrama de clases inicial



Fuente: Autor

Figura 3. Modelo entidad relación inicial



Fuente: Autor

5.1.5. Diseño del modelo de sistemas de información

5.1.5.1. Diagnóstico de la situación actual. A continuación se describen los sistemas de información actuales, relacionados con la implementación de Sistemas de Gestión de Seguridad de la Información. También se describen los requisitos que se cubren con ellos, determinando los requisitos del catálogo no cubiertos por las herramientas identificadas, *cuadro 5*.

5.1.5.1.1 Sistemas de información Actuales

- **Securia Sgsi.** Es una herramienta integral que cubre el proceso automático de implantación, puesta en funcionamiento, mantenimiento y mejora continua de un Sistema de Gestión de Seguridad de la Información (SGSI) según la norma internacional ISO 27001.
- Se enfoca en la construcción de módulos funcionales orientados a dar cumplimiento a requerimientos y controles establecidos en la norma ISO 27001, entre los módulos actualmente ofrecidos se encuentran: Gestión de Incidencias y No Conformidades, Mejora Continua, Gestión Documental, Análisis y Gestión de Riesgos.
- Funciona bajo licencia GNU (General Public License) Version 2 de Junio de 1991.
- **GlobalSGSI.** Cubre con el ciclo completo de las normas de SGSI, desde las fases de inicio y planificación del proyecto hasta el mantenimiento y mejora continua, pasando por el análisis de riesgos, el cuadro de mandos, cumplimiento de todos los requerimientos exigidos por la norma, análisis diferencial contra la norma ISO 27001, análisis de cumplimiento de los 133 controles del anexo A de la norma, permite realizar un inventario de activos, posee un catálogo de amenazas y vulnerabilidades, propone controles para gestionar los riesgos, seguimiento de la implantación de los controles, establece la declaración de aplicabilidad asociada al alcance de la implementación, permite gestionar no conformidades, acciones correctivas y preventivas, generar informes personalizados de cada uno de sus apartados, etc.

Ofrece la integración de otros productos como OSSIM, CMDB, Ticketing a través de módulos opcionales.

- Inmuno Suite. Es un conjunto de soluciones de seguridad de la información que permite gestionar de forma integral un Sistema de Gestión de Seguridad de la Información – SGSI de conformidad con la norma ISO/IEC 27001. Inmuno Suite está compuesta por dos soluciones las cuales son, Inmuno SGSI e Inmuno Risk.

Inmuno SGSI permite definir desde los objetivos de seguridad, la política, indicadores de gestión, el seguimiento a la implementación de los controles, los programas de capacitación, resultados de auditorías internas, gestión de incidentes, control de documentos gestión de registros, revisión por parte de la dirección, hasta gestión de acciones correctivas y preventivas, etc.

Inmuno Risk comprende las cuatro etapas fundamentales de la gestión de riesgos: Inventario de Activos con énfasis en la valoración de los activos, Análisis de Riesgos (ISO 27001, 27005) con énfasis en la determinación de la probabilidad de ocurrencia del riesgo, Evaluación del Riesgo con énfasis en la determinación de los niveles de impacto en caso de que el riesgo ocurra, y Tratamiento del Riesgo con énfasis en la propuesta de controles a fin de reducir el nivel de riesgo efectivo.

Sin embargo considerando el amplio alcance de las aplicaciones citadas, estas herramientas no se enfocan de forma directa en la implementación de las normas emitidas por el gobierno colombiano dirigidas especialmente a entidades financieras.

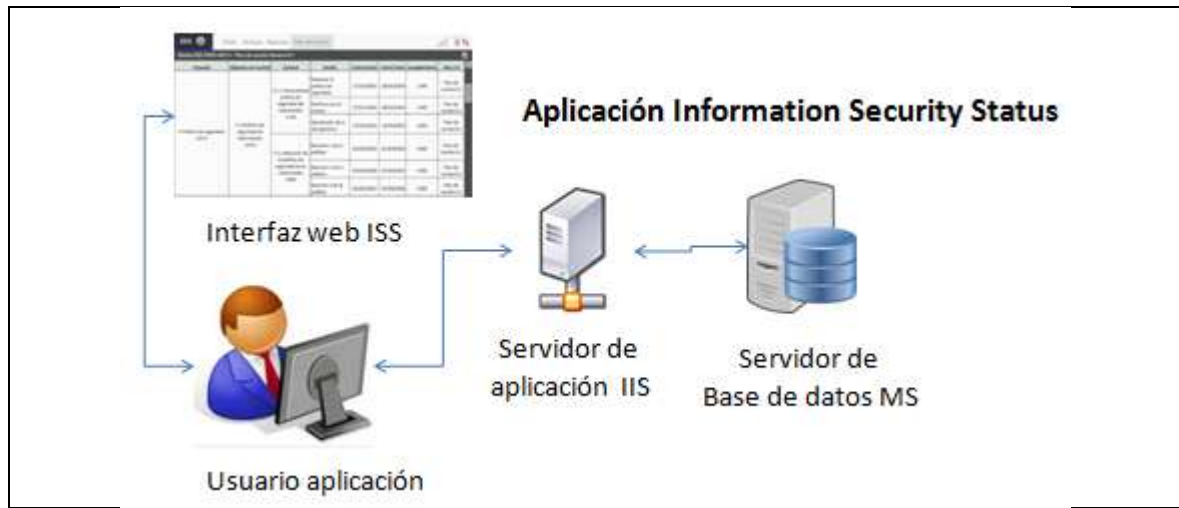
Cuadro 5. Comparación de requisitos vs sistemas de información

No	Requisito	Securia SGSI	Global SGSI	Inmuno Suite	Presente proyecto (ISS)
1	El sistema está orientado a la implementación de controles de diferentes normas de seguridad de la información				X
2	El sistema evalúa de forma general el estado de implementación del SGSI teniendo en cuenta el porcentaje establecido por la persona encargada de esta tarea.	X	X	X	X
3	El sistema reporta en todo momento el estado de implementación.	X	X	X	X
4.	El sistema permite consultar las diferentes normas implementadas a modo consulta	X	X	X	X
5.	El sistema permite agregar la licencia o comprobante de la norma comprada legalmente (Si aplica).				X
6.	El sistema permite acceder bajo 3 perfiles: auditor, consultor, y administrador.				X
7	Bajo ninguno de los perfiles se permite modificar los comentarios y observaciones realizados a los controles (Garantiza integridad)	X	X	X	X

Fuente: Autor

5.1.5.2. Definición del modelo de sistemas de información, *figura 4*. Se representa el conjunto de sistemas de información que da soporte a los procesos de la organización afectados, describiendo sus relaciones e interfaces.

Figura 4. Modelo del sistema de información.



Fuente: Autor

5.1.6. Definición de la arquitectura tecnológica

5.1.6.1. Selección de la arquitectura tecnológica. Bajo esta tarea se realiza la selección de plataforma tecnológica determinando la arquitectura tecnológica como del tipo Cliente-Servidor. La selección de la arquitectura Cliente-Servidor es tomada teniendo en cuenta el entorno tecnológico utilizado en la empresa ALFCOM S.A.

El impacto a la organización se describe a continuación haciendo énfasis en el análisis de costos, *cuadro 6*.

5.1.6.1.1 Análisis de costos

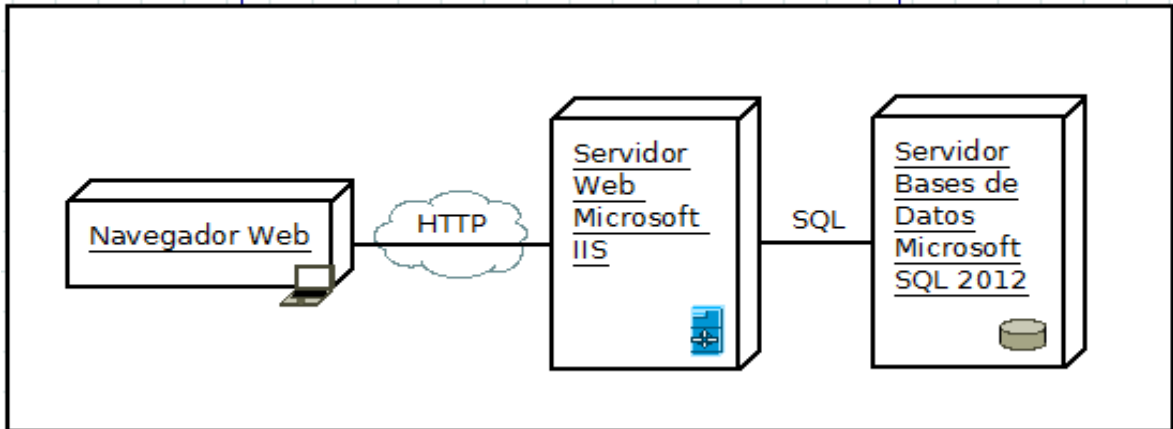
Cuadro 6. Costos fijos

Descripción	Cantidad	Valor Unitario	Valor Total
Computador portátil	2	\$ 2.000.000	\$ 4.000.000
Microsoft Visual Studio 2012 Express	2	\$ 0	\$ 0
Microsoft SQL server Express	2	\$ 0	\$ 0
NHibernate	2	\$ 0	\$ 0
Pentaho Reporting	2	\$ 0	\$ 0
Sueldo ingeniero (Mensual)	2	\$ 2.000.000	\$ 4.000.000
Windows 7 (Incluida en el equipo)	2	\$ 0	\$ 0
Subversion (Server)	1	\$ 0	\$ 0
AnkhSVN - Subversion Support for Visual Studio	2	\$ 0	\$ 0
TortoiseSVN	2	\$ 0	\$ 0
Servidor	1	\$ 2.000.000	\$ 2.000.000
Windows Server 2008	1	\$ 1.500.000	\$ 1.500.000
		Total	\$ 11.500.000

Fuente: Autor

5.1.6.1.2 Diagrama de representación, *figura 5*. Se utilizará una arquitectura cliente-servidor por medio del servidor web Microsoft IIS 7.0 y base de datos Microsoft SQL Server 2012. Los clientes podrán interactuar con la aplicación a través de un navegador web.

Figura 5. Diagrama de representación



Fuente: Autor

5.1.6.1.3 Impacto de la organización. El impacto en la organización se determina como de nivel bajo, dado que parte de los costos del proyecto son asumidos inicialmente por los creadores del proyecto, y los recursos necesarios para su implementación ya se encuentran disponibles por la empresa ALFCOM S.A. No se afectan procesos existentes de la organizaci

5.1.7. Definición del plan de acción, *cuadro 7*. En esta actividad se definen los proyectos y acciones a llevar a cabo para la implantación de los modelos de información y de sistemas de información, determinados en las actividades descritas a continuación.

Cuadro 7. Plan de acción - desarrollo de actividades.

Id.	Nombre de tarea	Comienzo	Fin	Duración	may 2014					jun 2014			
					27/4	4/5	11/5	18/5	25/5	1/6	8/6	15/6	22/6
1	PLANIFICACIÓN DE SISTEMAS DE INFORMACIÓN (PSI)	28/04/2014	07/05/2014	8d									
2	Inicio del plan de sistemas de información	28/04/2014	14/05/2014	13d									
3	Definición y organización	28/04/2014	14/05/2014	13d									
4	Estudio de la información relevante	28/04/2014	14/05/2014	13d									
5	Identificación de requisitos	28/04/2014	14/05/2014	13d									
6	Estudio de los sistemas de información actuales	28/04/2014	14/05/2014	13d									
7	Diseño del modelo de sistemas de información	28/04/2014	21/05/2014	18d									
8	Definición de la arquitectura tecnológica	28/04/2014	14/05/2014	13d									
9	Definición del plan de acción	28/04/2014	14/05/2014	13d									
10	DESARROLLO DE SISTEMAS DE INFORMACIÓN	28/04/2014	13/06/2014	35d									
11	Estudio de viabilidad del sistema (EVS)	14/05/2014	13/06/2014	23d									
12	Análisis del sistema de información (ASI)	28/04/2014	13/06/2014	35d									
13	Diseño del sistema de información (DSI)	28/04/2014	13/06/2014	35d									
14	Construcción del sistema de información (CSI)	28/04/2014	20/06/2014	40d									

Fuente: Autor

5.2. DESARROLLO DE SISTEMAS DE INFORMACIÓN

5.2.1. Estudio de viabilidad del sistema (EVS). El objetivo de esta tarea es el análisis de un conjunto concreto de necesidades para proponer una solución a corto plazo, que tenga en cuenta restricciones económicas, técnicas, legales y operativas. Para esta actividad se identifican los requisitos que se ha de satisfacer y se estudia la situación actual.

5.2.1.1. Establecimiento del alcance del sistema. Se estudia el alcance de la necesidad planteada, realizando una descripción general de la misma. Se determinan los objetivos, se inicia el estudio de los requisitos y se identifican las unidades organizativas afectadas estableciendo su estructura.

Se analizan las posibles restricciones, tanto generales como específicas, que puedan condicionar el estudio y la planificación de las alternativas de solución que se propongan.

5.2.1.1.1 Estudio de la solicitud

- Descripción general de la solución. El sistema de información planteado permitirá a la organización ALFCOM S.A. implementar un sistema de gestión de seguridad de la información basado inicialmente en el cumplimiento de los requisitos establecidos bajo la Norma Circular 052 de 2007 emitida por la Superintendencia Financiera de Colombia. El objetivo principal de esta herramienta es conocer el estado de implementación de la norma en todo momento, a través de un porcentaje de implementación total, el cual es calculado a través del porcentaje de cumplimiento definido en cada uno de los requerimientos de la norma a implementar; porcentaje de cumplimiento establecido por el personal encargado de esta tarea.

- Objetivos del Estudio de viabilidad
- Definir las características que convierten la solución planteada como factible.
- Fundamentar la puesta en marcha del proyecto

5.2.1.1.2 Identificación del alcance del sistema

- Descripción general del sistema, *cuadro 8*. El sistema involucra las unidades organizativas descritas a continuación, las cuales se ven asociadas en la implementación del proyecto:

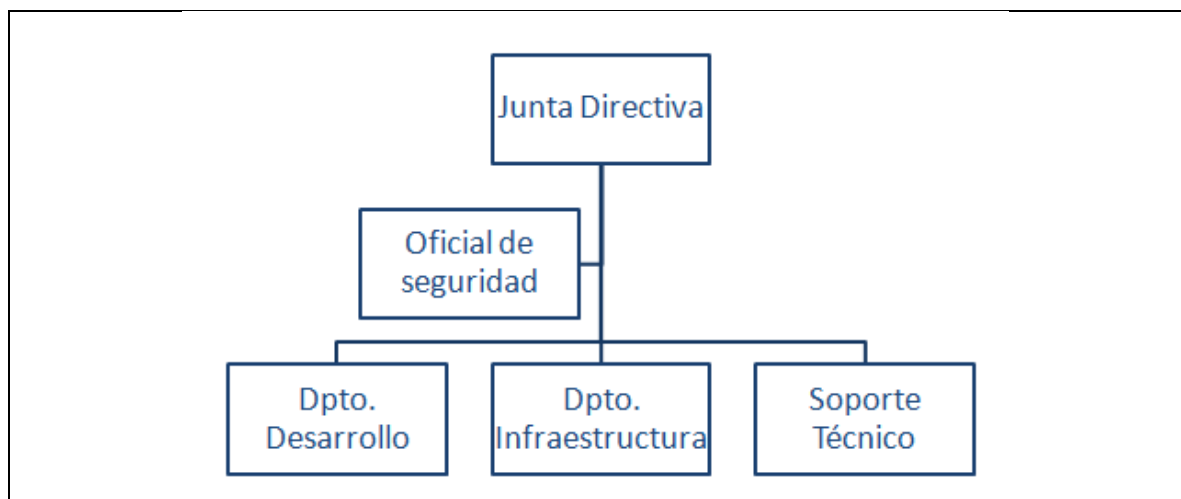
Cuadro 8. Descripción general del sistema - Áreas afectadas

Área afectada	Responsable
Departamento de desarrollo de software	Ingeniero Desarrollador
Departamento de infraestructura Tecnológica	Administrador de red
Alta Gerencia	Junta directiva
Soporte técnico	Jefe de soporte técnico
Seguridad de la información	Oficial de seguridad

Fuente: Autor

- Estructura organizativa: Se presenta la estructura organizativa, *figura 6*, de la organización.

Figura 6. Estructura organizativa



Fuente: Autor

- Catálogo de requisitos: El catalogo de requisitos se expone en el *cuadro 6*.

Cuadro 6. Catálogo de requisitos – Estudio de Viabilidad del Sistema

Área	Requisito
Junta directiva	Debe aprobar y apoyar la implementación del sistema de gestión de seguridad de la información (SGSI)
Área de desarrollo	Debe dar cumplimiento a las políticas establecidas por el SGSI aprobado por la alta gerencia
Área de infraestructura	Debe dar cumplimiento a las políticas establecidas por el SGSI aprobado por la alta gerencia y garantizar los recursos necesarios para el correcto funcionamiento del mismo.
Área de soporte técnico	Debe dar cumplimiento a las políticas establecidas por el SGSI aprobado por la alta gerencia ajustando sus procedimientos.
Oficial de seguridad	Debe implementar el SGSI con base en la aplicación planteada en el presente proyecto.

Fuente: Autor

5.2.1.1.3 Especificación del Alcance del EVS. Teniendo en cuenta los objetivos del Estudio de Viabilidad del Sistema, se determinan las actividades y tareas a realizar, *cuadro 7*. Plan de trabajo.

5.2.1.2. Estudio de la situación Actual. A través de esta actividad se describe el estado en el que se encuentran los sistemas de información existentes en el momento en el que se inicia su estudio.

5.2.1.3. Definición de requisitos del sistema. Esta actividad incluye la determinación de los requisitos generales definiendo los requisitos y sus prioridades, que se añaden al catálogo de requisitos que servirá para el estudio y valoración de las distintas alternativas de solución que se propongan.

5.2.1.3.1 Identificación de las Directrices Técnicas y de Gestión. Se definen los términos de referencia para el sistema en estudio desde el punto de vista de directrices tanto técnicas como de gestión.

- Políticas técnicas Desarrollo de sistemas:
 1. Se establece como norma de desarrollo del presente sistema de información el modelo Métrica V3²⁵.
 2. La técnica de programación utilizada corresponde al paradigma orientado a objetos.
 3. Como arquitectura se escoge el tipo cliente servidor, basado en el concepto de servidor web y navegador web como cliente.

²⁵ MÉTRICA es una metodología de planificación, desarrollo y mantenimiento de sistemas de información, promovida por el Ministerio de Hacienda y Administraciones Públicas (antiguo Ministerio de Administraciones Públicas) del Gobierno de España. <<https://es.wikipedia.org/wiki/M%C3%89TRICA>>

- Políticas de seguridad: Las políticas de seguridad, *cuadro 8*, describen las características a tener en cuenta con respecto a la seguridad.

Cuadro 8. Políticas de seguridad

Control de acceso	<ul style="list-style-type: none"> • Premisa: Todo debe estar prohibido a menos que se permita expresamente • Utilización de identificadores de usuarios únicos y acceso mediante contraseñas robustas • El nivel de acceso asignado es el adecuado de acuerdo a las funciones del usuario
Integridad de datos	<ul style="list-style-type: none"> • La empresa ALFCOM S.A. debe realizar backups periódicos de la información generada sobre el sistema. • La información ingresada en la aplicación (comentarios y/o observaciones) no debe ser modificada por ningún usuario.
Disponibilidad de aplicaciones	<ul style="list-style-type: none"> • La empresa ALFCOM S.A. debe garantizar una infraestructura de contingencia en caso de valorar la solución como crítica para el funcionamiento del negocio.

Fuente: Autor

- Directrices de planificación. Todo desarrollo debe ser evaluado antes de ser puesto en marcha de acuerdo a la necesidad del negocio planteado. Las mejoras que consideren las empresas sobre el producto, deben ser propuestas a los creadores del presente proyecto como una solución aparte.

5.2.1.3.2 Identificación de Requisitos

- La función principal de la aplicación debe ser la capacidad de implementar nuevas normas al sistema, de las cuales se podrá medir su implementación bajo el mismo esquema. Las normas pueden ser implementadas de acuerdo a las necesidades del negocio. Se pueden implementar cuantas normas sean necesarias, los requerimientos de las normas se implementan como controles.
- El sistema debe permitir hacer un control de la implementación de la norma escogida.
- Cada uno de los controles implementados acerca de la norma debe tener asociado un plan de acción. Cada Control debe tener asociado un valor de cumplimiento.
- El sistema medirá la implementación general de la norma a raíz de los valores de cumplimiento asignados a cada control.
- La aplicación permitirá la generación de reportes para conocer el estado de implementación de la norma.
- La solución debe permitir consultar las normas implementadas, convirtiéndola en una base de conocimientos.
- La aplicación permite conocer las últimas noticias relacionadas con la seguridad de la información a través de un canal rss enlazado con sitios de esta índole.

5.2.1.3.3 Catalogación de Requisitos. En el *cuadro 9*, se explica en detalle los requisitos, su catalogación y la prioridad.

Cuadro 9. Catálogo de requisitos

Requisito	Catalogación	Prioridad
La función principal de la aplicación debe ser la capacidad de implementar nuevas normas al sistema, de las cuales se podrá medir su implementación bajo el mismo esquema. Las normas pueden ser implementadas de acuerdo a las necesidades del negocio. Se pueden implementar cuantas normas sean necesarias, los requerimientos de las normas se implementan como controles.	Funcional	Alta
El sistema debe permitir hacer un control de la implementación de la norma escogida.	Funcional	Media
Cada uno de los controles implementados acerca de la norma debe tener asociado un plan de acción. Cada plan de acción debe tener asociado un valor de cumplimiento.	Funcional	Media
El sistema medirá la implementación general de la norma a raíz de los valores de cumplimiento asignados a cada control a través de su plan de acción.	Funcional	Media
La aplicación permite la generación de reportes para conocer el estado de implementación de la norma.	Funcional	Media
La solución debe permitir consultar las normas implementadas.	Funcional	Baja
La aplicación permitirá conocer las últimas noticias relacionadas con la seguridad de la información a través de un canal RSS enlazado con sitios de esta índole.	No funcional	Baja

Fuente: Autor

5.2.1.4. Estudio de alternativas de solución. Esta actividad se centra en proponer diversas alternativas que respondan satisfactoriamente a los requisitos planteados, considerando también los conceptos realizados en el Estudio de la Situación Actual.

5.2.1.4.1 Preselección de Alternativas de Solución, *cuadro 10*. El sistema inicialmente estará formado por las herramientas definidas en el diagrama de representación descrita anteriormente "Selección de la arquitectura tecnológica". A continuación se describen las soluciones a utilizar en el desarrollo del proyecto:

Cuadro 10. Preselección de alternativas de solución

Alternativas de solución	Información
Microsoft Visual Studio 2012 Express	Suite de desarrollo de aplicaciones de Microsoft.
Microsoft SQL server Express	Servidor de base de datos SQL de Microsoft
NHibernate	Framework de código abierto para mapear objetos a tablas relacionales para .Net, usando HQL o SQL nativo.
Pentaho Reporting	suite open-source que permite crear reportes relacionales y analíticos de un amplio rango de fuentes de datos y salidas de datos como: PDF, Excel, HTML, Texto, RTF (Rich-Text-File), XML y CSV
Subversion (Server)	Programa servidor para el control de versiones de productos de software
Internet Information Services	Servidor de aplicaciones basado en web de Microsoft.
PHP	PHP lenguaje de programación orientado a la creación de sitios web, y de ejecución del lado del servidor.
Servidor web Apache	Servidor web de código abierto, utilizado en la mayoría de sitios de internet

Fuente: Autor

5.2.1.4.2 Descripción de las Alternativas de Solución. En el *cuadro 11*, se describen las alternativas de la solución.

Cuadro 11. Descripción de alternativas de solución

Alternativas de solución	Descripción
Microsoft Visual Studio 2012 Express	Se utilizará la suite para el desarrollo total del sistema, utilizando como base de programación el lenguaje C# y asp.
Microsoft SQL server Express	Servidor de base de datos SQL donde se almacenarán todos los registros al utilizar la aplicación. Se soporta en el software Nhibernate para el envío de la información a la base de datos.
NHibernate	Mapea objetos a tablas relacionales para .Net, usando HQL o SQL nativo.
Pentaho Reporting	Software para la generación de reportes de lo implementado en el software.
Subversion (Server)	Programa servidor para el control de versiones de productos de software. Garantizará el control de modificación del software durante su implementación y mantenimiento.
Internet Information Services	Servidor en el que se ejecuta la aplicación utilizando tecnologías como: html, aspx, css, javascript.
PHP	Servidor de aplicaciones que se interpreta en ejecución a través de archivos de extensión php.
Servidor Apache	Servidor web donde son ejecutados los archivos .php de la aplicación.

Fuente: Autor

5.2.1.5. Valoración de las alternativas. A continuación se realiza una valoración de las alternativas de solución, considerando el impacto en la organización, tanto desde el punto de vista tecnológico, organizativo, de operación, y los posibles beneficios que se esperan contrastados con los costes asociados. Se realiza también un análisis de los riesgos, decidiendo cómo enfocar el plan de acción para minimizar los mismos y cuantificando los recursos y plazos precisos para planificar cada alternativa.

5.2.1.5.1 Estudio de la Inversión

- Estimación de costes. La estimación de costes, *cuadro 12*, amplía la visión sobre los mismos.

Cuadro 12. Estimación de costes

Tipo	Detalle	Costo	Periodo de tiempo
Adquisición de hardware y software	Servidor	\$ 2.000.000	1 vez
	Windows Server 2003/2008	\$ 1.500.000	1 vez
Gastos de comunicaciones	Funciona bajo red local	\$ 500.000	1 vez
Gastos de instalación	Servicio de instalación de la aplicación	\$ 200.000	1 vez
Coste de desarrollo	Total – “Tabla 6. costos fijos”	\$ 11.500.000	1 vez
Gastos del mantenimiento del sistema	Soporte a la aplicación	\$ 1.000.000	Anual

Fuente: Autor

- Estimación de beneficios tangibles. La estimación de los beneficios tangibles se presenta en el *cuadro 13* descrito a continuación.

Cuadro 13. Estimación de beneficios tangibles

Tipo	Detalle	Valor	Periodo de tiempo
Ahorro de gastos de mantenimiento	El mantenimiento del sistema se encuentra bajo el modelo de soporte técnico anual. Se evita el gasto de una persona dedicada al mantenimiento del sistema. [(SM [\$800.000] *12 meses)]	9600000	Anual
Horas de trabajo en la implementación de normas	Se disminuye hasta 1/3 las horas de trabajo dedicadas a la implementación de normas: Ahorro: [(Sueldo Mensual/3)*2*12 meses]. Para una persona que gana 1000000 mensuales y dedica la tercera parte de su tiempo (Sueldo mes dedicado = \$333.333) en la implementación de la norma.	4000000	Anual

Fuente: Autor

- Estimación de beneficios intangibles. La estimación de los beneficios tangibles se presenta en la *cuadro 14*.

Cuadro 14. Estimación de beneficios intangibles

Tipo	Detalle
Medición en la implementación de normas	Se mide la implementación de una norma a medida que se realiza.
Menor tiempo en la implementación de normas	Se utiliza la herramienta como mecanismo centralizado de información, con un concepto de medición constante
Información actualizada dirigida a la alta gerencia	Permite la toma de decisiones de forma eficaz y eficiente.
Mejor respuesta ante auditorias	La información es mostrada a los organismos de auditoría de forma centralizada

Fuente: Autor

- Retorno de inversión. El ROI o retorno de la inversión se contempla en la *tabla 1*.

Tabla 1. Retorno de la inversión

Año	Coste	Beneficio	Beneficio Neto
0	\$ 16.700.000	\$ 0	
1	\$ 1.000.000	\$ 13.600.000	\$ 12.600.000
2	\$ 1.000.000	\$ 13.600.000	\$ 12.600.000
Beneficio Total			\$ 25.200.000

Fuente: Autor

El año de la recuperación de la inversión se produce en el año 2 cuando el Beneficio total es igual o superior al costo del año 0.

5.2.1.5.2 Estudio de los Riesgos

- Identificación de las alternativas críticas descritas en el *cuadro 15*.

Cuadro 15. Identificación de las alternativas críticas

Tipo de riesgo	Valoración
Complejidad en el uso de la solución (C)	3. Alta complejidad en el uso de la solución 2. Media complejidad en el uso de la solución 1. Baja complejidad en el uso de la solución
Coste de adquisición de la tecnología (CA)	3. Alto Costo = \$ 30.000.000 2. Medio Costo = \$ 15.000.000 1. Bajo Costo = \$ 2.000.000
Tiempo de sustitución de lo antiguo (T)	3. Largo tiempo = 2 años 2. Medio tiempo = 1 año 1. Corto tiempo = 3 meses
Rechazo cultural de la organización (R)	3. Alto tiempo de capacitación y baja eficiencia al realizar las tareas 2. Tiempo razonable en capacitación y eficiencia moderada al realizar las tareas 1. Fácil aprendizaje de la solución y mejora de la eficiencia al realizar las tareas
Miedo ante la elección de tecnologías inmaduras (M)	3. Alto riesgo de mal funcionamiento y errores; no existe soporte técnico 2. Medio riesgo de mal funcionamiento y errores esporádicos; soporte técnico deficiente 1. Funcionamiento correcto y esperado; soporte técnico eficiente

Fuente: Autor

Valoración de riesgos, son descritas a través del *cuadro 16* Inventario de alternativas críticas.

- De la sumatoria de C+CA+T+R+M se obtienen diferentes valores
- Se debe priorizar la selección por el nivel crítico alcanzado
- Las que sumen entre 11 y 15 puntos son descartables
- Las que sumen entre 8 y 10 son opcionales
- Las que sumen menos de 8 puntos son seleccionables.

Cuadro 16. Inventario de alternativas críticas

Inventario de alternativas críticas (AC)								
Lista de alternativa	Peligros y pérdidas	Evaluación de riesgo					AC	
		C	CA	T	R	M		
Microsoft Visual Studio 2012 Express	Factores de riesgo en el desarrollo del proyecto	2	1	1	2	1	7	
Microsoft SQL server Express	Factores de riesgo en el desarrollo del proyecto	2	1	1	2	1	7	
NHibernate	Factores de riesgo en el desarrollo del proyecto	2	1	1	1	2	7	
Pentaho Reporting	Factores de riesgo en el desarrollo del proyecto	3	1	1	3	3	11	
Subversion (Server)	Factores de riesgo en el desarrollo del proyecto	1	1	1	1	1	5	
Internet Information Services	Factores de riesgo en la implementación del proyecto	1	1	1	1	1	5	
PHP	Factores de riesgo en el desarrollo del proyecto	1	1	3	2	2	9	
Servidor Apache	Factores de riesgo en la implementación del proyecto	1	1	1	1	1	5	

Fuente: Autor

5.2.1.5.3 Planificación de Alternativas, *figura 7*. A continuación se describe el plan de acción de uso de las alternativas, asociado a las etapas de construcción (CSI) e implantación (IAS) del sistema.

Figura 7. Planificación de las alternativas.

Id.	Nombre de tarea	Comienzo	Fin	Duración	may 2014					jun 2014			
					27/4	4/5	11/5	18/5	25/5	1/6	8/6	15/6	
1	Microsoft Visual Studio 2012 Express	28/04/2014	13/06/2014	35d									
2	Microsoft SQL server Express	28/04/2014	13/06/2014	35d									
3	NHibernate	28/04/2014	13/06/2014	35d									
4	Subversion (Server)	28/04/2014	13/06/2014	35d									
5	Internet Information Services	02/06/2014	20/06/2014	15d									
6	Pentaho Reporting	02/06/2014	20/06/2014	15d									

Fuente: Autor

5.2.1.6. Selección de la Solución

5.2.1.6.1 Evaluación de las Alternativas y Selección

- Solución propuesta. La solución escogida para el proyecto utiliza como lenguaje de programación C# asociado al lenguaje de programación web asp, todo desarrollado a través de la suite Microsoft Visual Studio 2012.

Se utiliza el servidor de bases de datos Microsoft SQL Server 2012 como Sistema de gestión de bases de datos (SGBD).

Para la publicación e implementación de la aplicación se utiliza el servidor de aplicaciones Microsoft Internet Information Services, ejecutado bajo el sistema operativo Windows Server 2008.

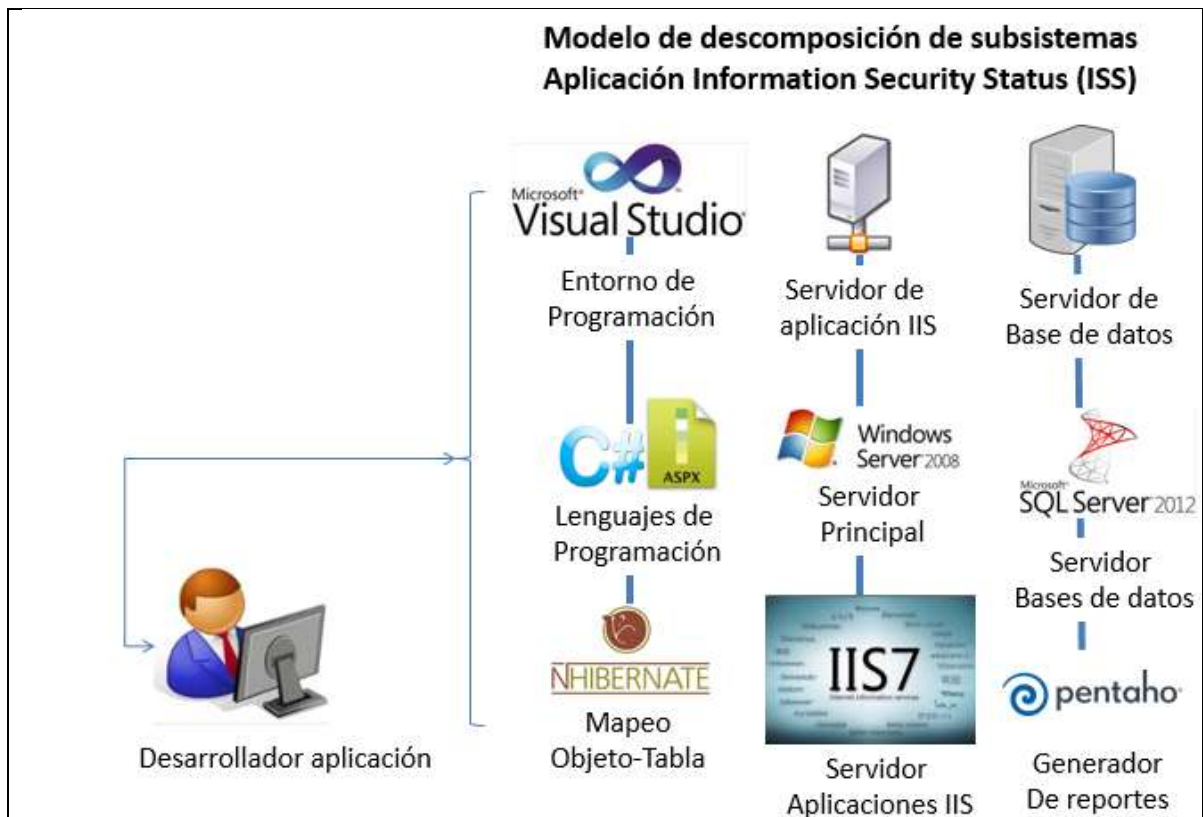
Para el manejo de versionamiento en el desarrollo de la aplicación se utiliza el servidor Subversión.

Se utiliza la aplicación NHibernate para mapear objetos de la aplicación a tablas relacionales definidas sobre el SGBD Microsoft SQL Server 2012.

Se utiliza como herramienta de generación de reportes la aplicación Pentaho Community.

A continuación, *figura 8*, se describen la selección de la solución a través del modelo de descomposición de Subsistemas, Matriz Procesos / Localización Geográfica, *cuadro 17* y Matriz Datos / Localización Geográfica, *cuadro 18*.

Figura 8. Modelo de descomposición de subsistemas



Fuente: Autor

Cuadro 17. Matriz Procesos / Localización Geográfica

Procesos \ Localización	Dpto. Seguridad	Junta Directiva	Dpto. Desarrollo	Dpto. Infraestructura	Soporte Técnico
Implementación del Sistema de gestión de seguridad de la información	x				
Mantenimiento de la infraestructura de la aplicación				x	
Soporte a la aplicación					x
Revisión del SGSI	x	x			
Mejora continua del SGSI	x	x	x	x	x

Fuente: Autor

Cuadro 18. Matriz Datos / Localización Geográfica

Datos \ Localización	Dpto. Seguridad	Junta Directiva	Dpto. Desarrollo	Dpto. Infraestructura	Soporte Técnico
Normas	x	x			
Controles	x		x	x	x
Procesos	x	x	x	x	x
Documentación		x	x	x	x

Cuadro 18. (Continuación)

Datos \ Localización	Dpto. Seguridad	Junta Directiva	Dpto. Desarrollo	Dpto. Infraestructura	Soporte Técnico
Plan de la organización		x			
Información del negocio		x			
Datos Clientes		x			x

Fuente: Autor

- Estrategia de implantación global del sistema definido en el *cuadro 19*. El sistema será implementado en la infraestructura existente de la empresa ALFCOM S.A., haciendo uso del licenciamiento, servidores, redes de comunicaciones y sistemas de almacenamiento de datos dispuestos por la organización, teniendo presente los requerimientos del sistema.

Cuadro 19. Elementos de la implantación

Elemento	Descripción
Servidor	Microsoft Windows Server 2008 R2
Base de datos	Microsoft SQL server 2012
Licenciamiento	Visual Studio 2012 Microsoft SQL Server 2012 Microsoft Office
Almacenamiento	NAS de la organización

Autor: Fuente

5.2.1.6.2 Aprobación de la Solución. La solución es aprobada por la empresa ALFCOM S.A., dado que se cuenta con los recursos necesarios para el Desarrollo, Implementación y Mantenimiento de la solución dentro de la organización.

5.2.2. Análisis del sistema de información (ASI). El objetivo de este proceso es la obtención de una especificación detallada del sistema de información que satisfaga las necesidades de información de los usuarios y sirva de base para el posterior diseño del sistema.

5.2.2.1. Definición del sistema. Se realiza la descripción del sistema, delimitando su alcance, estableciendo las interfaces con otros sistemas e identificando a los usuarios representativos. Se parte de los productos obtenidos en las actividades anteriores, para proceder con la adecuación y definición del sistema de información.

5.2.2.1.1 Determinación del Alcance del sistema. A continuación en el *cuadro 20* se delimita el sistema de información, utilizando como punto de partida el modelo de procesos especificado en la descripción de la solución del proceso Estudio de Viabilidad del Sistema. En esta actividad se realiza la definición del catálogo de requisitos del sistema. Igualmente se establece el contexto del sistema a través de la definición del modelo de del negocio.

- Catálogo de requisitos

Cuadro 20. Requisito 1 – Implementar nuevas normas

Identificador	R1 – Implementar nuevas normas	Prioridad	Alta
Descripción	La función principal de la aplicación debe ser la capacidad de implementar nuevas normas al sistema, de las cuales se podrá medir su implementación bajo el mismo esquema.		
Inicio	La nueva norma o circular a agregar a la aplicación		
Fin	La base de datos actualizada.		
Requerimientos No funcionales asociados	Disponibilidad, Seguridad	Actor	Administrador

Cuadro 20. (Continuación)

Control de versiones		
Fecha	Descripción	Responsable
24/04/2014	Creación del requerimiento	Juan Ayala – Dario Acosta

Fuente: Autor

Implementar nuevas formas; la función principal de la aplicación debe ser la capacidad de implementar nuevas normas al sistema, de las cuales se podrá medir su implementación bajo el mismo esquema, *cuadro 21*.

Cuadro 21. Requisito 2 – Controlar las normas

Identificador	R2 – Controlar las normas	Prioridad	Media
Descripción	El sistema debe permitir hacer un control de la implementación de la norma escogida.		
Inicio	La nueva norma a la que se le aplicará el control.		
Fin	La base de datos actualizada.		
Requerimientos No funcionales asociados	No aplica	Actor	consultor
Control de versiones			
Fecha	Descripción	Responsable	
24/04/2014	Creación del requerimiento	Juan Ayala – Dario Acosta	

Fuente: Autor

Controlar las normas; El sistema debe permitir hacer un control de la implementación de la norma escogida, *cuadro 22*.

Cuadro 22. Requisito 3 – Crear planes de acción sobre los controles

Identificador	R3 – Crear planes de acción sobre los controles	Prioridad	Media
Descripción	Cada uno de los controles implementados acerca de la norma debe tener asociado un plan de acción. Cada plan de acción debe tener asociado un valor de cumplimiento.		
Inicio	A cada control se le pueden asociar planes de acción.		
Fin	La base de datos actualizada.		
Requerimientos No funcionales asociados	No aplica	Actor	consultor
Control de versiones			
Fecha	Descripción	Responsable	
24/04/2014	Creación del requerimiento	Juan Ayala – Dario Acosta	

Fuente: Autor

Crear planes de acción sobre los controles, *cuadro 23*; Cada uno de los controles implementados acerca de la norma debe tener asociado un plan de acción. Cada plan de acción debe tener asociado un valor de cumplimiento.

Cuadro 23. Requisito 4 – Medir implementación general

Identificador	R4 – Medir implementación general	Prioridad	Media
Descripción	El sistema medirá la implementación general de la norma a raíz de los valores de cumplimiento asignados a cada control a través de su plan de acción.		
Inicio	Norma a medir.		
Fin			
Requerimientos No funcionales asociados	No aplica	Actor	Administrador, consultor y auditor.
Control de versiones			
Fecha	Descripción	Responsable	
24/04/2014	Creación del requerimiento	Juan Ayala – Dario Acosta	

Fuente: Autor

Medir implementación general, *cuadro 24*; El sistema medirá la implementación general de la norma a raíz de los valores de cumplimiento asignados a cada control a través de su plan de acción.

Cuadro 24. Requisito 5 – Medir implementación general

Identificador	R5 – Medir implementación general	Prioridad	Media
---------------	-----------------------------------	-----------	-------

Cuadro 24. (Continuación)

Descripción	La aplicación permitirá la generación de reportes para conocer el estado de implementación de la norma.		
Inicio	Selección de la norma a la que se le quiere generar el reporte		
Fin	Reporte generado.		
Requerimientos No funcionales asociados	No aplica	Actor	Administrador, consultor y auditor.
Control de versiones			
Fecha	Descripción	Responsable	
24/04/2014	Creación del requerimiento	Juan Ayala – Dario Acosta	

Fuente: Autor

Medir implementación general, *cuadro 24*; La aplicación permitirá la generación de reportes para conocer el estado de implementación de la norma.

Cuadro 25. Requisito 6 – Consultar las normas

Identificador	R6 – Consultar las normas	Prioridad	Media
Descripción	La solución debe permitir consultar las normas implementadas.		
Inicio	Norma a consultar		
Fin	Lectura de la base de datos.		

Cuadro 25. (Continuación)

Requerimientos No funcionales asociados	No aplica	Actor	Administrador, consultor y auditor.
Control de versiones			
Fecha	Descripción	Responsable	
24/04/2014	Creación del requerimiento	Juan Ayala – Dario Acosta	

Fuente: Autor

Consultar las normas, *cuadro 25*; La solución debe permitir consultar las normas implementadas.

- Modelo de Negocio

En la *cuadro 26*, se expone el modelo de negocio.

Cuadro 26. Modelo de negocio - Lienzo de modelo de negocios²⁶

Problema	Solución	Proposición de valor única	Ventaja especial	Segmentos de clientes
Problemas en la implementación de un SGSI. - Falta de medición. - Múltiples documentos perdiendo el orden de las cosas. - Información poco clara para la alta gerencia. - Actualización constante de normas y leyes.	Aplicación centralizada que permite registrar el avance en la implementación de un SGSI o norma.	Aplicación centralizada que permite registrar el avance en la implementación de un SGSI o norma.	Modalidad adaptable para implementar nuevas normas	Empresas dedicadas al desarrollo de software financiero en Colombia. Dirigida inicialmente a la empresa ALFCOM S.A.
	Métricas clave Medición constante de la implementación de un SGSI o Norma.	Modalidad adaptable para implementar nuevas normas. Medición dirigida a la alta gerencia para la toma de las decisiones.	Canales Empresa Alcom Internet.	
Estructura de costos Costos Primer año: \$16.700.000 Costos años siguientes: \$1.000.000		Flujo de ingreso Ahorro anual en gastos de mantenimiento: \$9.600.000 Ahorro anual de horas de trabajo en implementación de normas: \$4.000.000		

Fuente: Autor

²⁶ GIZN. Modelo de negocios. En: Mercadeo para emprendedores. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <<http://gizn.com/recursos/modelo-de-negocios/>>.

5.2.2.1.2 Identificación del Entorno Tecnológico. Se describe el entorno tecnológico de los ambientes de desarrollo, pruebas y producción de la siguiente forma.

- Entorno de desarrollo. Está compuesto por los dos equipos portátiles de los creadores del proyecto, utilizando las herramientas de desarrollo definidas en el modelo de descomposición de subsistemas descrita en la sección Evaluación de las alternativas de solución. El versionamiento de la aplicación es efectuado en el servidor ubicado en la empresa ALFCOM.

El software utilizado por el equipo de desarrollo son:

1. Microsoft Visual Studio 2013
2. Nhibernate versión 3.3.3
3. Servidor de bases de datos Microsoft SQL Server 2012.
4. Aplicación de diseño de imágenes InkScape versión 0.48
5. Navegador Mozilla Firefox
6. Servidor de versiones de software Subversión 1.8.8

- Entorno de pruebas. Teniendo en cuenta la magnitud del proyecto, el entorno de desarrollo es implementado con los mismos equipos portátiles de los creadores del proyecto. Las herramientas de software utilizadas son las mismas anteriormente definidas en el entorno de desarrollo.

- Entorno de producción. El entorno de producción está compuesto por el servidor Windows Server 2008 configurado en la empresa ALFCOM S.A., el cual tiene instalado el servidor de aplicaciones Microsoft Internet Information Services IIS y el servidor de bases de datos Microsoft SQL Server 2012.

La aplicación puede ser abierta por cualquier computador cliente a nivel de red local utilizando un navegador web.

El software utilizado se describe a continuación:

1. Microsoft Windows Server 2008
2. Microsoft Internet Information Services ISS 7
3. Microsoft SQL Server 2012
4. Navegador web (Internet Explorer, Mozilla Firefox, Google Chrome, etc.)

5.2.2.1.3 Especificación de Estándares y Normas. Se consideran las referencias para el sistema de información en estudio, desde el punto de vista de estándares, normativas, leyes o recomendaciones, que deben tenerse en cuenta a lo largo de todo el proceso de desarrollo. A continuación, en el *cuadro 27*, se describe el catálogo de normas seleccionado:

Cuadro 27. Catálogo de normas

Norma	Descripción
Norma ISO 27001	Modelo para establecer, implementar, operar, monitorear, revisar, mantener y mejorar un Sistema de Gestión de Seguridad de la Información (SGSI).
Norma ISO 27002	Objetivos de control y controles diseñados para ser implementados para satisfacer los requerimientos identificados para una evaluación de riesgo.
Circular 052 de 2007	Obliga a las entidades vigiladas tener en cuenta en la contratación de outsourcing lineamientos asociados con la seguridad de la información.
Métrica V3.	Metodología para planificar, desarrollar y mantener sistemas de información.

Fuente: Autor

5.2.2.1.4 Identificación de los Usuarios Participantes y Finales. A continuación se identifican los usuarios participantes y finales, relacionados tanto en la obtención de requisitos como en la validación de los distintos productos y la aceptación final del sistema.

- Catálogo de usuarios. En el *cuadro 28*, se describen los usuarios involucrados en la creación del proyecto. Se tiene en cuenta el plan de trabajo definido en la Especificación del alcance del Estudio de viabilidad del sistema.

Cuadro 28. Catálogo de usuarios

Usuario	Descripción	Tipo
Creadores del proyecto	Estudiantes de la especialización en seguridad informática de la Universidad Piloto de Colombia, autores del presente documento, quienes se encargan de la planeación, desarrollo y mantenimiento de la solución propuesta.	Participante
Tutor Tesis	Valida y autoriza el desarrollo del presente proyecto.	Participante
Junta Directiva ALFCOM	Valida y autoriza la implementación del presente proyecto	Participante
Departamento de Desarrollo ALFCOM	Coordina con los autores del proyecto para lograr su desarrollo e implementación.	Interlocutor
Departamento de infraestructura ALFCOM	Coordina con los autores del proyecto para lograr su implementación	Participante
Departamento de seguridad ALFCOM	Implementa el SGSI con base en la solución propuesta en este proyecto.	Participante

Fuente: Autor

5.2.2.2. Establecimiento de requisitos. Se definen, analizan y validan los requisitos a partir de la información facilitada por el usuario, completándose el catálogo de requisitos obtenido en la actividad Definición del Sistema (ASI 1). El objetivo de esta actividad es obtener un catálogo detallado de los requisitos, a partir del cual se pueda comprobar que los productos generados en las actividades de modelización se ajustan a los requisitos de usuario.

5.2.2.2.1 Obtención de Requisitos. A continuación se define los modelo de casos de uso, además del catálogo de requisitos a través de la categorías: Funcionales, Rendimiento, Seguridad, Implantación y Sistema.

- Catálogo de requisitos

Requisitos Funcionales

- El sistema debe estar orientado a la implementación de controles de diferentes normas de seguridad de la información
- El sistema debe evaluar de forma general el estado de implementación del SGSI teniendo en cuenta el porcentaje establecido por la persona encargada de esta tarea.
- El sistema debe reportar en todo momento el estado de implementación a través de un valor (0-100%) en la esquina superior derecha de la interfaz gráfica.
- El sistema debe permitir consultar las diferentes normas implementadas en modo de lectura.
- El sistema debe permitir agregar la licencia o comprobante de la norma comprada legalmente (Si aplica).
- El sistema debe permitir acceder bajo 3 perfiles: auditor, consultor, y administrador.
- Bajo ninguno de los perfiles se permitirá modificar los comentarios y observaciones realizados a los controles (Garantizar integridad).

Requisitos de Rendimiento

- La aplicación debe funcionar simultáneamente para mínimo 5 usuarios sin verse afectado su rendimiento.
- La aplicación debe ser escalable en función de las características del servidor de aplicaciones y servidor de base de datos.

Requisitos Seguridad

- El sistema debe contar con un mecanismo de control de acceso, donde se limite el ingreso a los funcionarios autorizados.
- La aplicación debe permitir la asignación de roles de acuerdo a las funciones del usuario.
- Solo el rol administrador es el encargado de la generación de nuevos usuarios y la asignación de roles.

Requisitos Implantación

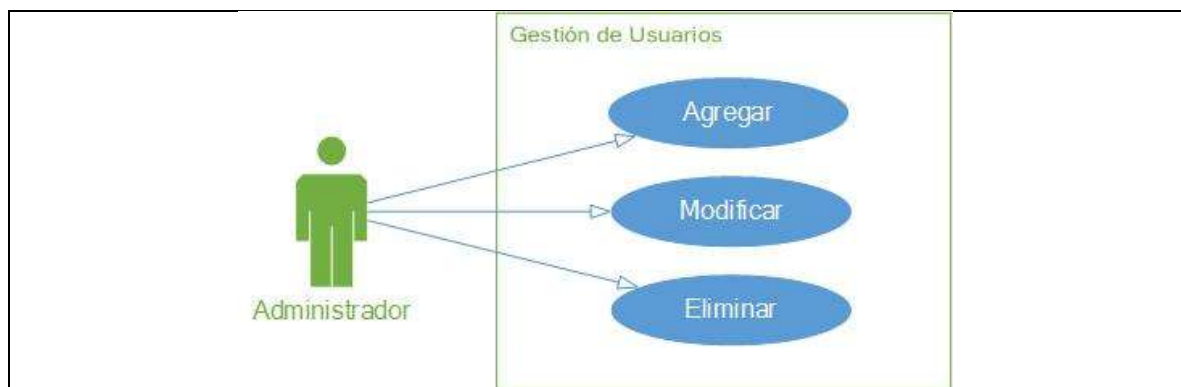
- El sistema debe poder ser instalado en la infraestructura existente de la compañía ALFCOM.
- La aplicación debe ser ejecutada bajo plataformas Windows.

Requisitos de disponibilidad del sistema

- La aplicación debe poder ser accesible desde cualquier lugar y dispositivo de la compañía, incluso desde internet.

- Modelo de casos de uso

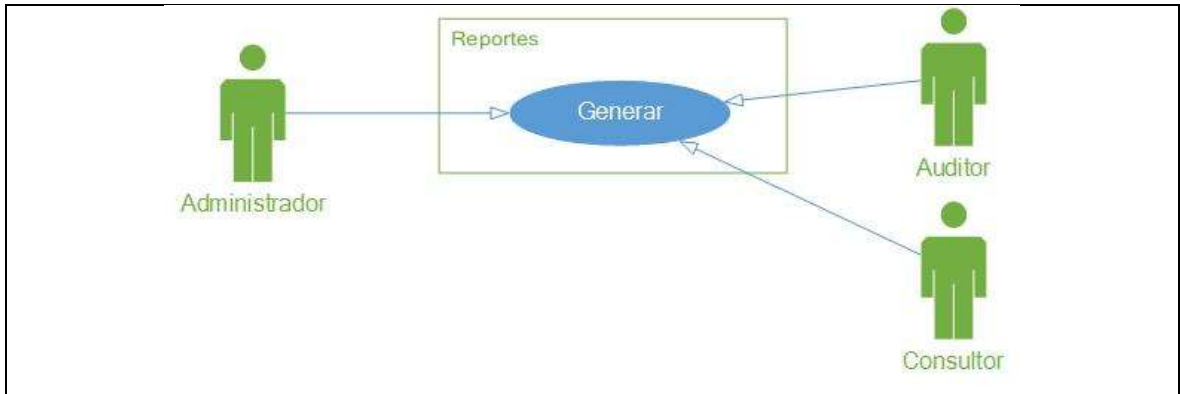
Figura 9. Gestión de Usuarios



Fuente: Autor

La *figura 9* muestra el usuario de tipo administrador con los privilegios sobre la gestión de usuarios.

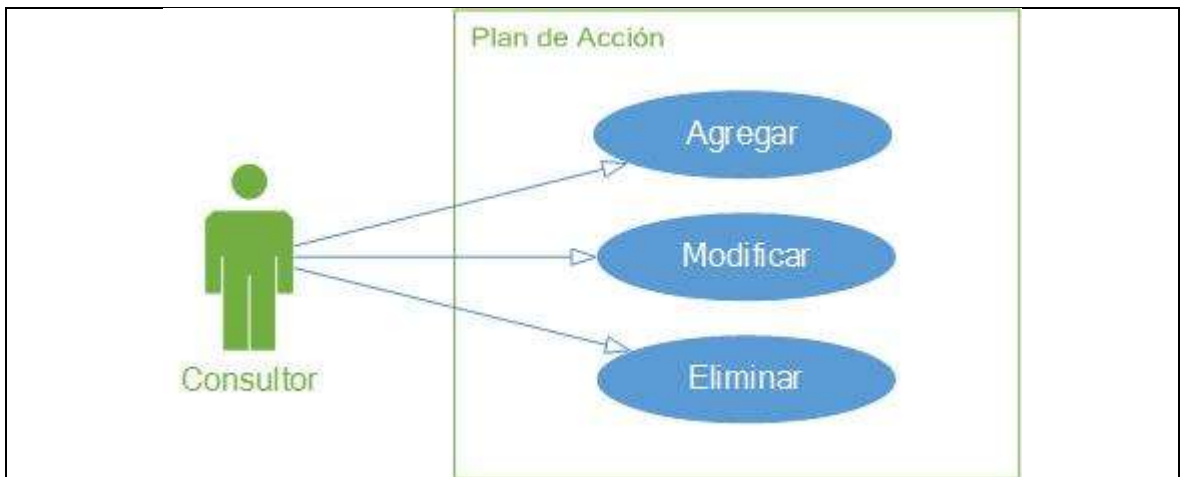
Figura 10. Generación de reportes



Fuente: Autor

Para la generación de reportes, todos los usuarios podrán generarlos como se muestra en la *figura 10*.

Figura 11. Gestión de plan de Acción



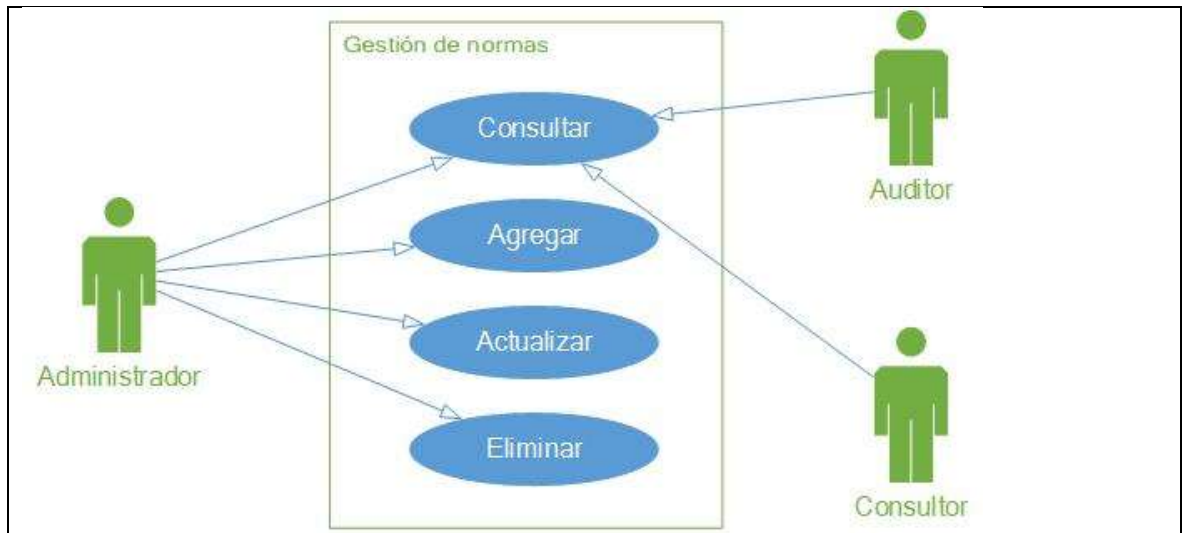
Fuente: Autor

La *figura 11*, muestra el usuario con rol consultor y las acciones que tiene sobre los planes de acción.

Para la gestión de normas, *figura 12*, se muestran los roles de los usuarios con los privilegios sobre las mismas.

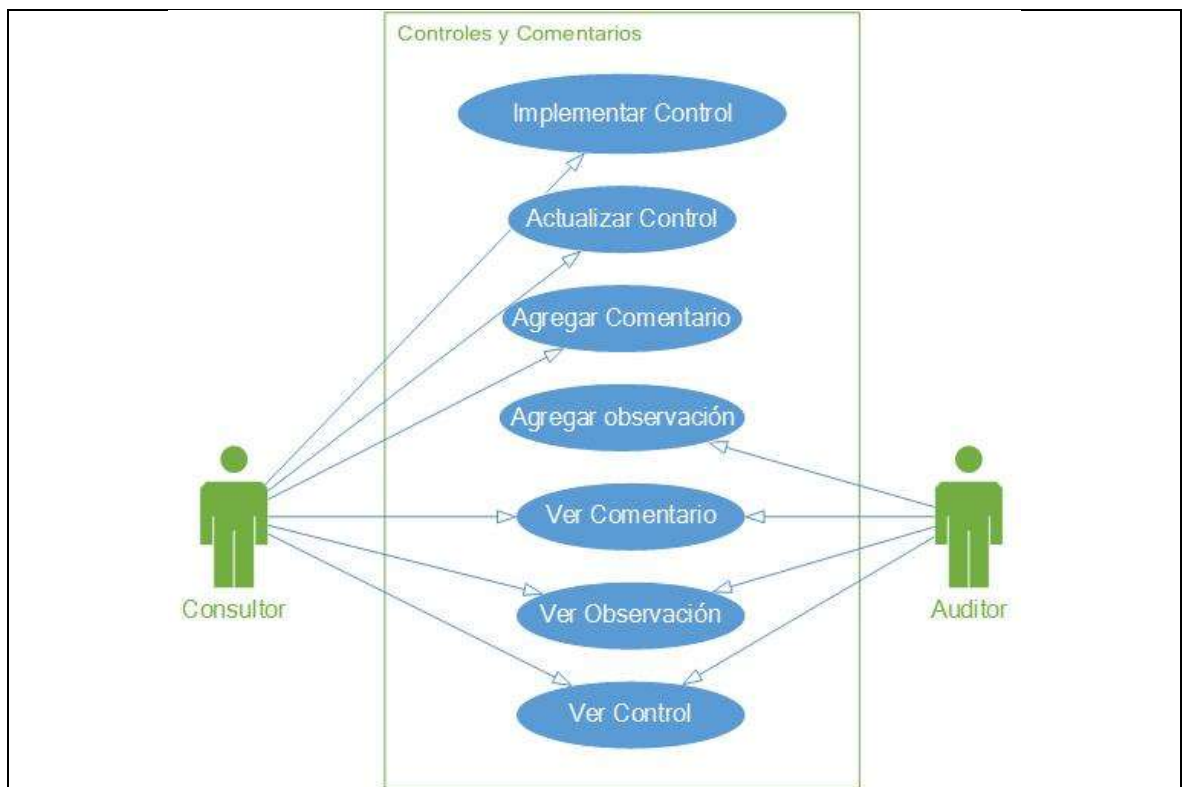
La gestión de controles y comentarios, *figura 13*, se especifican los privilegios de los roles para realizar acciones sobre cada uno.

Figura 12. Gestión de Normas



Fuente: Autor

Figura 13. Gestión de Controles y comentarios



Fuente: Autor

5.2.2.2.2 Especificación de Casos de Uso. Se especifica cada caso de uso identificado en la tarea anterior, desarrollando su escenario.

Cuadro 29. Especificación de Casos de uso UC-001

ID	UC-001	
Nombre	Gestión de Usuarios	
Tipo	Indispensable (Implementación obligatoria)	
Prioridad	Alta	
Actores involucrados	Administrador	
DESCRIPCIÓN DEL CASO DE USO		
El administrador del sistema podrá gestionar a los usuarios de la aplicación, tendrá el privilegio de Agregar, modificar o eliminar a los usuarios.		
ENTRADAS		SALIDAS
<ul style="list-style-type: none"> • Usuario 		<ul style="list-style-type: none"> • Mensaje del estado del evento.
PRECONDICIONES		POSTCONDICIONES
<ul style="list-style-type: none"> • Usuario con privilegios de administrador esté inscrito al sistema. 		<ul style="list-style-type: none"> • No aplica
DEPENDENCIA		EXTENSIÓN
<ul style="list-style-type: none"> • No aplica 		<ul style="list-style-type: none"> • No aplica
FLUJO NORMAL DE EVENTOS		
No.	Acción del actor	Respuesta del sistema
1	Seleccionar el evento a realizar	Permitir o denegar la acción. Se envía el formulario para ser diligenciado dependiendo de la acción.

Cuadro 29. (Continuación)

2	Ingresar los datos para crear o modificar el usuario. Si la acción es la eliminación se deberá confirmar que desea eliminar el registro.	Valida la información e informa de errores en caso de existir.		
CAMINOS ALTERNATIVOS				
No. actividad en flujo normal		No	Acción del actor	Respuesta del sistema
Se presenta la alternativa	Se integra al flujo normal			
No aplica	No aplica		No aplica	No aplica
CAMINOS DE EXCEPCIÓN				
<ul style="list-style-type: none"> En la actividad 2 el sistema puede retornar una excepción del tipo de llave duplicada, deberá seleccionar otro dato de ingreso 				
CRITERIOS DE ACEPTACIÓN				
<ul style="list-style-type: none"> Únicamente el usuario con el rol de administrador puede realizar las acciones descritas en este caso de uso. 				
REQUERIMIENTOS NO FUNCIONALES ASOCIADOS				
No aplica				
SEGUIMIENTO AL CASO DE USO				
Fecha	Descripción		Gestor	
02/04/2014	Creación del caso		Juan Ayala – Dario Acosta	

Fuente: Autor

Cuadro 30. Especificación de Casos de uso UC-002

ID	UC-002	
Nombre	Generar reportes	
Tipo	Indispensable (Implementación obligatoria)	
Prioridad	Alta	
Actores involucrados	Administrador, Auditor y Consultor	
DESCRIPCIÓN DEL CASO DE USO		
Los reportes que genera la aplicación.		
ENTRADAS		SALIDAS
<ul style="list-style-type: none"> Datos para la generación del reporte. 		<ul style="list-style-type: none"> Reporte con los datos suministrados.
PRECONDICIONES		POSTCONDICIONES
<ul style="list-style-type: none"> Los datos del reporte los selecciona el actor. 		<ul style="list-style-type: none"> No aplica
DEPENDENCIA		EXTENSIÓN
<ul style="list-style-type: none"> No aplica 		<ul style="list-style-type: none"> No aplica
FLUJO NORMAL DE EVENTOS		
N o.	Acción del actor	Respuesta del sistema
1	Ingresa a la página deseada para generar el reporte.	Muestra los datos del reporte
2	Selecciona los datos que desee que lleve el reporte	Muestra los datos filtrados
3	Genera el reporte	Envía el reporte.

Cuadro 30. (Continuación)

CAMINOS ALTERNATIVOS				
No. actividad en flujo normal		No	Acción del actor	Respuesta del sistema
Se presenta la alternativa	Se integra al flujo normal			
No aplica	No aplica		No aplica	No aplica
CAMINOS DE EXCEPCIÓN				
<ul style="list-style-type: none"> Ninguno 				
CRITERIOS DE ACEPTACIÓN				
<ul style="list-style-type: none"> El reporte se deberá mostrar con los datos seleccionados. 				
REQUERIMIENTOS NO FUNCIONALES ASOCIADOS				
<ul style="list-style-type: none"> No aplica 				
SEGUIMIENTO AL CASO DE USO				
Fecha	Descripción		Gestor	
20/04/2014	Creación del caso de uso UC-002		Juan Ayala – Dario Acosta	

Fuente: Autor

El *cuadro 29*, presenta a profundidad la gestión de normas representada en la *figura 9*.

La función generar reportes, *cuadro 30*, muestra los detalles del caso de uso de la *figura 10*.

Cuadro 31. Especificación de Casos de uso UC-003

ID	UC-003	
Nombre	Plan de Acción	
Tipo	Indispensable (Implementación obligatoria)	
Prioridad	Alta	
Actores involucrados	Consultor	
DESCRIPCIÓN DEL CASO DE USO		
Caso de uso para la gestión de los planes de acción.		
ENTRADAS		SALIDAS
<ul style="list-style-type: none"> Detalle de la acción a realizar sobre el plan de acción 		<ul style="list-style-type: none"> Retorno del estado de la acción realizada.
PRECONDICIONES		POSTCONDICIONES
<ul style="list-style-type: none"> El usuario pueda realizar las acciones. 		<ul style="list-style-type: none"> No aplica
DEPENDENCIA		EXTENSIÓN
<ul style="list-style-type: none"> No aplica 		<ul style="list-style-type: none"> No aplica
FLUJO NORMAL DE EVENTOS		
No.	Acción del actor	Respuesta del sistema
1	Selecciona la acción a realizar.	Formulario con los datos a ingresar del plan de acción.
2	Datos del plan de acción a crear o modificar, en caso que sea eliminar deberá confirmarse el evento.	Ventana con el resultado de la ejecución.
CAMINOS ALTERNATIVOS		

Cuadro 31. (Continuación)

No. actividad en flujo normal		No.	Acción del actor	Respuesta del sistema
Se presenta la alternativa	Se integra al flujo normal			
No aplica	No aplica		No aplica	No aplica
CAMINOS DE EXCEPCIÓN				
<ul style="list-style-type: none"> No aplica 				
CRITERIOS DE ACEPTACIÓN				
<ul style="list-style-type: none"> Los usuarios que tengan permiso para las acciones mencionadas las puedan realizar sin inconvenientes y que la información suministrada por el actor quede almacenada. 				
REQUERIMIENTOS NO FUNCIONALES ASOCIADOS				
No aplica				
SEGUIMIENTO AL CASO DE USO				
Fecha	Descripción	Gestor		
21/04/2014	Creación del caso de uso	Juan Ayala – Dario Acosta		

Fuente: Autor

El *cuadro 31*, detalla los privilegios de los actores sobre el plan de acción.

La gestión de normas, *cuadro 32*, se expanden las funciones representadas en la *figura 10*.

Cuadro 32. Especificación de Casos de uso UC-004

ID	UC-004	
Nombre	Gestión de Normas	
Tipo	Indispensable (Implementación obligatoria)	
Prioridad	Alta	
Actores involucrados	Administrador, Consultor y auditor.	
DESCRIPCIÓN DEL CASO DE USO		
Caso de uso para la gestión de normas, el auditor y el consultor solo podrán consultarlas, mientras que el administrador también puede agregar, actualizar y eliminarlas.		
ENTRADAS		SALIDAS
<ul style="list-style-type: none"> • Usuario actualmente logueado a la aplicación. 		<ul style="list-style-type: none"> • Retorno del estado de la acción realizada.
PRECONDICIONES		POSTCONDICIONES
<ul style="list-style-type: none"> • El usuario pueda realizar las acciones. 		<ul style="list-style-type: none"> • Actualizar las referencias de la norma en la aplicación.
DEPENDENCIA		EXTENSIÓN
<ul style="list-style-type: none"> • No aplica 		<ul style="list-style-type: none"> • No aplica
FLUJO NORMAL DE EVENTOS		
No.	Acción del actor	Respuesta del sistema
1	Selecciona la acción a realizar.	Formulario con los datos a ingresar del plan de acción.
2	Datos del plan de acción a crear o modificar, en caso que sea eliminar deberá confirmarse el evento.	Ventana con el resultado de la ejecución.

Cuadro 32. (Continuación)

CAMINOS ALTERNATIVOS				
No. actividad en flujo normal		No	Acción del actor	Respuesta del sistema
Se presenta la alternativa	Se integra al flujo normal			
No aplica	No aplica		No aplica	No aplica
CAMINOS DE EXCEPCIÓN				
<ul style="list-style-type: none"> La norma puede tener relaciones con otras tablas lo que no permite su eliminación. 				
CRITERIOS DE ACEPTACIÓN				
<ul style="list-style-type: none"> Los usuarios que tengan permiso para las acciones mencionadas las puedan realizar sin inconvenientes y que la información suministrada por el actor quede almacenada. 				
REQUERIMIENTOS NO FUNCIONALES ASOCIADOS				
No aplica				
SEGUIMIENTO AL CASO DE USO				
Fecha	Descripción		Gestor	
21/04/2014	Creación del caso de uso		Juan Ayala – Dario Acosta	

Fuente: Autor

El cuadro 33, muestra a profundidad las acciones que se pueden realizar sobre los controles y los comentarios.

Cuadro 33. Especificación de Casos de uso UC-005

ID	UC-005	
Nombre	Gestión de controles y comentarios	
Tipo	Indispensable (Implementación obligatoria)	
Prioridad	Alta	
Actores involucrados	Consultor y auditor.	
DESCRIPCIÓN DEL CASO DE USO		
Caso de uso para la gestión de los controles y comentarios.		
ENTRADAS		SALIDAS
<ul style="list-style-type: none"> • Usuario actualmente logueado a la aplicación. 		<ul style="list-style-type: none"> • Retorno del estado de la acción realizada.
PRECONDICIONES		POSTCONDICIONES
<ul style="list-style-type: none"> • El usuario pueda realizar las acciones. 		<ul style="list-style-type: none"> • No aplica
DEPENDENCIA		EXTENSIÓN
<ul style="list-style-type: none"> • No aplica 		<ul style="list-style-type: none"> • No aplica
FLUJO NORMAL DE EVENTOS		
N o.	Acción del actor	Respuesta del sistema
1	Selecciona la acción a realizar.	Formulario con los datos a ingresar del plan de acción.
2	Datos del plan de acción a crear o modificar, en caso que sea eliminar deberá confirmarse el evento.	Ventana con el resultado de la ejecución.
CAMINOS ALTERNATIVOS		

Cuadro 33. (Continuación)

No. actividad en flujo normal		No	Acción del actor	Respuesta del sistema
Se presenta la alternativa	Se integra al flujo normal			
No aplica	No aplica		No aplica	No aplica
CAMINOS DE EXCEPCIÓN				
<ul style="list-style-type: none"> El control puede tener relaciones con otras tablas lo que no permite su eliminación. 				
CRITERIOS DE ACEPTACIÓN				
<ul style="list-style-type: none"> Los usuarios que tengan permiso para las acciones mencionadas las puedan realizar sin inconvenientes y que la información suministrada por el actor quede almacenada. 				
REQUERIMIENTOS NO FUNCIONALES ASOCIADOS				
No aplica				
SEGUIMIENTO AL CASO DE USO				
Fecha	Descripción		Gestor	
21/04/2014	Creación del caso de uso		Juan Ayala – Dario Acosta	

Fuente: Autor

5.2.2.2.3 Análisis de Requisitos. En el *cuadro 34* se analizan los requisitos en función de su prioridad (Alta, Media y Baja), Valor que le da a la organización (entre más alto mayor valor 0-5), versión del requisito, tipo (funcional, no funcional) y la relación que tiene con los demás requisitos identificados.

Cuadro 34. Análisis de requisitos

No	Requisito	Prioridad	Valor	Versión	Tipo	Relación requisitos
1	El sistema debe estar orientado a la implementación de controles de diferentes normas de seguridad de la información.	Alta	5	1	Funcional	4, 5
2	El sistema debe evaluar de forma general el estado de implementación del SGSI teniendo en cuenta el porcentaje establecido por la persona encargada de esta tarea.	Alta	5	1	Funcional	3
3	El sistema debe reportar en todo momento el estado de implementación a través de un valor (0-100%) en la esquina superior derecha de la interfaz gráfica.	Media	5	1	Funcional	2
4	El sistema debe permitir consultar las diferentes normas implementadas a modo de consulta.	Baja	3	1	Funcional	5, 1
5	El sistema debe permitir agregar la licencia o comprobante de la norma comprada legalmente (Si aplica)	Baja	1	1	Funcional	4, 1
6	El sistema debe permitir acceder bajo 3 perfiles: auditor, consultor, y administrador.	Alta	4	1	Funcional	7, 8, 10, 11, 12

Cuadro 34. (Continuación)

No	Requisito	Prioridad	Valor	Versión	Tipo	Relación requisitos
7	Bajo ninguno de los perfiles se permitirá modificar los comentarios y observaciones realizados a los controles (Garantizar integridad)	Alta	3	1	Funcional	6, 8, 10, 11, 12
8	La aplicación debe funcionar simultáneamente para mínimo 5 usuarios sin verse afectado su rendimiento [Rendimiento]	Media	3	1	No funcional	6, 7, 10, 11, 12
9	La aplicación debe ser escalable en función de las características del servidor de aplicaciones y servidor de base de datos. [Rendimiento]	Media	2	1	No funcional	13, 14, 15
10	El sistema debe contar con un mecanismo de control de acceso, donde se limite el ingreso a los funcionarios autorizados. [Seguridad]	Alta	3	1	Funcional	6, 7, 8, 11, 12
11	La aplicación debe permitir la asignación de roles de acuerdo a las funciones del usuario [Seguridad]	Media	3	1	Funcional	6, 7, 8, 10, 12
12	Solo el rol administrador es el encargado de la generación de nuevos usuarios y la asignación de roles [Seguridad]	Media	3	1	Funcional	6, 7, 8, 10, 11

Cuadro 34. (Continuación)

13	El sistema debe poder ser instalado en la infraestructura existente de la compañía ALFCOM. [Implantación]	Media	5	1	No funcional	14, 15, 9
14	La aplicación debe ser ejecutada bajo plataformas Windows. [Implantación]	Media	5	1	Funcional	13, 15, 9
15	La aplicación debe poder ser accesible desde cualquier lugar y dispositivo de la compañía, incluso desde internet. [Disponibilidad]	Baja	4	1	Funcional	13, 14, 9

Fuente: Autor

5.2.2.2.4 Validación de Requisitos. Bajo esta actividad los usuarios confirman que los requisitos especificados en el catálogo de requisitos, así como los casos de uso, son válidos, consistentes y completos.

- Prototipo de interfaz. Se decide utilizar el esquema definido para la interfaz de usuario descrita a continuación en la *figura 14*.

Figura 14. Interfaz inicial propuesta para la aplicación.

Clausula	Objetivo de Control	Control	Acción	Fecha Inicial	Fecha Final	Cumplimiento	Mas [+]
5 Política de seguridad 0,0 %	5.1 Política de seguridad de información 0,0 %	5.1.1 Documentar política de seguridad de información 0.1%	Redactar la política de seguridad	27/10/2013	26/01/2014	1,0%	Plan de acción [+]
			Verificar por el comité	27/01/2014	26/02/2014	0,0%	Plan de acción [+]
			Aprobación de la alta gerencia	27/02/2014	15/03/2014	0,0%	Plan de acción [+]
		5.1.2 Revisión de la política de seguridad de la información	Revisión 1 de la política	01/03/2015	01/04/2015	0,0%	Plan de acción [+]
			Revisión 2 de la política	01/03/2016	01/04/2015	0,0%	Plan de acción [+]

Fuente: Autor

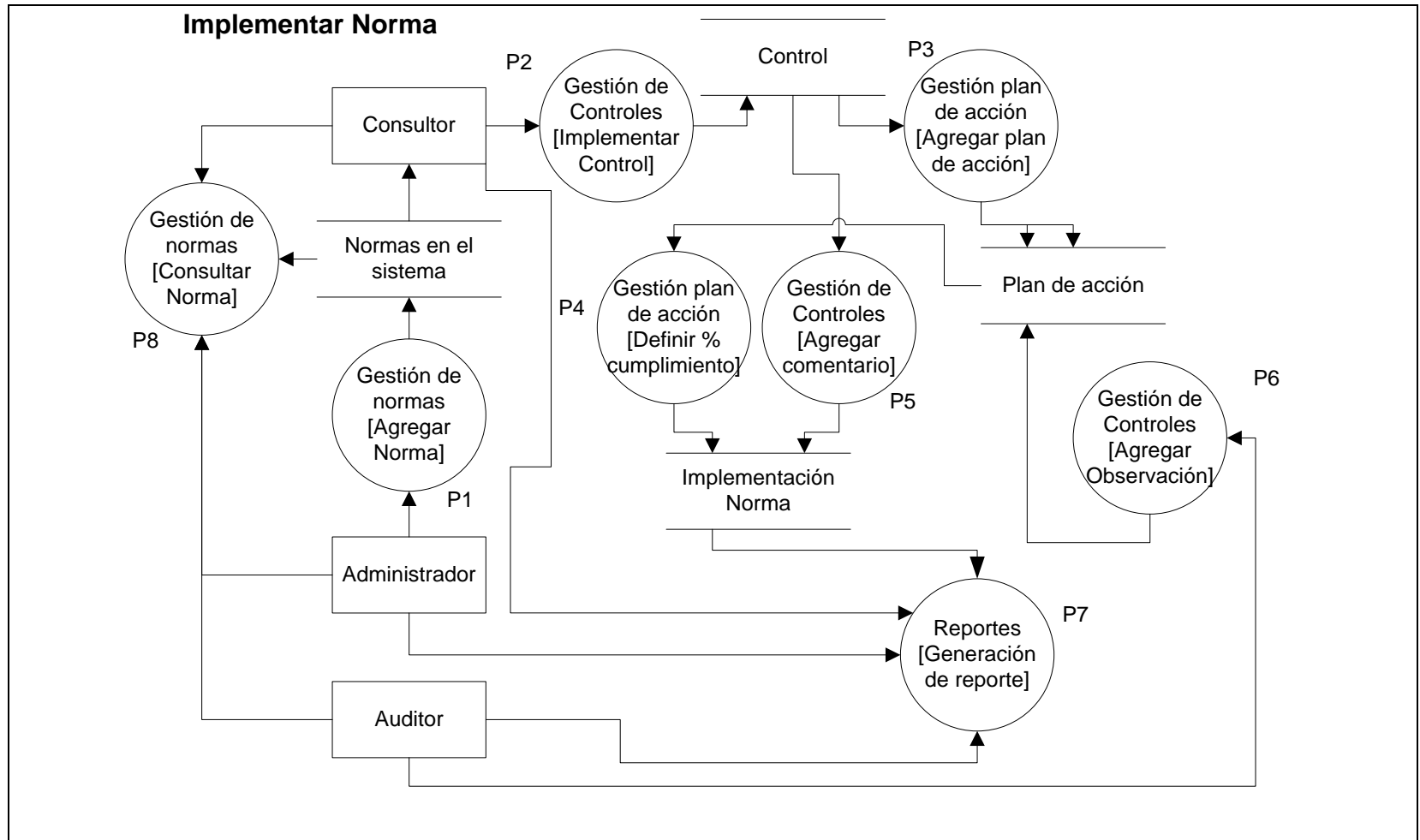
- Recorrido de casos de uso. Los desarrolladores del proyecto determinan realizar los casos de uso planteados con base en los requerimientos definidos en la tabla especificada en el numeral anterior - análisis de requisitos.
- Revisión periódica de requisitos. Con la siguiente tabla se busca hacer seguimiento sobre la implementación de los diferentes requerimientos a medida que se trabaja sobre el proyecto, teniendo en cuenta la opinión de los participantes del proyecto, validando su código y determinando posibles variaciones. En caso de decidir una variación es necesario registrarlo bajo el valor “no”, siendo necesario justificar la acción a través del Anexo 1, Requisitos Faltantes – Aplicación IIS.

5.2.2.3. Identificación de subsistemas de análisis. El objetivo de esta actividad es análisis del sistema de información llevando a cabo la descomposición del sistema en subsistemas.

5.2.2.3.1 Determinación de Subsistemas de Análisis. La descomposición en subsistemas descrita a continuación, está orientada a los procesos del negocio. A continuación se identifican las dependencias entre subsistemas analizando los elementos compartidos entre ellos o las interfaces entre subsistemas.

- Descripción de subsistemas de análisis

Figura 15. Diagrama de flujo de datos – Subsistemas de análisis

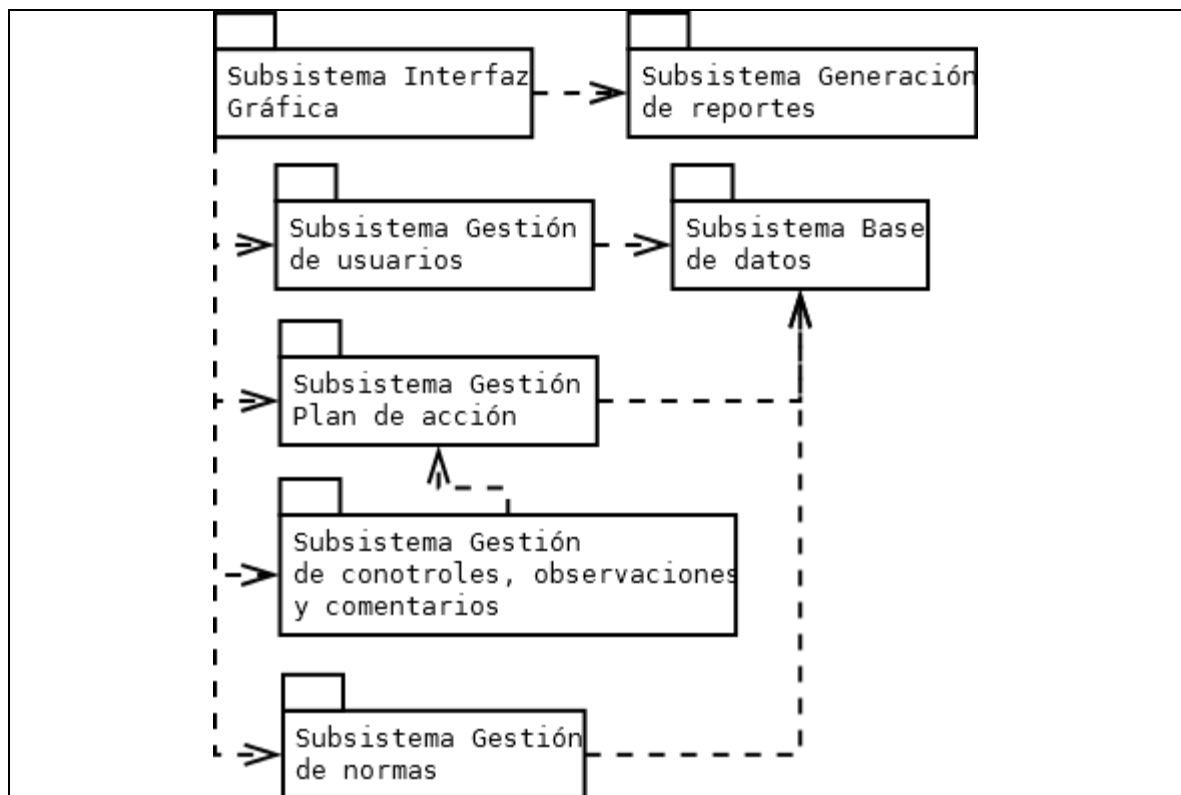


Fuente: Autor

La *figura 15* describe los pasos involucrados en el proceso principal de la implementación de una norma.

- Descripción de interfaces entre subsistemas

Figura 16. Diagrama de paquetes - Interfaces entre subsistemas

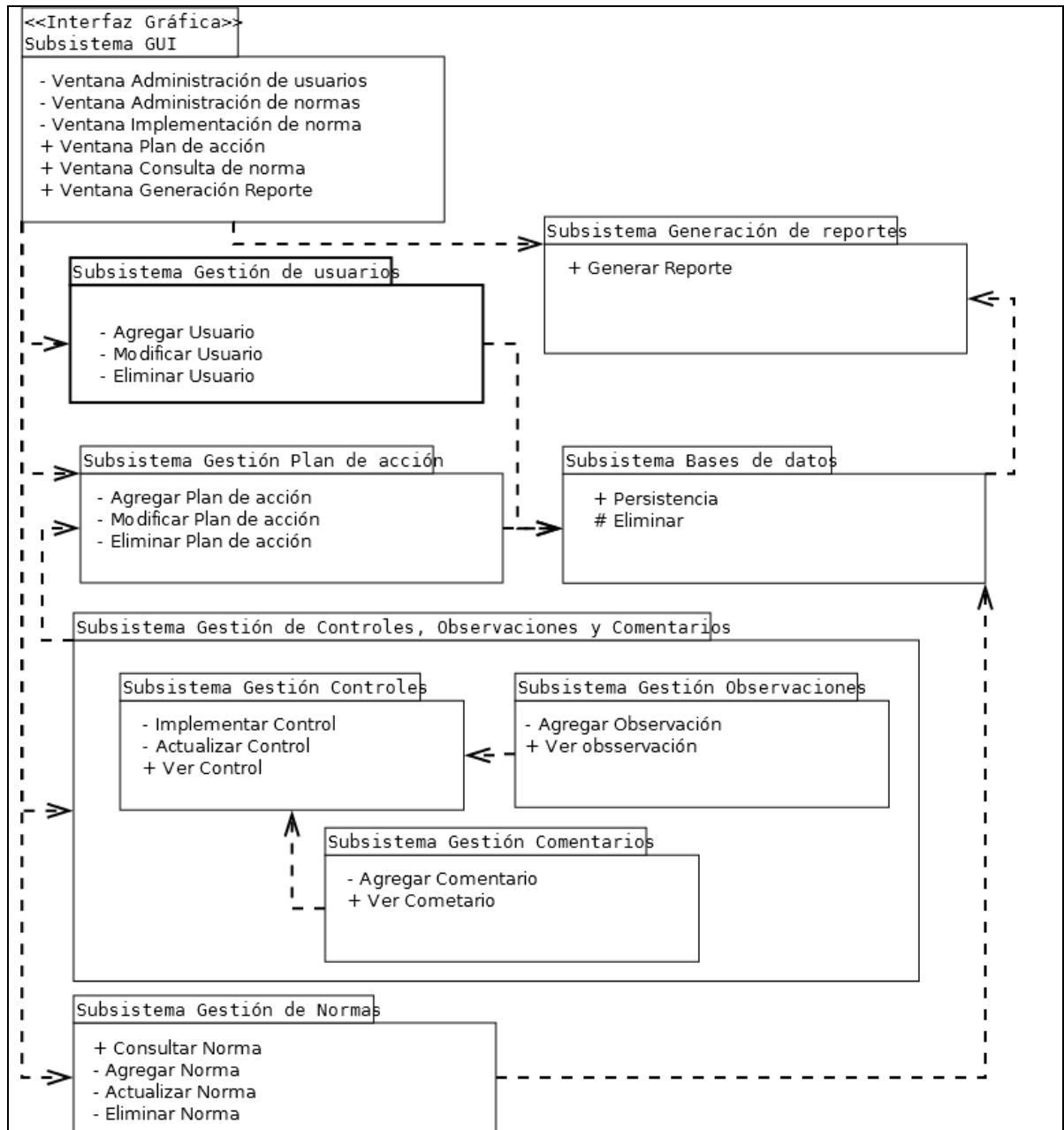


Fuente: Autor

La *figura 16* describe las interfaces existentes entre los subsistemas identificados de la aplicación.

5.2.2.3.2 Integración de Subsistemas de Análisis, *figura 17*. En esta tarea se coordina la elaboración de los distintos modelos de análisis de cada subsistema, asegurando la ausencia de duplicidad de elementos.

Figura 17. Diagrama de paquetes – Integración de subsistemas

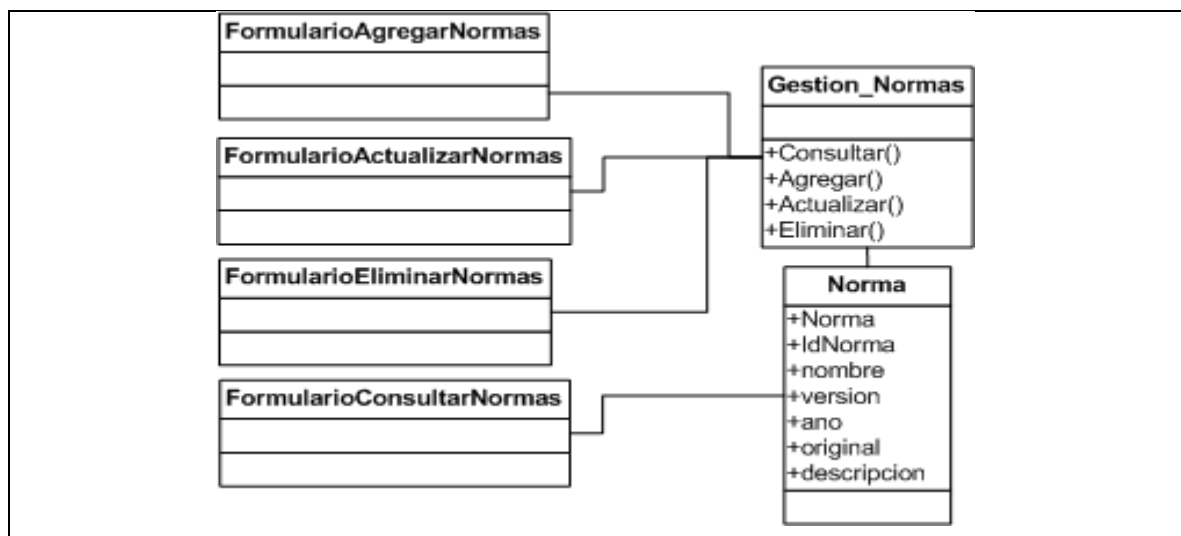


Fuente: Autor

5.2.2.4. Análisis de los casos de uso. Bajo esta actividad se identifican las clases cuyos objetos son necesarios para realizar casos de uso y describir su comportamiento mediante la interacción de dichos objetos.

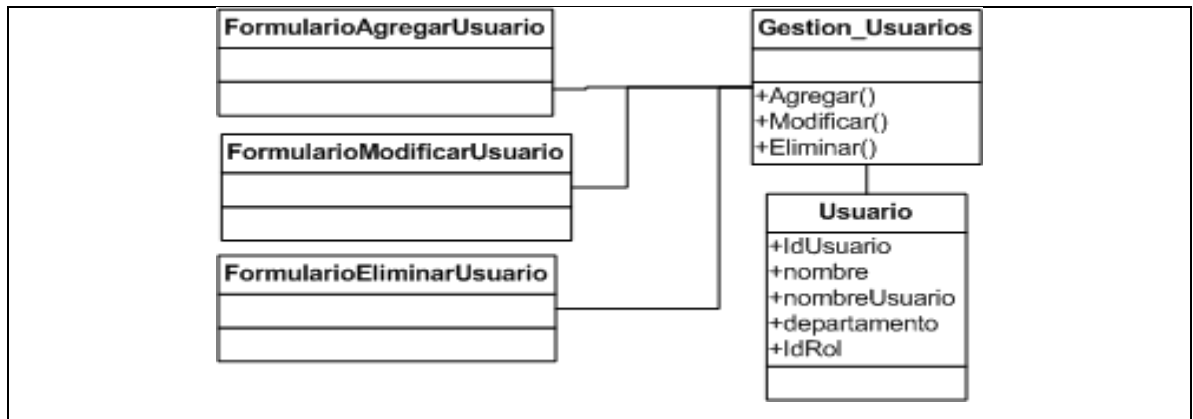
5.2.2.4.1 Identificación de Clases Asociadas a un Caso de Uso. En la *figura 18* se identifican los objetos necesarios para realizar los casos de uso, basándose en las especificaciones del mismo.

Figura 18. Identificación de clases – caso de uso Gestión de normas



Fuente: Autor

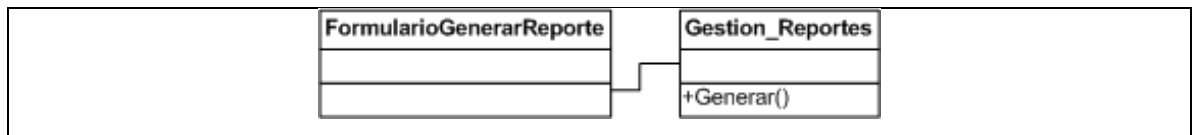
Figura 19. Identificación de clases – caso de uso Gestión de usuarios



Fuente: Autor

La *figura 19* describe las clases identificadas en el caso de uso de la gestión de usuarios.

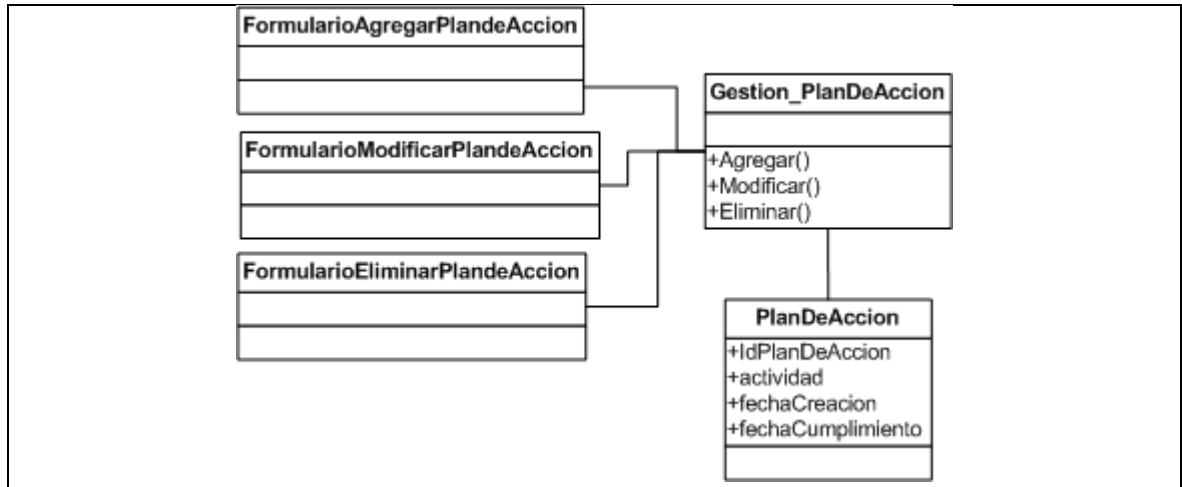
Figura 20. Identificación de clases – caso de uso Gestión Reportes



Fuente: Autor

La *figura 20* describe las clases identificadas en el caso de uso de la gestión de reportes.

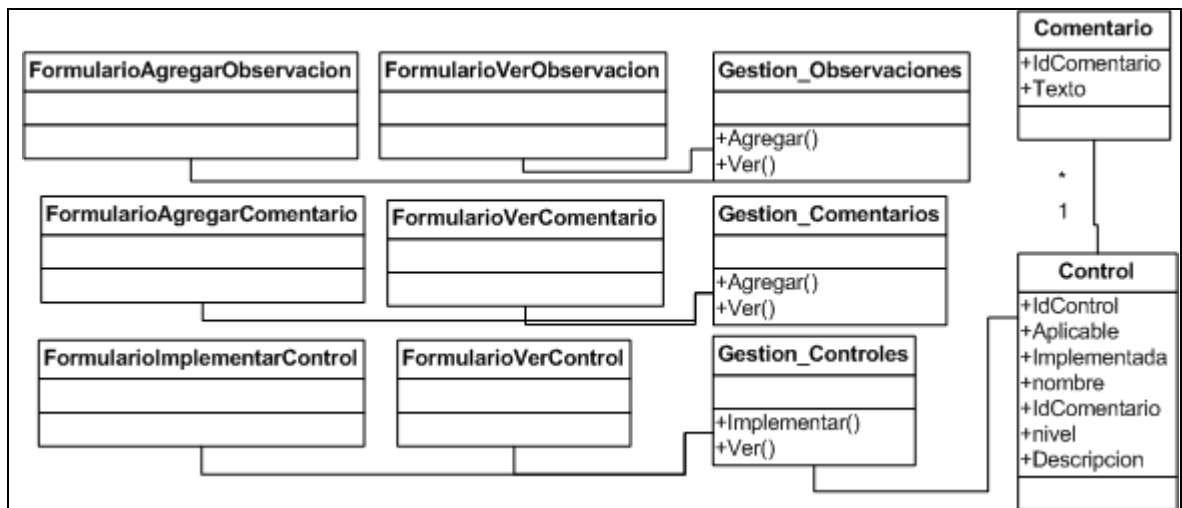
Figura 21. Identificación de clases – caso de uso Gestión Plan de Acción



Fuente: Autor

La *figura 21* describe las clases identificadas en el caso de uso de la gestión de planes de acción.

Figura 22. Identificación de clases – caso de uso Gestión Controles, Comentarios y Observaciones

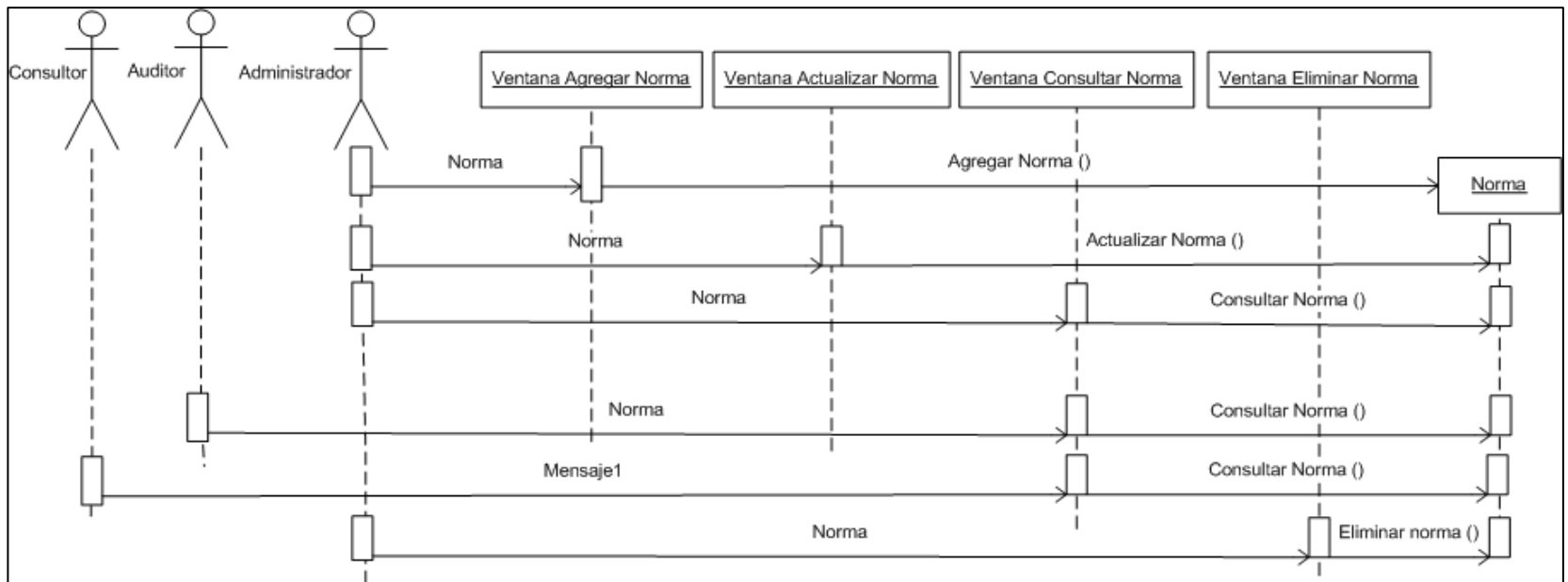


Fuente: Autor

La *figura 22* describe las clases identificadas en el caso de uso de la gestión de controles, comentarios y observaciones.

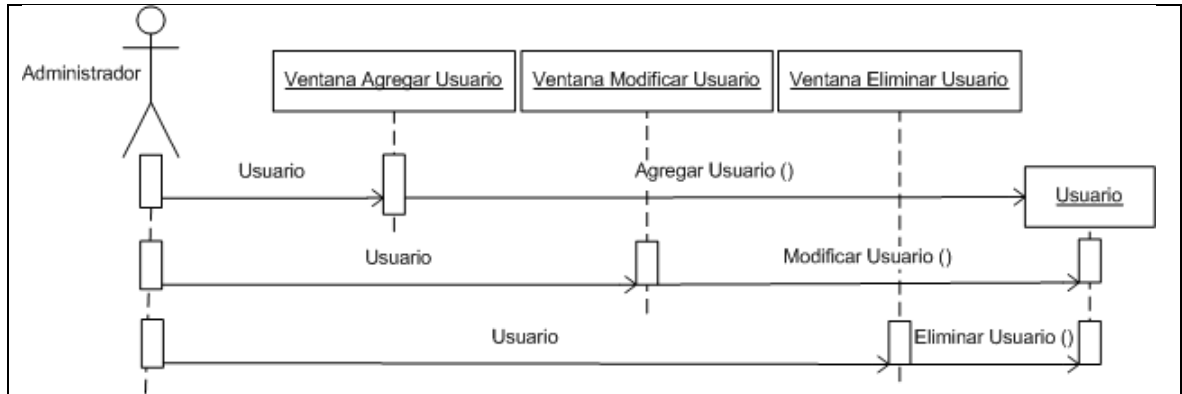
5.2.2.1. Descripción de la Interacción de Objetos. Por medio de esta tarea se describe la cooperación entre objetos utilizados para la realización de los casos de uso identificados anteriormente. Se representa a continuación los diagramas de interacción, *figura 23*, describiendo las instancias de los actores participantes, objetos, secuencia e intercambio de mensajes.

Figura 23. Diagrama de interacción - caso de uso Gestión de Normas



Fuente: Autor

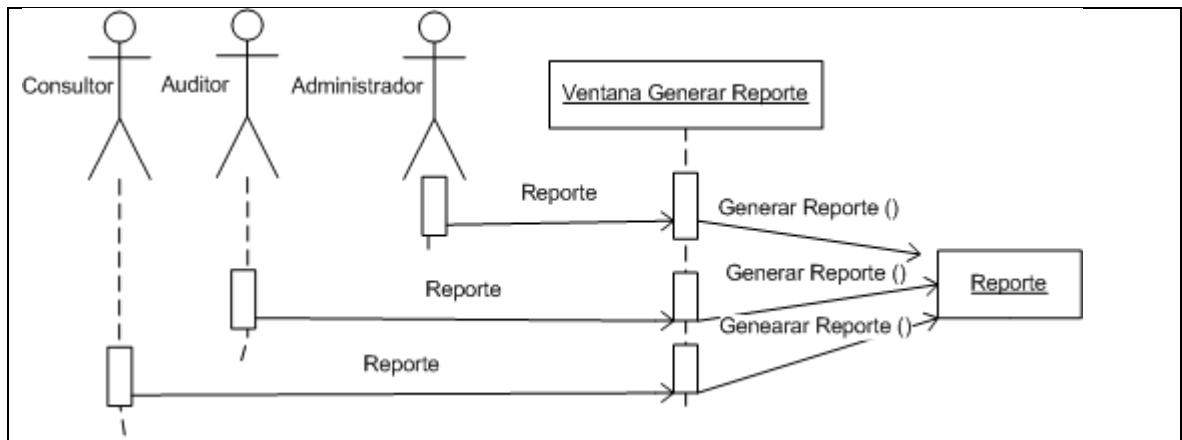
Figura 24. Diagrama de interacción - caso de uso Gestión de usuarios



Fuente: Autor

En la *figura 24*, se observa el diagrama de interacción para el caso de uso gestión de usuarios.

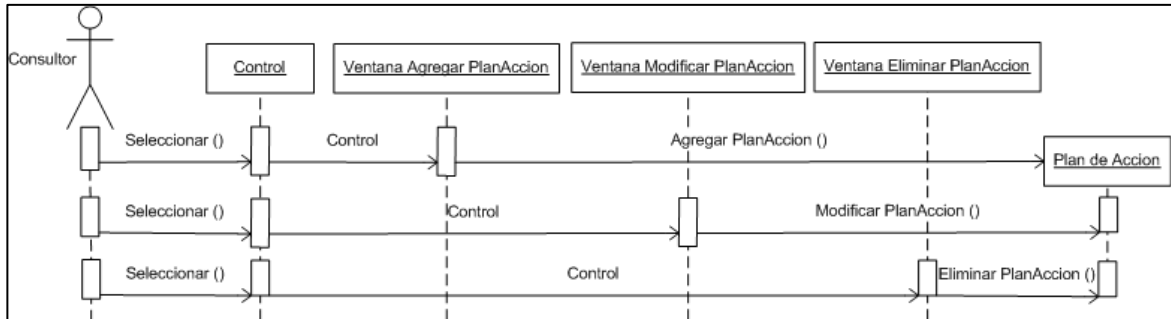
Figura 25. Diagrama de interacción - caso de uso Gestión Reportes



Fuente: Autor

El diagrama de interacción para el caso de uso gestión reportes se presenta en la *figura 25*.

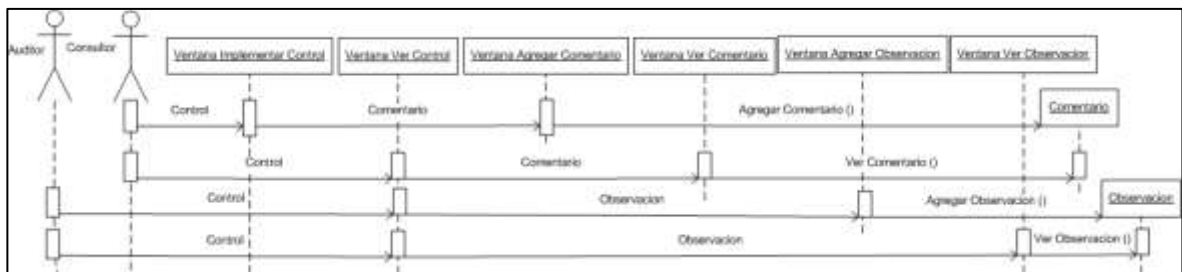
Figura 26. Diagrama de interacción - caso de uso Gestión Plan de Acción



Fuente: Autor

Para el plan de acción se describen los pasos a seguir por el consultor para la gestión de las mismas, figura 26.

Figura 27. Diagrama de interacción - caso de uso Gestión Controles, Comentarios y Observaciones



Fuente: Autor

Los usuarios con rol de consultor y auditor se reflejan en la figura 27.

5.2.2.2. Análisis de clases. Se describe cada una de las clases que ha surgido, identificando las responsabilidades que tienen asociadas, sus atributos, y las relaciones entre ellas.

5.2.2.2.1 Identificación de Responsabilidades y Atributos. A través de esta tarea se identifican las responsabilidades y atributos relevantes de las clases.

Cuadro 35. Responsabilidades y atributos de la clase FormularioAgregarNormas

CA 01	FormularioAgregarNormas	
Descripción		
Responsabilidades	Permitir al usuario administrador agregar normas al sistema	
Atributos	Descripción	Significado
	ID	Identificador de la norma
	Nombre	Nombre de la norma
	Version	Versión de la norma
	ano	Año de la norma
	Original	Licencia de la norma si aplica
	Descripción	Descripción de la norma
Operaciones	Descripción	Significado
	AgregarNorma()	Agregar norma en el sistema
	MostrarNorrna()	Muestra la norma del sistema

Fuente: Autor

Los atributos y las operaciones del formulario agregar normas, se representan en el *cuadro 35*.

Cuadro 36. Responsabilidades y atributos de la clase FormularioActualizarNormas

CA 02	FormularioActualizarNormas	
Descripción		
Responsabilidades	Permitir al usuario administrador actualizar normas del sistema	
Atributos	Descripción	Significado
	Nombre	Nombre de la norma
	Version	Versión de la norma
	ano	Año de la norma
	Original	Licencia de la norma si aplica
	Descripción	Descripción de la norma
Operaciones	Descripción	Significado
	MostrarNorrna()	Muestra la norma del sistema
	ActualizarNorma()	Actualiza la norma en el sistema

Fuente: Autor

En el *cuadro 36*, se presentan los atributos y operaciones del formulario actualizar normas.

Cuadro 37. Responsabilidades y atributos de la clase FormularioEliminarNormas

CA 03	FormularioEliminarNormas	
Descripción		
Responsabilidades	Permitir al usuario administrador eliminar normas del sistema	
Atributos	Descripción	Significado
Operaciones	Descripción	Significado
	MostrarNorma()	Muestra la norma del sistema
	EliminarNorma()	Elimina la norma en el sistema

Fuente: Autor

Los atributos y las operaciones del formulario eliminar normas, se representan en el *cuadro 37*.

Cuadro 38. Responsabilidades y atributos de la clase FormularioConsultarNormas

CA 04	FormularioConsultarNormas	
Descripción		
Responsabilidades	Permitir a los usuario del sistema listar las normas agregadas al sistema	
Atributos	Descripción	Significado
Operaciones	Descripción	Significado
	ConsultarNormas()	Muestra la lista de normas agregadas al sistema
	ActualizarVistaNormas()	Actualiza la lista de normas en el sistema.

Fuente: Autor

Los atributos y las operaciones del formulario consultar normas, se representan en el *cuadro 38*.

Cuadro 39. Responsabilidades y atributos de la clase FormularioAgregarUsuario

CA 05	FormularioAgregarUsuario	
Descripción		
Responsabilidades	Permitir al usuario administrador agregar usuarios al sistema	
Atributos	Descripción	Significado
	Rol	Usuario: Consultor, administrador o auditor
Operaciones	Descripción	Significado
	AgregarUsuario()	Crea el usuario dentro del sistema
	MostrarUsuario()	Mostrar el usuario dentro del sistema.

Fuente: Autor

Los atributos y las operaciones del formulario agregar usuario, se representan en el *cuadro 39*.

Cuadro 40. Responsabilidades y atributos de la clase FormularioModificarUsuario

CA 06	FormularioModificarUsuario	
Descripción		
Responsabilidades	Permitir al usuario administrador modificar usuarios del sistema	
Atributos	Descripción	Significado
	Rol	Usuario: Consultor, administrador o auditor
Operaciones	Descripción	Significado
	ModificarUsuario()	Modifica el usuario dentro del sistema
	MostrarUsuario()	Mostrar el usuario dentro del sistema.

Fuente: Autor

Los atributos y las operaciones del formulario modificar usuario, se representan en el *cuadro 50*.

Cuadro 41. Responsabilidades y atributos de la clase FormularioEliminarUsuario

CA 07	FormularioEliminarUsuario	
Descripción		
Responsabilidades	Permitir al usuario administrador eliminar usuarios del sistema	
Atributos	Descripción	Significado
	Rol	Usuario: Consultor, administrador o auditor
Operaciones	Descripción	Significado
	EliminarUsuario()	Eliminar el usuario dentro del sistema
	MostrarUsuario()	Mostrar el usuario dentro del sistema.

Fuente: Autor

Los atributos y las operaciones del formulario eliminar usuario, se representan en el *cuadro 41*.

Cuadro 42. Responsabilidades y atributos de la clase FormularioGenerarReporte

CA 08	FormularioGenerarReporte	
Descripción		
Responsabilidades	Permitir a los usuarios administrador, auditor y consultor generar reportes de la norma implementada.	
Atributos	Descripción	Significado
	Norma	Norma a la cual se le generará el reporte
Operaciones	Descripción	Significado
	GenerarReporte()	Genera el reporte de la implementación de la norma seleccionada

Fuente: Autor

Los atributos y las operaciones del formulario generar reporte, se representan en el *cuadro 42*.

Cuadro 43. Responsabilidades y atributos clase FormularioAgregarPlandeAccion

CA 09	FormularioAgregarPlandeAccion	
Descripción		
Responsabilidades	Permitir al usuario consultor crear un plan de acción	
Atributos	Descripción	Significado
	Control	Control de la norma al cual se le generará el plan de acción
Operaciones	Descripción	Significado
	AgregarPlandeAccion()	Crea el plan de acción del control seleccionado
	MostrarPlandeAccion()	Mostrar el plan de acción del control seleccionado

Fuente: Autor

Los atributos y las operaciones del formulario agregar plan de acción, se representan en el *cuadro 43*.

Cuadro 44. Responsabilidades y atributos clase FormularioModificarPlandeAccion

CA 10	FormularioModificarPlandeAccion	
Descripción		
Responsabilidades	Permitir al usuario consultor modificar un plan de acción	
Atributos	Descripción	Significado
	Control	Control de la norma al cual se le generará el plan de acción
Operaciones	Descripción	Significado
	ModificarPlandeAccion()	Modificar el plan de acción del control seleccionado
	MostrarPlandeAccion()	Mostrar el plan de acción del control seleccionado

Fuente: Autor

Los atributos y las operaciones del formulario modificar plan de acción, se representan en el *cuadro 44*.

Cuadro 45. Responsabilidades y atributos clase FormularioEliminarPlandeAccion

CA 11	FormularioEliminarPlandeAccion	
Descripción		
Responsabilidades	Permitir al usuario consultor eliminar un plan de acción	
Atributos	Descripción	Significado
	Control	Control de la norma al cual se le generará el plan de acción
Operaciones	Descripción	Significado
	EliminarPlandeAccion()	Eliminar el plan de acción del control seleccionado

Fuente: Autor

Los atributos y las operaciones del formulario eliminar plan de acción, se representan en el *cuadro 45*.

Cuadro 46. Responsabilidades y atributos clase FormularioAgregarObservación

CA 12	FormularioAgregarObservacion	
Descripción		
Responsabilidades	Permitir al auditor agregar una observación al control	
Atributos	Descripción	Significado
	Control	Control de la norma al cual se le generará la observación
Operaciones	Descripción	Significado
	AgregarObservacion()	Agregar observación al control seleccionado
	MostrarObservacion()	Mostrar observación del control seleccionado

Fuente: Autor

Los atributos y las operaciones del formulario agregar observación, se representan en el *cuadro 46*.

Cuadro 47. Responsabilidades y atributos de la clase FormularioVerObservación

CA 13	FormularioVerObservacion	
Descripción		
Responsabilidades	Permitir al auditor, consultor y administrador ver la observación del control	
Atributos	Descripción	Significado
	Control	Control de la norma al cual se le generará la observación
Operaciones	Descripción	Significado
	VerObservacion()	Ver observación del control seleccionado

Fuente: Autor

Los atributos y las operaciones del formulario ver observación, se representan en el *cuadro 47*.

Cuadro 48. Responsabilidades y atributos clase FormularioAgregarComentario

CA 14	FormularioAgregarComentario	
Descripción		
Responsabilidades	Permitir al consultor agregar un comentario al control	
Atributos	Descripción	Significado
	Control	Control de la norma al cual se le generará el comentario
Operaciones	Descripción	Significado
	AgregarComentario()	Agregar comentario al control seleccionado
	MostrarComentario()	Mostrar comentario del control seleccionado

Fuente: Autor

Los atributos y las operaciones del formulario agregar comentario, se representan en el *cuadro 48*.

Cuadro 49. Responsabilidades y atributos clase FormularioVerComentario

CA 15	FormularioVerComentario	
Descripción		
Responsabilidades	Permitir al consultor, administrador y auditor ver el comentario del control	
Atributos	Descripción	Significado
	Control	Control de la norma al cual se le generará el comentario
Operaciones	Descripción	Significado
	VerComentario()	Ver comentario del control seleccionado

Fuente: Autor

Los atributos y las operaciones del formulario ver comentario, se representan en el *cuadro 49*.

Cuadro 50. Responsabilidades y atributos de la clase FormularioImplementarControl

CA 16	FormularioImplementarControl	
Descripción		
Responsabilidades	Permitir al consultor implementar un control	
Atributos	Descripción	Significado
	Norma	Norma a la cual se le implementará el control
Operaciones	Descripción	Significado
	ImplementarControl()	Implementar control a la norma seleccionada
	MostrarControl()	Mostrar control de la norma seleccionada

Fuente: Autor

Los atributos y las operaciones del formulario implementar control, se representan en el *cuadro 50*.

Cuadro 51. Responsabilidades y atributos clase FormularioVerObservación

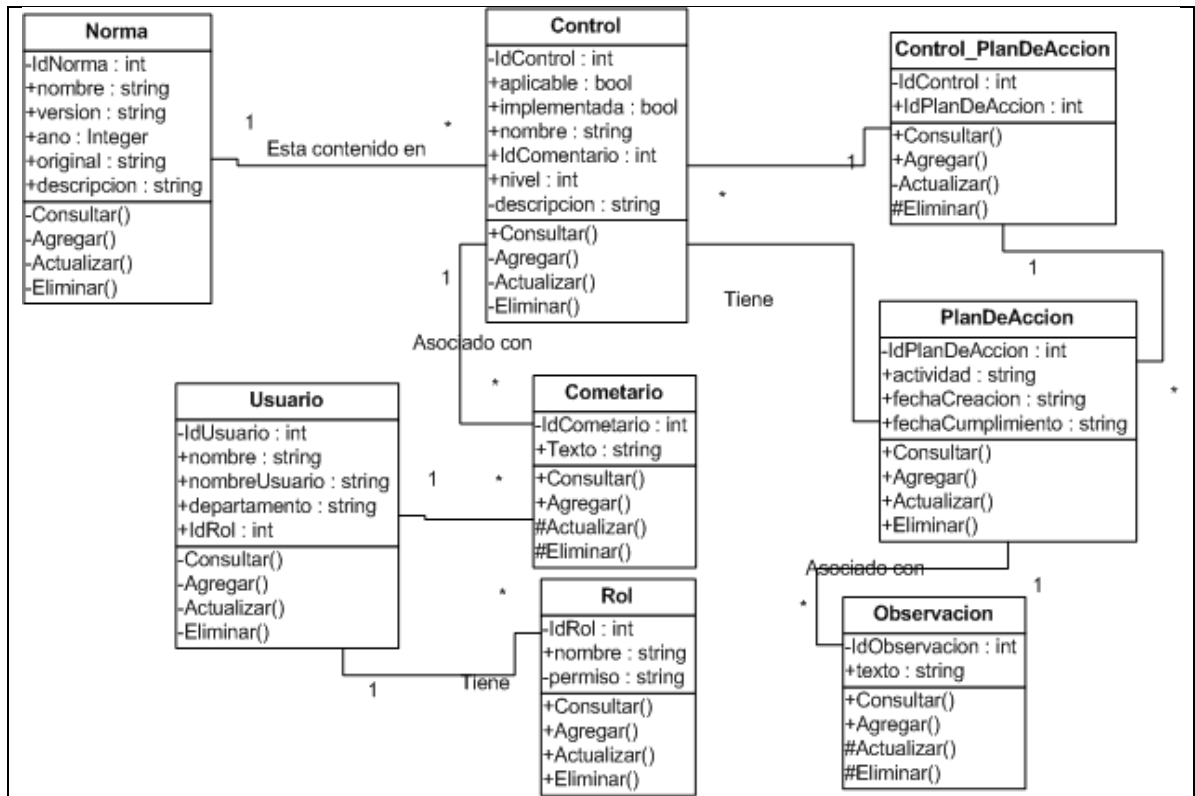
CA 17	FormularioVerControl	
Descripción		
Responsabilidades	Permitir al consultor, administrador y auditor ver la control implementado de la norma seleccionada	
Atributos	Descripción	Significado
	Norma	Norma a la cual se le implementará el control
Operaciones	Descripción	Significado
	VerControl()	Ver control de la norma seleccionada

Fuente: Autor

Los atributos y las operaciones del formulario ver observación, se representan en el *cuadro 51*.

5.2.2.2.2 Identificación de Asociaciones y Agregaciones de clases. En esta actividad se determina qué asociaciones existen entre las clases correspondientes. Estas asociaciones suelen corresponderse con expresiones verbales incluidas en las especificaciones.

Figura 28. Asociaciones y agregaciones de clases



Fuente: Autor

Las asociaciones y agregaciones de las clases se exploran en la *figura 28*.

5.2.2.2.3 Identificación de Generalizaciones. Dadas las especificaciones del sistema estas son reflejadas a través del diagrama de asociaciones y agregaciones de clases.

5.2.2.3. Definición de interfaces de usuario. Aquí se especifican las interfaces entre el sistema y el usuario: formatos de pantallas, diálogos, e informes, principalmente. El objetivo es realizar un análisis de los procesos del sistema de información en los que se requiere una interacción del usuario, con el fin de crear una interfaz que satisfaga todos los requisitos establecidos, teniendo en cuenta los diferentes perfiles a quienes va dirigido.

5.2.2.3.1 Especificación de Principios Generales de la Interfaz. A través de esta tarea se especifican los estándares, directrices y elementos generales a tener en cuenta en la definición de la interfaz de usuario.

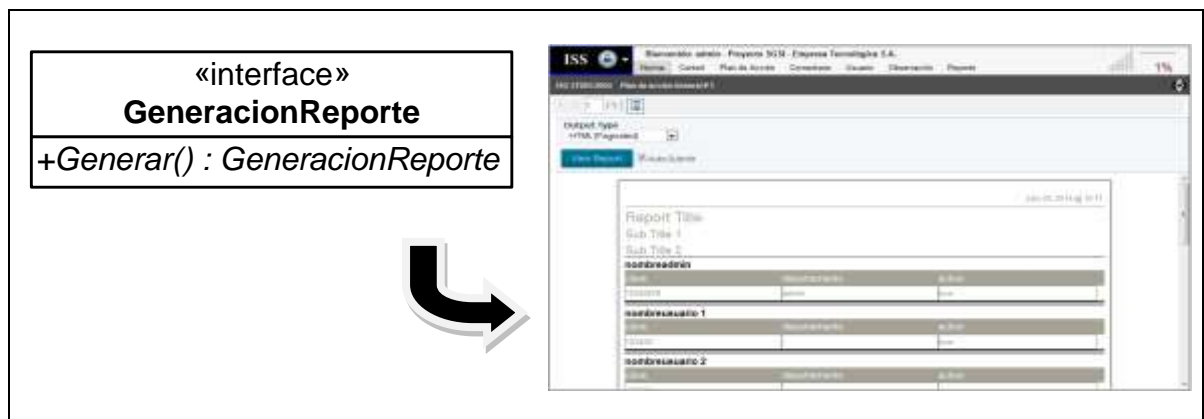
- Principios generales de la interfaz. Aplicación Information Security Status ISS
 1. Interfaz gráfica de acceso mediante teclado y ratón.
 2. Interfaz creada bajo lenguaje HTML, con adaptación de hojas de estilo en cascada CSS y lenguaje de programación ASPX.
 3. Los mensajes de error serán mostrados por una alerta ubicada en el centro de la pantalla.
 4. Los menús de la aplicación se encuentran distribuidos de forma horizontal en la parte superior de la interfaz de la aplicación.
 5. La presentación de ayudas sobre la interfaz se harán a través del atributo “alt” de las etiquetas HTML para mostrar texto alternativo.
 6. El esquema de la aplicación funcionara de forma estática en la pantalla con una resolución mínima de 1024X600px.

5.2.2.3.2 Especificación de Formatos Individuales de la Interfaz de Pantalla

- Interfaz Generación de reportes

La interfaz para generar reportes se muestra a continuación en la *figura 29*.

Figura 29. Interfaz generación de reportes

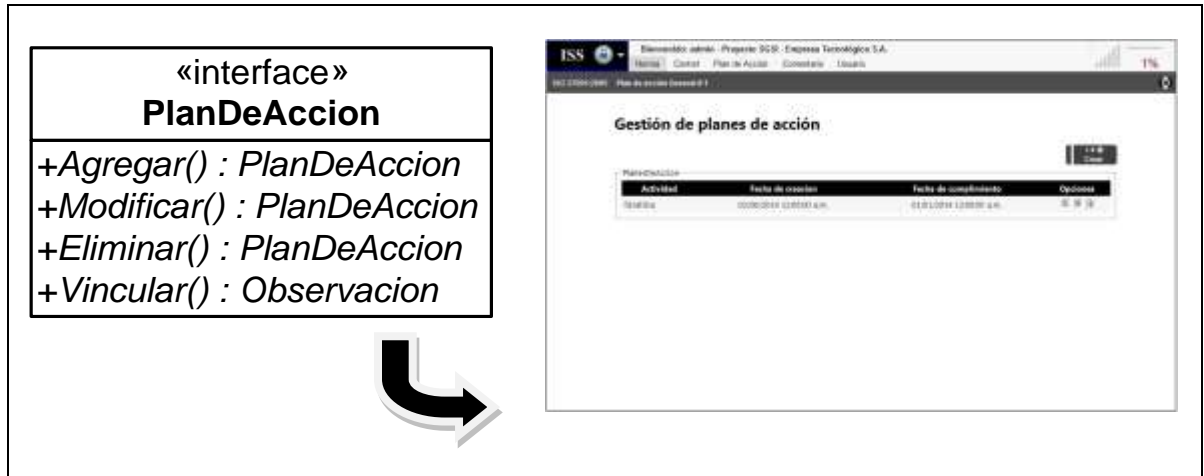


Fuente: Autor

- Interfaz Plan de acción

La interfaz del plan de acción se muestra a continuación en la *figura 30*.

Figura 30. Interfaz plan de acción

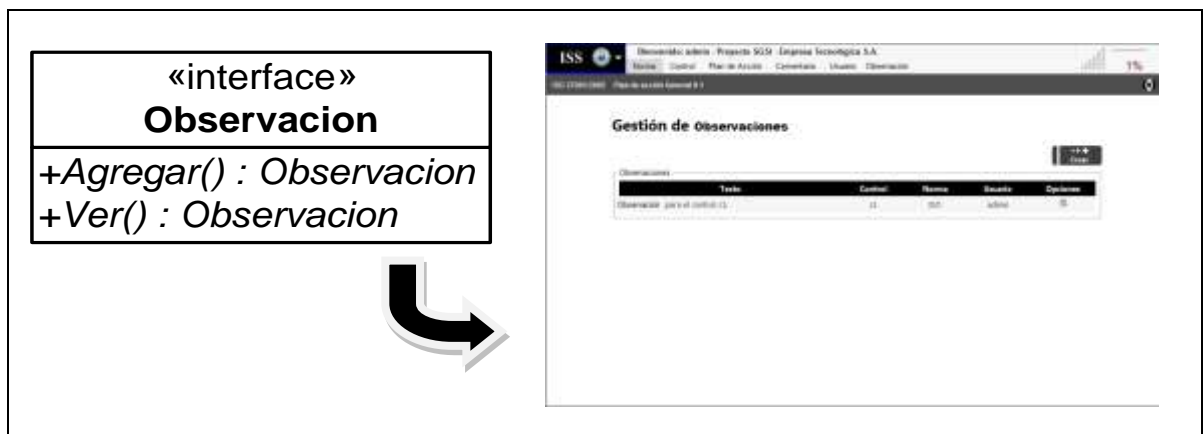


Fuente: Autor

- Interfaz Observaciones

La interfaz para la gestión de observaciones se muestra a continuación en la *figura 31*.

Figura 31. Interfaz observación

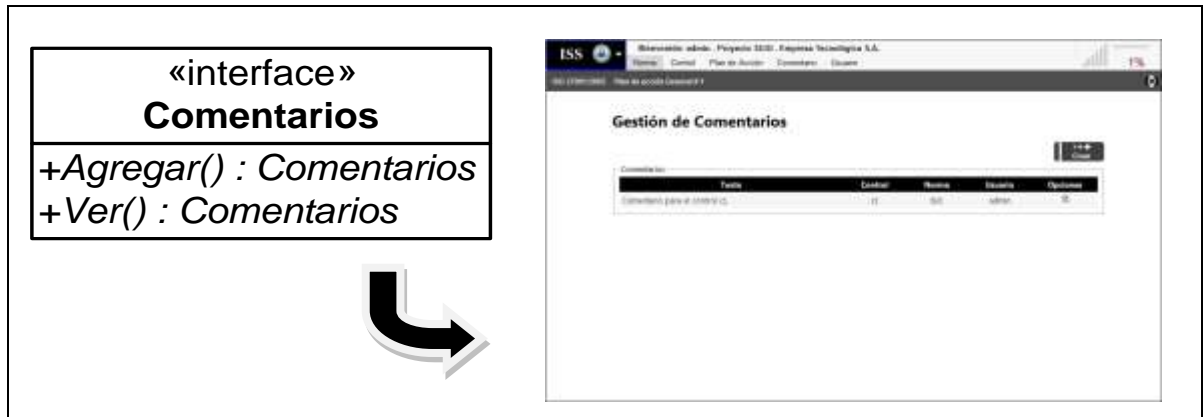


Fuente: Autor

- Interfaz Comentarios

La interfaz para la gestión de comentarios se muestra a continuación en la *figura 32*.

Figura 32. Interfaz comentarios

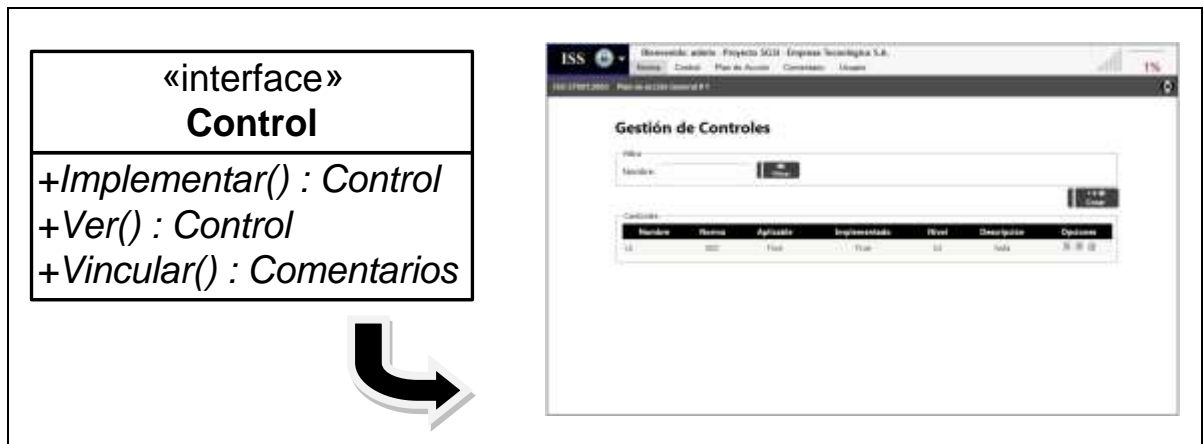


Fuente: Autor

- Interfaz Controles

La interfaz para generar la gestión de controles se muestra a continuación en la *figura 33*.

Figura 33. Interfaz controles



Fuente: Autor


- Interfaz normas

La interfaz para generar la gestión de normas se muestra a continuación en la figura 34.

Figura 34. Interfaz normas

**«interface»
Normas**

+Consultar() : Normas
+Agregar() : Normas
+Actualizar() : Normas
+Eliminar() : Normas
+Vincular() : Control



Fuente: Autor


- Interfaz Usuarios

La interfaz para generar la gestión de usuarios se muestra a continuación en la figura 35.

Figura 35. Interfaz usuarios

**«interface»
Usuarios**

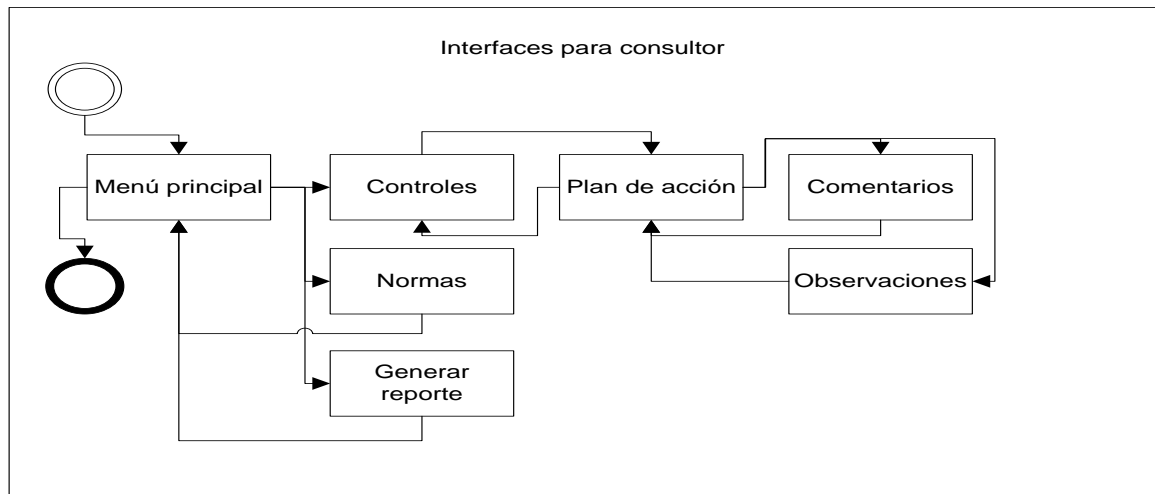
+Agregar() : Usuarios
+Modificar() : Usuarios
+Eliminar() : Usuarios



Fuente: Autor

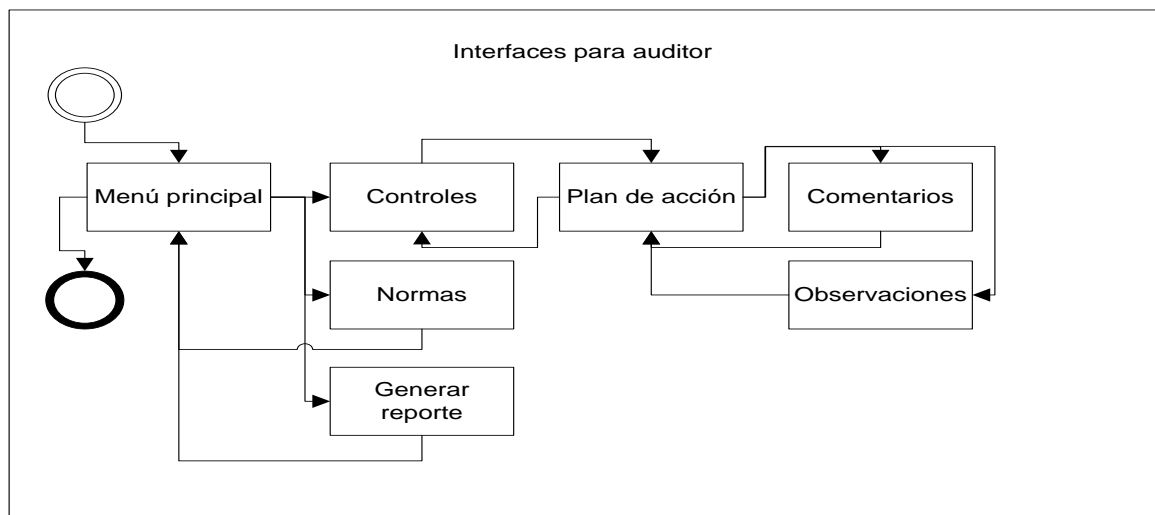
5.2.2.3.3 Especificación del Comportamiento Dinámico de la Interfaz. A continuación se definen los flujos entre los distintos formatos de interfaz de pantalla, y también dentro del propio formato a través de las figuras 36 a 38.

Figura 36. Comportamiento dinámico de interfaz de consultor



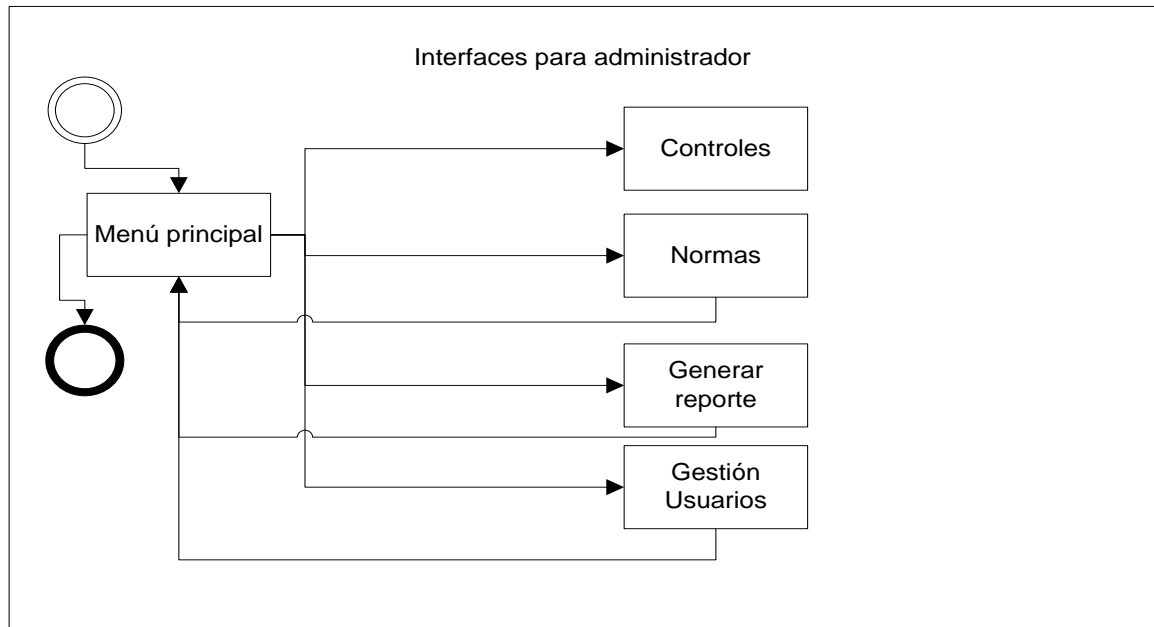
Fuente: Autor

Figura 37. Comportamiento dinámico de interfaz de auditor



Fuente: Autor

Figura 38. Comportamiento dinámico de interfaz de administrador



Fuente: Autor

5.2.2.4. Análisis de consistencia y especificación de requisitos. A través de esta tarea se busca garantizar la calidad de los distintos modelos generados en el proceso de Análisis del Sistema de Información, y asegurar que los usuarios y los Analistas tienen el mismo concepto del sistema. Para cumplir dicho objetivo, se llevan a cabo las siguientes acciones:

5.2.2.4.1 Verificación de los Modelos, se validan los modelos a través de cuadro 52.

Cuadro 52. Verificación de los modelos definidos

Verificación de modelos	Estado	Fecha
Especificación de Interfaz de Usuario	OK	02/06/2014
Modelo Lógico de Datos Normalizado	OK	02/06/2014
Modelo de Procesos	OK	02/06/2014
Modelo de Casos de Uso	OK	02/06/2014
Especificación de Casos de Uso	OK	02/06/2014
Modelo de Clases de Análisis	OK	02/06/2014
Comportamiento de Clases de Análisis	OK	02/06/2014
Análisis de la Realización de los Casos de Uso	OK	02/06/2014
Descripción de Subsistemas de Análisis	OK	02/06/2014
Descripción Interfaces entre Subsistemas	OK	02/06/2014

Fuente: Autor

5.2.2.4.2 Análisis de Consistencia entre Modelos

Se describen los roles que dentro de la matriz pueden interactuar con los Diagramas dinámicos (Formularios Principales de la aplicación) con relación a las clases utilizadas en el sistema, el cual es descrito en el cuadro 53.

La notación utilizada en la gráfica a continuación se expone a cómo: C equivale al “Rol Consultor”, M equivale a “Rol Administrador” y A equivale a “Rol Auditor”.

Cuadro 53. Consistencia Modelo de clases – Diagramas dinámicos

Modelo de clases	Diagramas Dinámicos			
	Agregar	Ver	Modificar	Eliminar
Norma	M	MCA	M	M
Control	C	MCA	C	C
Plan de acción	C	CA	C	C
Comentarios	C	CA		
Observaciones	A	CA		
Usuarios	M	MCA	M	M

Fuente: Autor

Se describen las funciones permitidas dentro de cada una de las interfaces de la aplicación con relación a las clases utilizadas en el sistema a través del cuadro 54.

La notación utilizada en la gráfica a continuación se expone a cómo: V equivale a la función “Crear()”, E equivale a la función “Editar()”, G equivale a la función “Gestion()” y V equivale a la función “Ver()”.

Cuadro 54. Consistencia Modelo de clases – Interfaces de usuario

Modelo de clases	Interfaz de usuario					
	Plan de acción	Observación	Comentarios	Control	Normas	Usuarios
Norma				V	CEGV	
Control	V	V	V	CEGV	V	
Plan de acción	CEGV			VE		

Cuadro 54. (Continuación)

Modelo de clases	Interfaz de usuario					
	Plan de acción	Observación	Comentarios	Control	Normas	Usuarios
Comentarios	V		CV	V		
Observaciones	V	CV		V		
Usuarios						CEGV

Fuente: Autor

5.2.2.4.3 Validación de los Modelos A continuación se describen los requisitos y su prototipo de interfaz de usuario GUI asociado que buscan validar los modelos de clases definidos.

Requisito 1: El sistema está orientado a la implementación de controles de diferentes normas de seguridad de la información como se observa en las figuras 39 y 40.

Figura 39. Prototipo interfaz- Implementación de normas en el sistema.



Fuente: Autor

Figura 40. Prototipo interfaz - Implementación de controles en el sistema.



Fuente: Autor

Requisito 2: El sistema evalúa de forma general el estado de implementación del SGSI teniendo en cuenta el porcentaje establecido por la persona encargada de esta tarea, como se observa en la figura 41.

Figura 41. Prototipo interfaz - Porcentaje de cumplimiento establecido por analista de seguridad.



Fuente: Autor

Requisito 3: El sistema reporta en todo momento el estado de implementación a través de un valor (0-100%) en la esquina superior derecha de la interfaz gráfica como se observa en la figura 42.

Figura 42. Prototipo interfaz - Porcentaje de implementación general de la norma.



Fuente: Autor

Requisito 4: El sistema permite consultar las diferentes normas implementadas a modo consulta, el cual es descrito en la figura 43.

Figura 43. Prototipo interfaz – Consulta de normas.



Fuente: Autor

Requisito 5: El sistema permite agregar la licencia o comprobante de la norma comprada legalmente (Si aplica) como se observa en la figura 44.

Figura 44. Prototipo interfaz – licenciamiento de norma.



Fuente: Autor

Requisito 6: El sistema permite acceder bajo 3 perfiles: auditor, consultor, y administrador como se observa en la figura 45.

Figura 45. Prototipo interfaz – Perfiles de acceso a la aplicación.



Fuente: Autor

Requisito 7: Bajo ninguno de los perfiles se permitirá modificar los comentarios y observaciones realizados a los controles (Garantiza integridad) a través de la figura 46.

Figura 46. Prototipo interfaz – Controles y comentarios en modo lectura.



Fuente: Autor

5.2.2.4.4 Elaboración de la Especificación de Requisitos Software (ERS). La siguiente especificación de requisitos del sistema toma como referencia la norma IEEE 830 de 1998 de la siguiente manera²⁷.

A continuación se describen los requisitos de software clasificados bajo las siguientes categorías, estos son representados en el cuadro 55:

- Interfaces Externas: requisitos que afectan a la interfaz del usuario con otros sistemas (hardware y software).

²⁷ Universidad Complutense. IEEE 830 de 1998 - Especificaciones De Los Requisitos Del Software. En: informatica.ucm.es. [en línea]. No. 1 (2014) [consultado 04 sep. 2014]. Disponible en <<https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>>.

- Funciones: acciones de la aplicación orientadas a los requisitos de la aplicación y los requisitos de los usuarios Auditor, Administrador y Consultor.
- Rendimiento: establecen las capacidades en cuanto a rendimiento que debe cumplir la aplicación.
- Diseño: Condiciones que restringen el desarrollo de la aplicación (limitaciones de hardware, aplicación, etc)
- Seguridad: Determina las condiciones de seguridad planteadas para la aplicación y el sistema en general.
- Disponibilidad: Orientado a las necesidades del negocio en cuanto a acceso a la aplicación.

Cuadro 55. Especificación de requisitos de software

Requisito	Tipo	Descripción
1	Interfaces Externas	La aplicación se contacta con un servidor web y un servidor de bases de datos para el manejo de la información.
2	Funciones	El sistema debe estar orientado a la implementación de controles de diferentes normas de seguridad de la información
3	Funciones	El sistema debe evaluar de forma general el estado de implementación del SGSI teniendo en cuenta el porcentaje establecido por la persona encargada de esta tarea.
4	Funciones	El sistema debe reportar en todo momento el estado de implementación a través de un valor (0-100%) en la esquina superior derecha de la interfaz gráfica.
5	Funciones	El sistema debe permitir consultar las diferentes normas implementadas en modo de lectura.
6	Funciones	El sistema debe permitir agregar la licencia o comprobante de la norma comprada legalmente (Si aplica).

Cuadro 55. (Continuación)

Requisito	Tipo	Descripción
7	Funciones	El sistema debe permitir acceder bajo 3 perfiles: auditor, consultor, y administrador.
8	Funciones	El usuario Administrador tiene la capacidad de Agregar, Modificar y Eliminar usuarios a la aplicación
9	Funciones	El usuario Administrador tiene la capacidad de generar reportes de la implementación de normas de la aplicación.
10	Funciones	El usuario Administrador tiene la capacidad de Consultar, Agregar, Actualizar y Eliminar normas de la aplicación
11	Funciones	El usuario Auditor tiene la capacidad de generar reportes de la implementación de normas de la aplicación.
12	Funciones	El usuario Auditor tiene la capacidad de consultar las normas agregadas en la aplicación.
13	Funciones	El usuario Auditor tiene la capacidad de Ver controles.
14	Funciones	El usuario Auditor tiene la capacidad de Agregar y Ver observaciones
15	Funciones	El usuario Auditor tiene la capacidad de Ver comentarios
16	Funciones	El usuario Consultor tiene la capacidad de generar reportes de la implementación de normas de la aplicación.
17	Funciones	El usuario Consultor tiene la capacidad de Agregar, Modificar y Eliminar planes de acción.

Cuadro 55. (Continuación)

Requisito	Tipo	Descripción
18	Funciones	El usuario Consultor tiene la capacidad de consultar las normas agregadas en la aplicación.
19	Funciones	El usuario Consultor tiene la capacidad de Implementar, Actualizar, y Ver controles.
20	Funciones	El usuario Consultor tiene la capacidad Agregar y Ver Comentarios.
21	Funciones	El usuario Consultor tiene la capacidad de Ver Observaciones.
22	Rendimiento	La aplicación debe funcionar simultáneamente para mínimo 5 usuarios sin verse afectado su rendimiento.
23	Rendimiento	La aplicación debe ser escalable en función de las características del servidor de aplicaciones y servidor de base de datos.
24	Diseño	Interfaz gráfica de acceso mediante teclado y ratón.
25	Diseño	Interfaz creada bajo lenguaje HTML, con adaptación de hojas de estilo en cascada CSS y lenguaje de programación ASPX.
26	Diseño	Los mensajes de error serán mostrados por una alerta ubicada en el centro de la pantalla.
27	Diseño	Los menús de la aplicación se encuentran distribuidos de forma horizontal en la parte superior de la interfaz de la aplicación.
28	Diseño	La presentación de ayudas sobre la interfaz se hará a través del atributo "alt" de las etiquetas HTML para mostrar texto alternativo.

Cuadro 55. (Continuación)

Requisito	Tipo	Descripción
29	Diseño	El esquema de la aplicación funcionara de forma estática en la pantalla con una resolución mínima de 1024X600px.
30	Diseño	La aplicación debe ser ejecutada bajo la arquitectura cliente - servidor.
31	Diseño	La aplicación debe ser desarrollada utilizando las herramientas de desarrollo e infraestructura utilizada por la empresa ALFCOM S.A.
32	Diseño	La aplicación debe ser ejecutada bajo plataformas Windows.
33	Seguridad	El sistema debe contar con un mecanismo de control de acceso, donde se limite el ingreso a los funcionarios autorizados.
34	Seguridad	La aplicación debe permitir la asignación de roles de acuerdo a las funciones del usuario.
35	Seguridad	Solo el rol administrador es el encargado de la generación de nuevos usuarios y la asignación de roles.
36	Seguridad	El sistema debe validar las entradas de los campos de entrada de la aplicación, mitigando riesgos de seguridad informáticos asociados.
37	Disponibilidad	La aplicación debe poder ser accesible desde cualquier lugar y dispositivo de la compañía, incluso desde internet.

Fuente: Autor

5.2.2.5. Especificación del plan de pruebas. A continuación se inicia la definición del plan de pruebas, el cual sirve como guía para la realización de las pruebas, y permite verificar que el sistema de información cumple las necesidades establecidas por el usuario, con las debidas garantías de calidad.

5.2.2.5.1 Definición del Alcance de las Pruebas. El plan busca establecer las pruebas a realizar en cada uno de los módulos con el objetivo de verificar su funcionalidad de la aplicación IIS (Internet Information Services). Se realizarán las pruebas clasificadas bajo las siguientes categorías:

- Criterios de severidad de las pruebas. Se reportarán los siguientes niveles de severidad a través del cuadro 56 al momento de realizar las pruebas:

Cuadro 56. Criterios de severidad de las pruebas

Nivel de Severidad	Descripción	Ejemplo
1	Se detiene la ejecución de la aplicación y no es posible volver utilizarla.	Error crítico que impide ejecutar nuevamente la aplicación.
2	La función a realizar no es ejecutada y la aplicación es detenida.	La función ejecutada genera errores generando que la aplicación se cierre. La aplicación puede ser abierta nuevamente.
3	Las funciones no pueden ser ejecutadas, pero la aplicación sigue en funcionamiento.	Se agregan datos a la aplicación y se reciben mensajes de error desconocidos. La aplicación sigue funcionando.
4	Modificación Menor	Errores menores de uso de la aplicación, presentación inconsistente de interfaz gráfica, etc.

Fuente: Autor

- Pruebas funcionales. Evalúan los casos de uso definidos para el funcionamiento de la aplicación, las cuales son descritas de la figura 57 a 74.

Cuadro 57. Prueba funcional – Agregar Usuario

Objetivo de la prueba	Validar la correcta adición de un nuevo usuario sobre el sistema.
Estrategia	<ul style="list-style-type: none"> Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos. Se ingresarán datos válidos y no válidos analizando su comportamiento.
Herramientas necesarias	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> El usuario es agregado dentro de la aplicación con base en su rol (Administrador, Auditor o Consultor), y actualizado en la base de datos. Si la información ingresada en la creación del usuario no es válida, la aplicación debe generar un mensaje de error notificando el problema.
Consideraciones especiales	<ul style="list-style-type: none"> Solo puede ser realizado desde el rol de usuario Administrador.

Fuente: Autor

Cuadro 58. Prueba funcional – Modificar Usuario

Objetivo de la prueba	Validar la correcta modificación de un usuario ya inscrito en el sistema.
Estrategia	<ul style="list-style-type: none"> Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos. Se ingresarán datos válidos y no válidos analizando su comportamiento.
Herramientas necesarias	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> El usuario es modificado dentro de la aplicación y actualizado en la base de datos. Si la información ingresada en la modificación del usuario no es válida, la aplicación debe generar un mensaje de error notificando el problema.
Consideraciones especiales	<ul style="list-style-type: none"> Solo puede ser realizado desde el rol de usuario Administrador.

Fuente: Autor

Cuadro 59. Prueba funcional – Eliminar Usuario

Objetivo de la prueba	Validar la correcta modificación de un usuario ya inscrito en el sistema.
Estrategia	<ul style="list-style-type: none"> • Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos. • Se ingresarán datos válidos y no válidos analizando su comportamiento.
Herramientas necesarias	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> • Se elimina el usuario de la aplicación y la base de datos.
Consideraciones especiales	<ul style="list-style-type: none"> • Solo puede ser realizado desde el rol de usuario Administrador.

Fuente: Autor

Cuadro 60. Prueba funcional – Generar reporte

Objetivo de la prueba	Validar la correcta generación del reporte de implementación de una norma.
Estrategia	<ul style="list-style-type: none"> • Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos. • Se ingresarán datos válidos y no válidos analizando su comportamiento.
Herramientas necesarias	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> • Se genera el reporte de la implementación de la norma seleccionada.
Consideraciones especiales	<ul style="list-style-type: none"> • Ninguna.

Fuente: Autor

Cuadro 61. Prueba funcional – Agregar plan de acción

Objetivo de la prueba	Validar la correcta adición de un nuevo plan de acción asociado a un control.
Estrategia	<ul style="list-style-type: none"> • Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos. • Se ingresarán datos válidos y no válidos analizando su comportamiento.
Herramientas	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> • Se incluye un nuevo plan de acción asociado a un control de la norma implementada. • Si la información ingresada en la modificación del usuario no es válida, la aplicación debe generar un mensaje de error notificando el problema.
Consideraciones especiales	<ul style="list-style-type: none"> • Solo puede ser realizado desde el rol de usuario consultor.

Fuente: Autor

Cuadro 62. Prueba funcional – Modificar plan de acción

Objetivo de la prueba	Validar la correcta Modificación de un plan de acción asociado a un control.
Estrategia	<ul style="list-style-type: none"> • Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos. • Se ingresarán datos válidos y no válidos analizando su comportamiento.
Herramientas	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> • Se actualiza un plan de acción existente, asociado a un control de la norma implementada. • Si la información ingresada en la modificación del usuario no es válida, la aplicación debe generar un mensaje de error notificando el problema.
Consideraciones especiales	<ul style="list-style-type: none"> • Solo puede ser realizado desde el rol de usuario consultor.

Fuente: Autor

Cuadro 63. Prueba funcional – Eliminar plan de acción

Objetivo de la prueba	Validar la correcta Eliminación de un plan de acción asociado a un control.
Estrategia	<ul style="list-style-type: none"> Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos.
Herramientas necesarias	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> Se elimina un plan de acción existente, asociado a un control de la norma implementada.
Consideraciones especiales	<ul style="list-style-type: none"> Solo puede ser realizado desde el rol de usuario consultor.

Fuente: Autor

Cuadro 64. Prueba funcional – Agregar Norma

Objetivo de la prueba	Validar la correcta Adición de una norma al sistema.
Estrategia	<ul style="list-style-type: none"> Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos. Se ingresarán datos válidos y no válidos analizando su comportamiento.
Herramientas necesarias	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> Se Agrega una nueva norma al sistema. Si la información ingresada en la modificación del usuario no es válida, la aplicación debe generar un mensaje de error notificando el problema.
Consideraciones especiales	<ul style="list-style-type: none"> Solo puede ser realizado desde el perfil de usuario Administrador.

Fuente: Autor

Cuadro 65. Prueba funcional – Actualizar Norma

Objetivo de la prueba	Validar la correcta Actualización de una norma al sistema.
Estrategia	<ul style="list-style-type: none"> • Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos. • Se ingresarán datos válidos y no válidos analizando su comportamiento.
Herramientas necesarias	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> • Se Actualiza una norma incluida en el sistema. • Si la información ingresada en la modificación del usuario no es válida, la aplicación debe generar un mensaje de error notificando el problema.
Consideraciones especiales	<ul style="list-style-type: none"> • Solo puede ser realizado desde el perfil de usuario Administrador.

Fuente: Autor

Cuadro 66. Prueba funcional – Eliminar Norma

Objetivo de la prueba	Validar la correcta Eliminación de una norma del sistema.
Estrategia	<ul style="list-style-type: none"> • Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos.
Herramientas necesarias	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> • Se Elimina la norma del sistema.
Consideraciones especiales	<ul style="list-style-type: none"> • Solo puede ser realizado desde el perfil de usuario Administrador.

Fuente: Autor

Cuadro 67. Prueba funcional – Consultar Norma

Objetivo de la prueba	Validar la correcta Consulta de una norma del sistema.
Estrategia	<ul style="list-style-type: none"> Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos.
Herramientas necesarias	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> Se Consulta la norma del sistema.
Consideraciones especiales	<ul style="list-style-type: none"> Puede ser realizado desde los perfiles Auditor, Consultor y Administrador.

Fuente: Autor

Cuadro 68. Prueba funcional – Implementar Control

Objetivo de la prueba	Validar la correcta Implementación de un control sobre una norma del sistema.
Estrategia	<ul style="list-style-type: none"> Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos. Se ingresarán datos válidos y no válidos analizando su comportamiento.
Herramientas necesarias	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> Se Implementa un control sobre un norma del sistema. Si la información ingresada en la modificación del usuario no es válida, la aplicación debe generar un mensaje de error notificando el problema.
Consideraciones especiales	<ul style="list-style-type: none"> Solo puede ser realizado desde el perfil de usuario Consultor. El control ya debió ser agregado previamente por el usuario Administrador; control que está asociado con una norma.

Fuente: Autor

Cuadro 69. Prueba funcional – Actualizar Control

Objetivo de la prueba	Validar la correcta Actualización de un control sobre una norma del sistema.
Estrategia	<ul style="list-style-type: none"> • Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos. • Se ingresarán datos válidos y no válidos analizando su comportamiento.
Herramientas necesarias	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> • Se Actualiza un control ya implementado sobre un norma del sistema. • Si la información ingresada en la modificación del usuario no es válida, la aplicación debe generar un mensaje de error notificando el problema.
Consideraciones especiales	<ul style="list-style-type: none"> • Solo puede ser realizado desde el perfil de usuario Consultor.

Fuente: Autor

Cuadro 70. Prueba funcional – Ver Control

Objetivo de la prueba	Validar la correcta Consulta de un control sobre una norma del sistema.
Estrategia	<ul style="list-style-type: none"> • Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos.
Herramientas necesarias	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> • Se Consulta un control ya implementado sobre un norma del sistema.
Consideraciones especiales	<ul style="list-style-type: none"> • Puede ser realizado desde el perfil de usuario Consultor y Auditor.

Fuente: Autor

Cuadro 71. Prueba funcional – Agregar Comentario

Objetivo de la prueba	Validar la correcta Adición de un comentario sobre un control asociado a una norma del sistema.
Estrategia	<ul style="list-style-type: none"> Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos. Se ingresarán datos válidos y no válidos analizando su comportamiento.
Herramientas necesarias	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> Se Agrega un comentario sobre un control ya implementado sobre un norma del sistema.
Consideraciones especiales	<ul style="list-style-type: none"> Solo puede ser realizado desde el perfil de usuario Consultor.

Fuente: Autor

Cuadro 72. Prueba funcional – Ver Comentario

Objetivo de la prueba	Validar la correcta Consulta de un comentario sobre un control asociado a una norma del sistema.
Estrategia	<ul style="list-style-type: none"> Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos.
Herramientas necesarias	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> Se Consulta un comentario sobre un control ya implementado sobre un norma del sistema.
Consideraciones especiales	<ul style="list-style-type: none"> Puede ser realizado desde el perfil de usuario Consultor y Auditor.

Fuente: Autor

Cuadro 73. Prueba funcional – Agregar Observación

Objetivo de la prueba	Validar la correcta Adición de una observación sobre un control asociado a una norma del sistema.
Estrategia	<ul style="list-style-type: none"> Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos. Se ingresarán datos válidos y no válidos analizando su comportamiento.
Herramientas necesarias	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> Se Agrega una observación sobre un control ya implementado sobre un norma del sistema.
Consideraciones especiales	<ul style="list-style-type: none"> Solo puede ser realizado desde el perfil de usuario Auditor.

Fuente: Autor

Cuadro 74. Prueba funcional – Ver Observación

Objetivo de la prueba	Validar la correcta Consulta de una observación sobre un control asociado a una norma del sistema.
Estrategia	<ul style="list-style-type: none"> Validación del funcionamiento de los métodos teniendo en cuenta los pasos definidos en cada uno de los requisitos mínimos establecidos.
Herramientas necesarias	Aplicación, Base de datos
Criterio de éxito	<ul style="list-style-type: none"> Se Consulta una observación sobre un control ya implementado sobre un norma del sistema.
Consideraciones especiales	<ul style="list-style-type: none"> Puede ser realizado desde el perfil de usuario Consultor y Auditor.

5.2.2.5.2 Definición de Requisitos del Entorno de Pruebas. A continuación se recopilan los requisitos relativos al entorno de pruebas, completando el plan de pruebas a través del cuadro 75.

Cuadro 75. Requisitos del entorno de pruebas

Req.	Tipo	Descripción
1	Hardware	Servidor Dedicado con procesador de 2 GHz, Memoria 4GB y Disco Duro 40 GB
2	Software	Microsoft Windows Server 2008
3	Software	Microsoft Internet Information Services IIS 7
4	Software	Motor de bases de datos Microsoft Windows
5	Software	Nhibernate versión 3.3.3
6	Software	Navegador Web
7	Entorno	Infraestructura Cliente Servidor
8	Entorno	Base de datos de la aplicación "Gestion"
9	Entorno	Permisos a nivel de red para acceder al servidor.

Fuente: Autor

5.2.2.5.3 Definición de las Pruebas de Aceptación del Sistema

A continuación a través de los cuadros 76, 77, 78, y 79 se definen las pruebas de aceptación del sistema.

Cuadro 76. Definición de pruebas unitarias

Objetivo de la Prueba:	<ul style="list-style-type: none"> • Verificar la funcionalidad y estructura de cada componente del sistema de forma individual una vez codificado.
Estrategia:	<ul style="list-style-type: none"> • Se realiza sobre las funciones principales de la aplicación. • Casos de prueba que validen la correcta funcionalidad de las funciones generadas.
Herramienta requeridas:	Aplicación, Base de datos
Observaciones	<ul style="list-style-type: none"> • Las pruebas se realizaran por cada módulo. • Se validarán entradas y salidas de cada módulo.

Fuente: Autor

- Pruebas del sistema

Cuadro 77. Definición pruebas del sistema.

Objetivo de la Prueba:	<ul style="list-style-type: none"> • Buscar diferencias entre requerimientos y la solución desarrollada.
Estrategia:	<ul style="list-style-type: none"> • Probar la navegación, ingreso de datos, procesamiento y recuperación. • Metodología de caja negra, orientado a pruebas desde la GUI.
Herramienta requeridas:	Aplicación, Base de datos
Observaciones	<ul style="list-style-type: none"> • Pruebas de usabilidad • Pruebas de desempeño

Fuente: Autor

- Pruebas de integración

Cuadro 78. Definición pruebas de integración

Objetivo de la Prueba:	<ul style="list-style-type: none"> • Identificación de errores en la interacción de programas.
Estrategia:	<ul style="list-style-type: none"> • Verificar la interacción entre programas. • Verificar que las interfaces externas dirigidas a usuarios, y la aplicación funcionan correctamente.
Herramienta requeridas:	Aplicación, Base de datos, Aplicaciones externas.
Observaciones	<ul style="list-style-type: none"> • Comparar el resultado esperado con el obtenido. • Enfoque desde el punto de vista de usuario • Enfoque desde el trabajo entre aplicaciones y base de datos.

Fuente: Autor

- Pruebas de seguridad

Cuadro 79. Definición pruebas de seguridad

Objetivo de la Prueba:	<ul style="list-style-type: none"> • Medir el nivel de seguridad de la aplicación.
Estrategia:	<ul style="list-style-type: none"> • Validación de entradas • Acceso solo a lo autorizado • Protocolos de comunicación segura • Defecto seguro • Separación de privilegios
Herramienta requeridas:	Aplicación, Base de datos.
Observaciones	<ul style="list-style-type: none"> • Se realiza validación para cada tipo de usuario • Validar las interfaces de acceso ofrecido a los usuarios.

Fuente: Autor

5.2.2.6. Aprobación del análisis del sistema de información

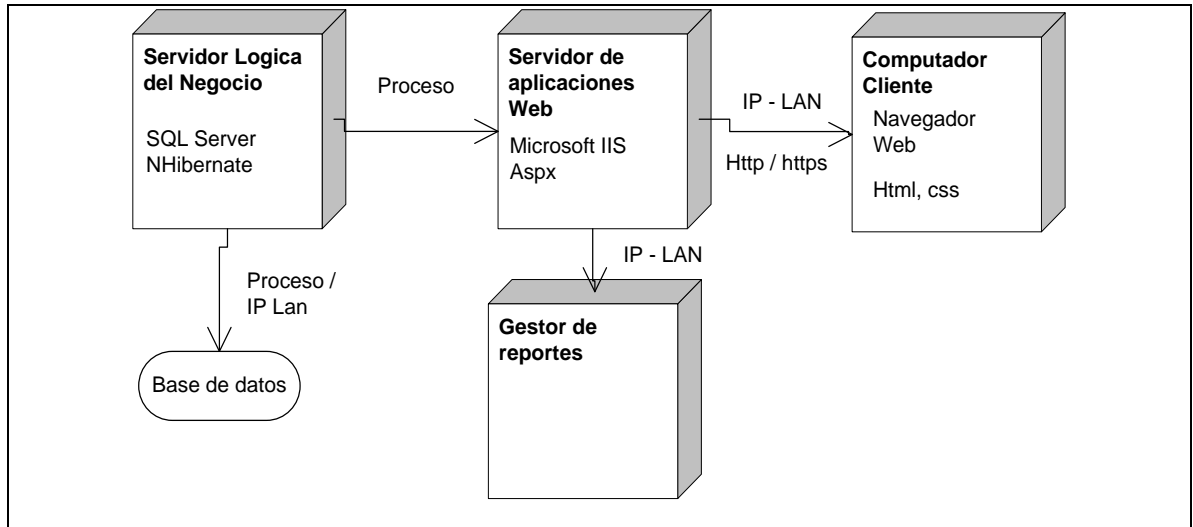
5.2.2.6.1 Presentación y Aprobación del Análisis del Sistema de Información. El análisis del sistema de información es aprobado por la empresa ALFCOM S.A., dado que se cubren las estrategias esperadas para continuar con el desarrollo de la aplicación.

5.2.3. Diseño del sistema de información (DSI). El propósito de esta tarea es definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información.

5.2.3.1. Definición de la arquitectura del sistema. A continuación se define la arquitectura general del sistema de información, especificando las distintas particiones físicas del mismo, la descomposición lógica en subsistemas de diseño y la ubicación de cada subsistema en cada partición.

5.2.3.1.1 Definición de Niveles de Arquitectura. A través de esta actividad se describen los niveles de la arquitectura software, mediante la definición de las principales particiones físicas del sistema de información, representadas como nodos y comunicaciones entre nodos, los cuales son descritos en la figura 47.

Figura 47. Niveles de arquitectura.



Fuente: Autor

5.2.3.1.2 Identificación de Requisitos de Diseño y Construcción. A continuación en el cuadro 80, se realiza la especificación de los requisitos que están directamente relacionados con la adopción o diseño de una arquitectura o infraestructura concreta, y que pueden condicionar el diseño o la construcción del sistema de información.

- Actualización de catálogo de requisitos

Cuadro 80. Requisitos de diseño y construcción (Actualización)

Requisito	Tipo	Descripción
38	Diseño	El entorno de programación debe ser Visual Studio
39	Diseño	EL lenguaje de programación utilizado es C# y ASPX
40	Diseño	La aplicación debe ser ejecutada desde un servidor web

Cuadro 80. (Continuación)

Requisito	Tipo	Descripción
41	Diseño	La interfaz gráfica debe ser diseñada utilizando hojas de estilo en cascada CSS
42	Interfaces Externas	La comunicación entre la capa lógica y la base de datos debe realizarse a través de la aplicación Hibernate
43	Diseño	La aplicación debe ser cargada sobre un servidor web Microsoft IIS
44	Diseño	La base de datos debe ser cargada en un servidor de Base de Datos Microsoft SQL Server
45	Diseño	La aplicación debe ser ejecutada sobre un servidor Windows Server 2008, dispuesto por la empresa Alfcom.
46	Interfaces Externas	El módulo de base de datos debe poder ser ejecutado desde un servidor diferente al de aplicación, garantizando independencia y escalabilidad.
47	Seguridad	La aplicación debe permitir el respaldo de la información alojada en su base de datos.

Fuente: Autor

5.2.3.1.3 Especificación de Excepciones. El objetivo de esta tarea es la definición de los comportamientos no habituales en el sistema, que reflejan situaciones anómalas o secundarias en el funcionamiento y ejecución del sistema de información. Para ello, se establece previamente el nivel de especificación de las mismas, así como los criterios de catalogación y clasificación.

Se define lista de excepciones en el documento “Requisitos faltantes - Aplicación ISS” Anexo 1.

5.2.3.1.4 Especificación de Estándares y Normas de Diseño y Construcción

A través del cuadro 81 se describen las normas escogidas en el diseño y construcción del presente proyecto.

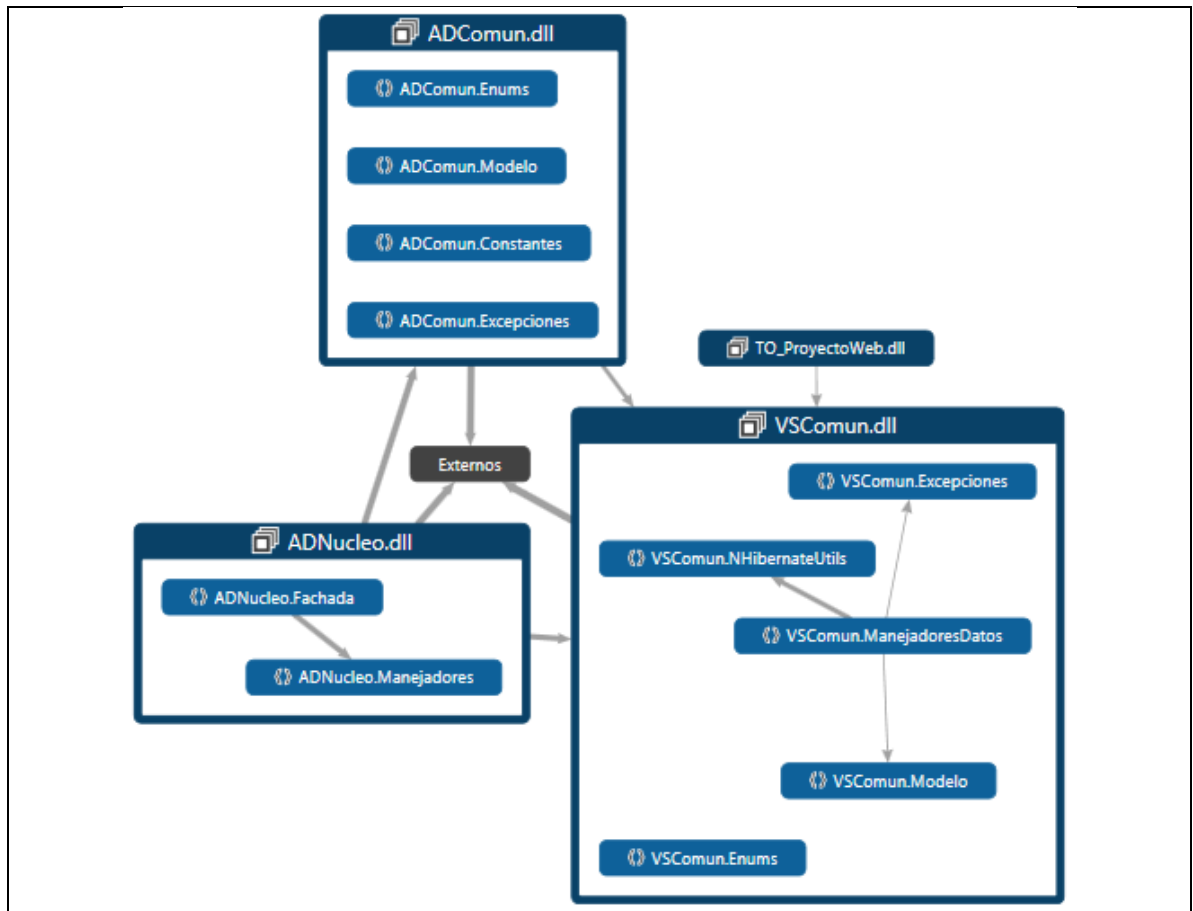
Cuadro 81. Catálogo de normas de diseño y construcción

Norma	Descripción
Norma ISO 27001	Modelo para establecer, implementar, operar, monitorear, revisar, mantener y mejorar un Sistema de Gestión de Seguridad de la Información (SGSI).
Norma ISO 27002	Objetivos de control y controles diseñados para ser implementados para satisfacer los requerimientos identificados para una evaluación de riesgo.
Circular 052 de 2007	Obliga a las entidades vigiladas tener en cuenta en la contratación de outsourcing lineamientos asociados con la seguridad de la información.
Métrica V3.	Metodología para planificar, desarrollar y mantener sistemas de información.
Norma Técnica Colombiana NTC 1486	Documentación. Presentación de tesis, trabajos de grado y otros trabajos de investigación.

Fuente: Autor

5.2.3.1.5 Identificación de Subsistemas de Diseño. A continuación se identifican los subsistemas de diseño de aplicación, asociado a los subsistemas de análisis especificados en el proceso de Análisis del Sistema de Información (ASI).

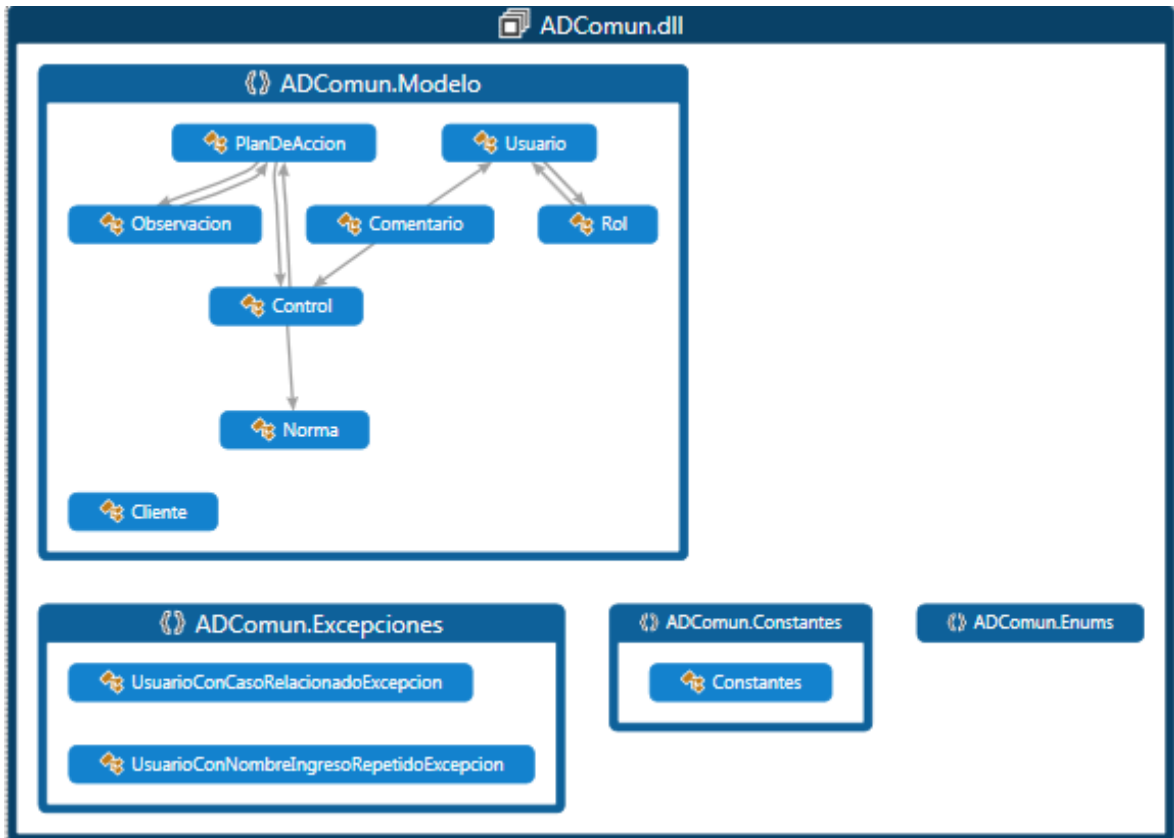
Cuadro 82. Vista general de subsistemas de diseño de aplicación



Fuente: Autor

En cuadro 82 se exponen la comunicación existente entre los subsistemas específicos (ADComun.dll) y los subsistemas de soporte (ADNucleo.dll y VSComun).

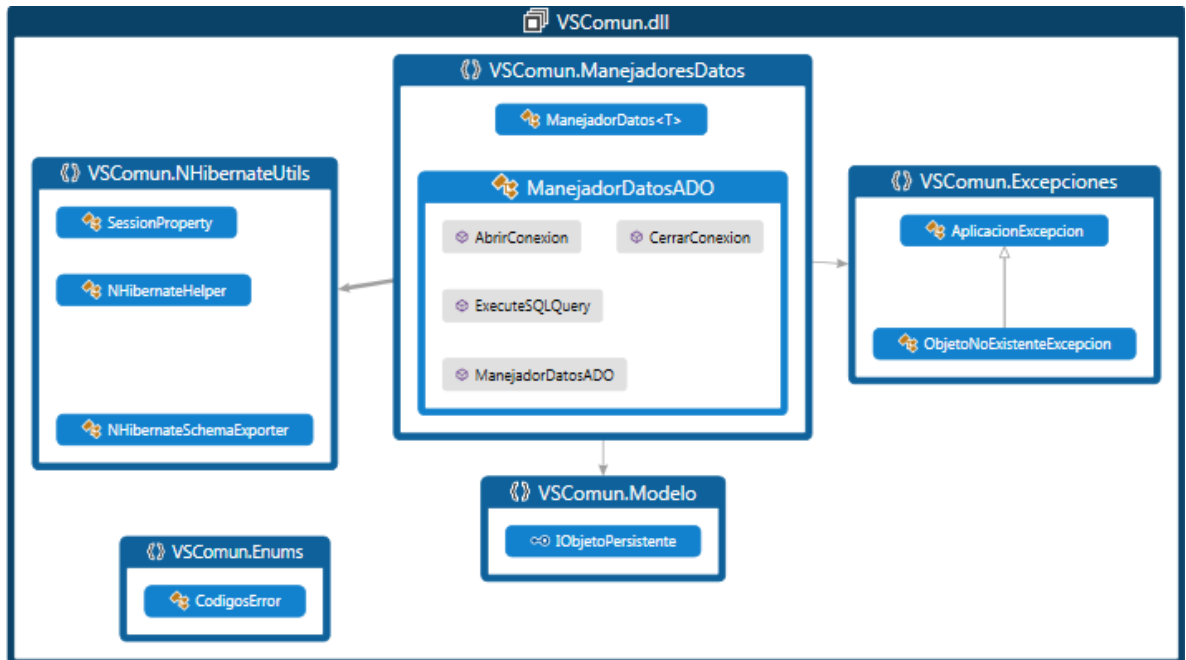
Figura 48. Subsistemas de diseño de aplicación – ADComun



Fuente: Autor

En la figura 48 se observa el subsistema específico ADComun.dll en donde se contemplan las funcionalidades propias del sistema de información.

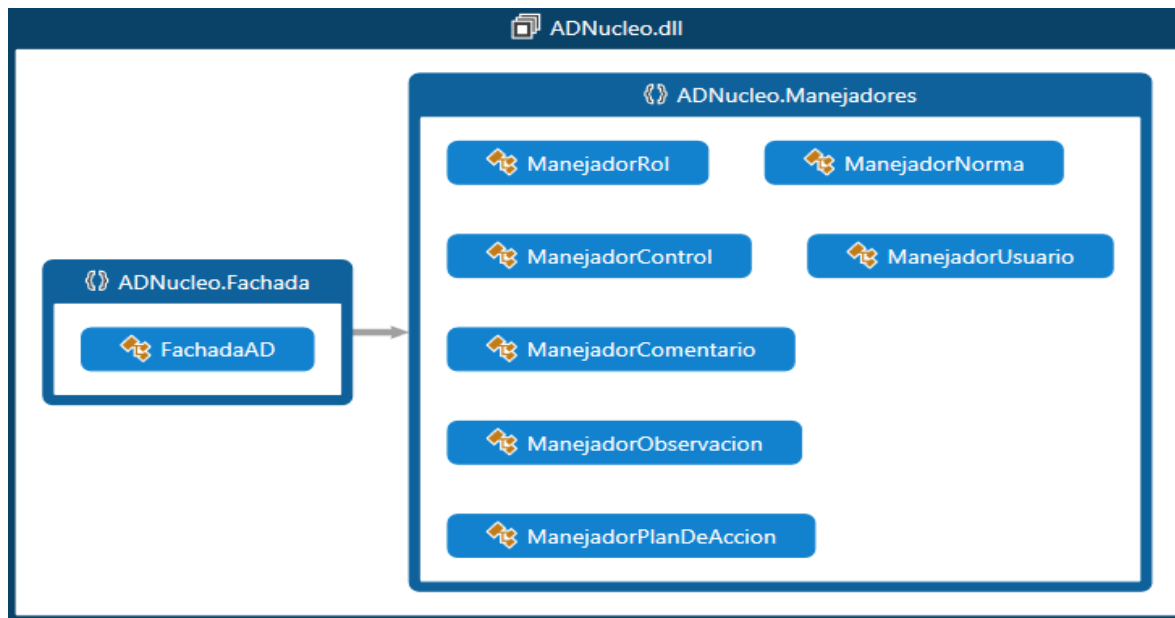
Figura 49. Subsistemas de diseño de aplicación – VSComun



Fuente: Autor

En la figura 49 se observa el subsistema específico VSComun.dll en donde se contemplan los servicios comunes proporcionando un acceso transparente a los distintos recursos del sistema de información.

Figura 50. Subsistemas de diseño de aplicación – ADNucleo



Fuente: Autor

En la figura 50 se observa el subsistema específico ADNucleo.dll en donde se contemplan los servicios comunes proporcionando un acceso transparente a los distintos recursos del sistema de información.

5.2.3.1.6 Especificación del Entorno Tecnológico. A continuación se describen las especificaciones del entorno tecnológico desde el punto de vista del Hardware, Software y Comunicaciones a través del cuadro 83.

Cuadro 83. Especificación del Entorno Tecnológico – Hardware y Software

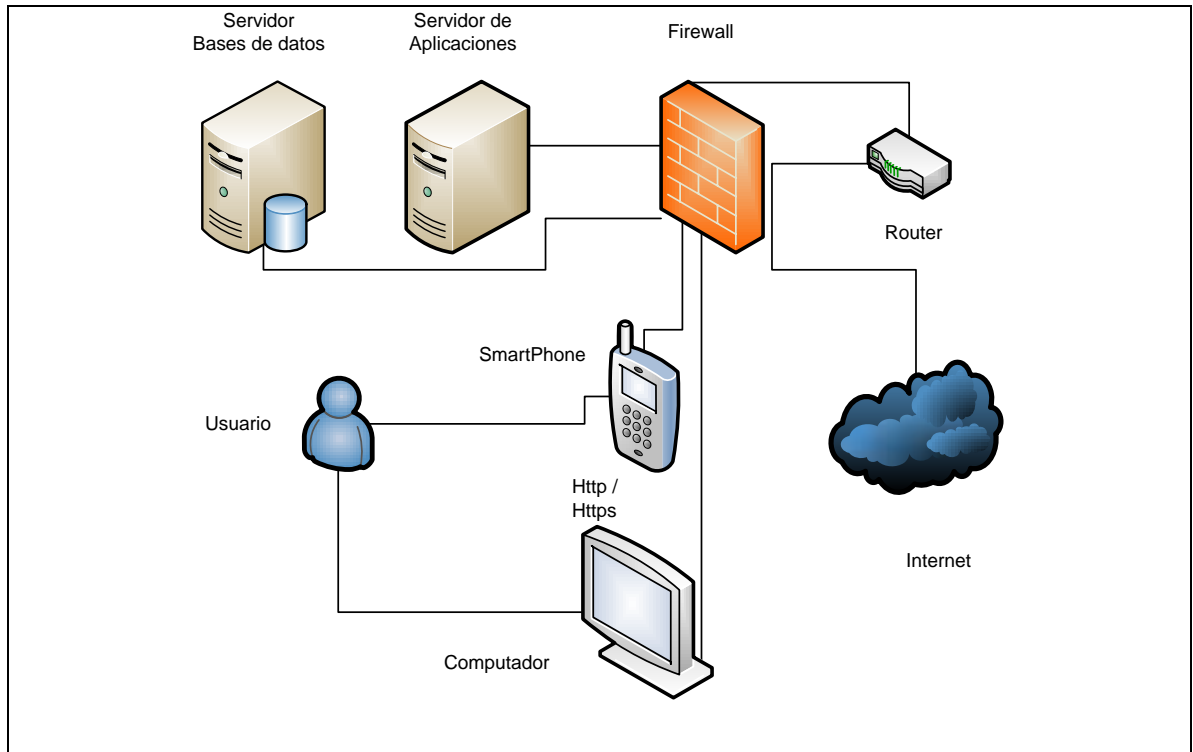
Especificación del Entorno Tecnológico		
Tipo	Componente	Requisito
Hardware	Procesador	3 GHz o Más
Hardware	Memoria	4GB
Hardware	Espacio en disco disponible	80 GB
Hardware	Unidad	DVD-ROM
Hardware	Pantalla	Super VGA (800X600)
Hardware	Periféricos	Teclado
Hardware	Periféricos	Mouse
Software	Sistema Gestor de Bases de datos	Microsoft SQL Server 2012
Software	Middleware	.NET 4.0
Software	Windows PowerShell	Windows PowerShell 2.0
Software	Sistema Operativo	Windows Server 2008 R2 SP1 Standard de 64 bits
Software	Internet Information Services	Internet Information Services (IIS) 7.5
Software	Nhibernate	Nhibernate 3.3.3
Software	Pentaho Reporting	Pentaho Reporting 5.0

Fuente: Autor

- Especificación del entorno tecnológico – Comunicaciones

A continuación en la figura 51 se muestra el entorno tecnológico y comunicaciones.

Figura 51. Especificación del Entorno Tecnológico – Comunicaciones



Fuente: Autor

5.2.3.1.7 Especificación de Requisitos de Operación y Seguridad

- Procedimientos de Seguridad y Control de Acceso: Acceso al sistema y sus recursos
 1. El sistema solo permite acceso a los usuarios autorizados, únicamente desde la interfaz de usuario.
 2. El acceso a las funciones del sistema se encuentran restringidas de acuerdo al rol del usuario.
 3. Solo el administrador de la aplicación puede crear nuevos usuarios.

4. La modificación de la base de datos debe ser restringido únicamente al personal de soporte de la aplicación.

- Mantenimiento de integridad y confidencialidad de los datos

1. Los campos de observaciones y comentarios definidos para los usuarios del tipo Auditor y Consultor respectivamente, no pueden ser editados desde la interfaz de usuario de la aplicación.
2. Solo el usuario del tipo administrador tiene la capacidad de eliminar las normas incluidas en el sistema.

- Copias de seguridad y recuperación de datos y periodicidad

1. Las copias de seguridad de la información de la aplicación debe realizarse haciendo una copia de seguridad de la base de datos llamada "Gestion" en ella se almacena toda la información ingresada por los usuarios.
2. La periodicidad de copia de datos queda a disposición de los administradores de la aplicación, se sugiere realizar una copia del sistema de forma mensual.

- Procedimientos de Operación y Administración del Sistema

Tratamiento en línea

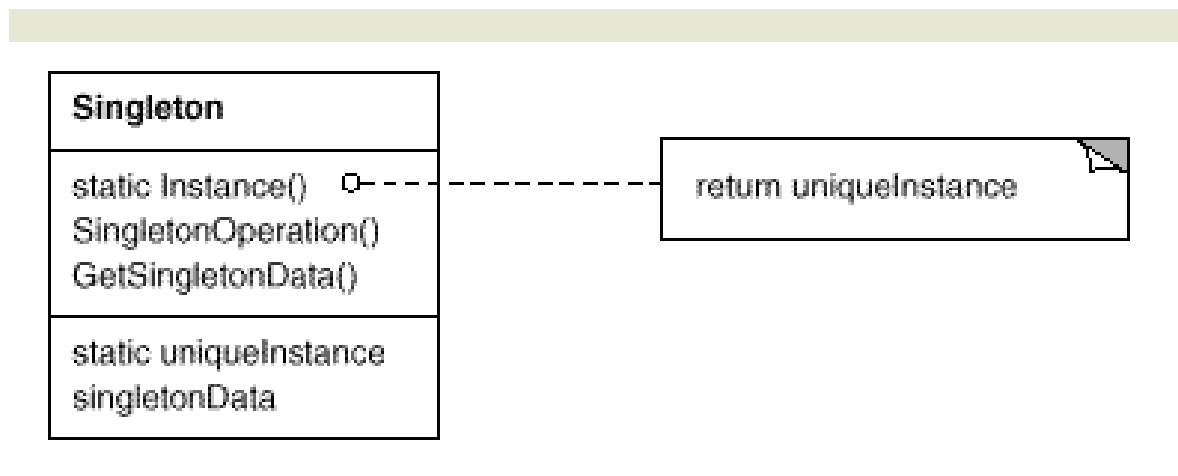
1. Número máximo de usuarios: 5
2. Periodos críticos de uso de la aplicación: De acuerdo a las necesidades de la empresa
3. La copia de seguridad de la base de datos de la aplicación debe ser realizada cuando la aplicación no se encuentre en uso.

5.2.3.2. Diseño de la arquitectura de soporte

5.2.3.2.1 Identificación de Mecanismos Genéricos de Diseño

- Patrón Singleton. El patrón Singleton garantiza que una clase solo tenga una instancia y proporciona un punto de acceso global a esta instancia el cual es especificado a través de la figura 52..

Figura 52. Patrón Singleton



Fuente: Autor

Se implementa el patrón Singleton en la clase Fachada como se observa en la figura 53.

Figura 53. Implementación Singleton

```
#region Propiedades
/// <summary>
/// Implementación del patrón Singleton
/// </summary>
public static FachadaAD Instancia
{
    get
    {
        if (m_Instancia == null)
        {
            m_Instancia = new FachadaAD();
        }
        return m_Instancia;
    }
}
#endregion Propiedades
```

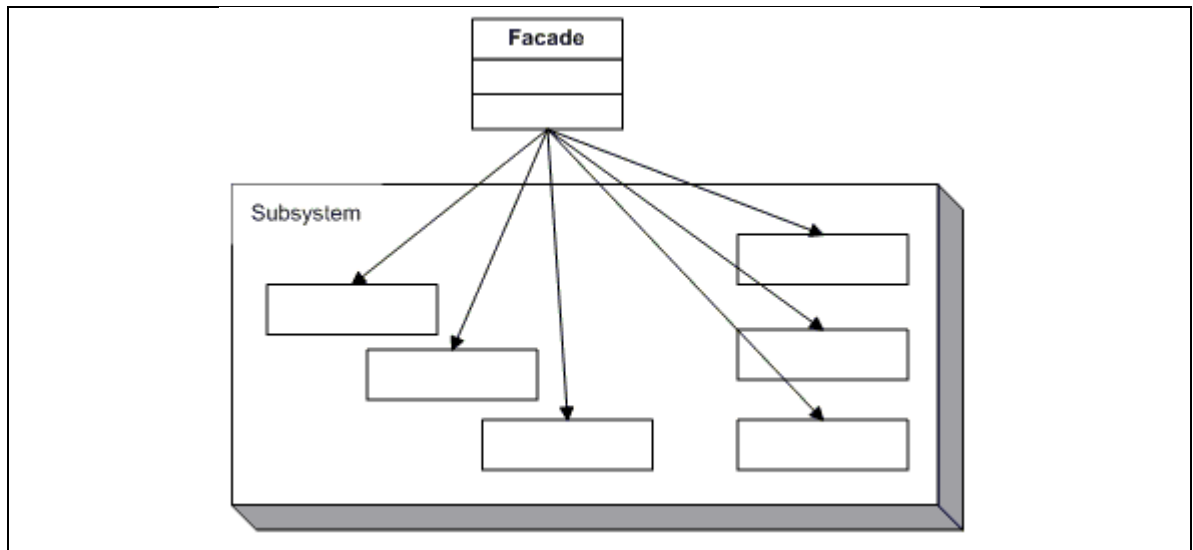
Fuente: Autor

Para poder acceder a los métodos de la clase fachada, es necesario llamar primero su instancia, la cual se crea a petición y asegura que en el proyecto se maneje una única instancia.

Patrón Fachada (FACADE)

Permite simplificar la comunicación entre objetos reduciendo su complejidad. En la figura 54 se define esta configuración.

Figura 54. Patrón Fachada



Fuente: Autor

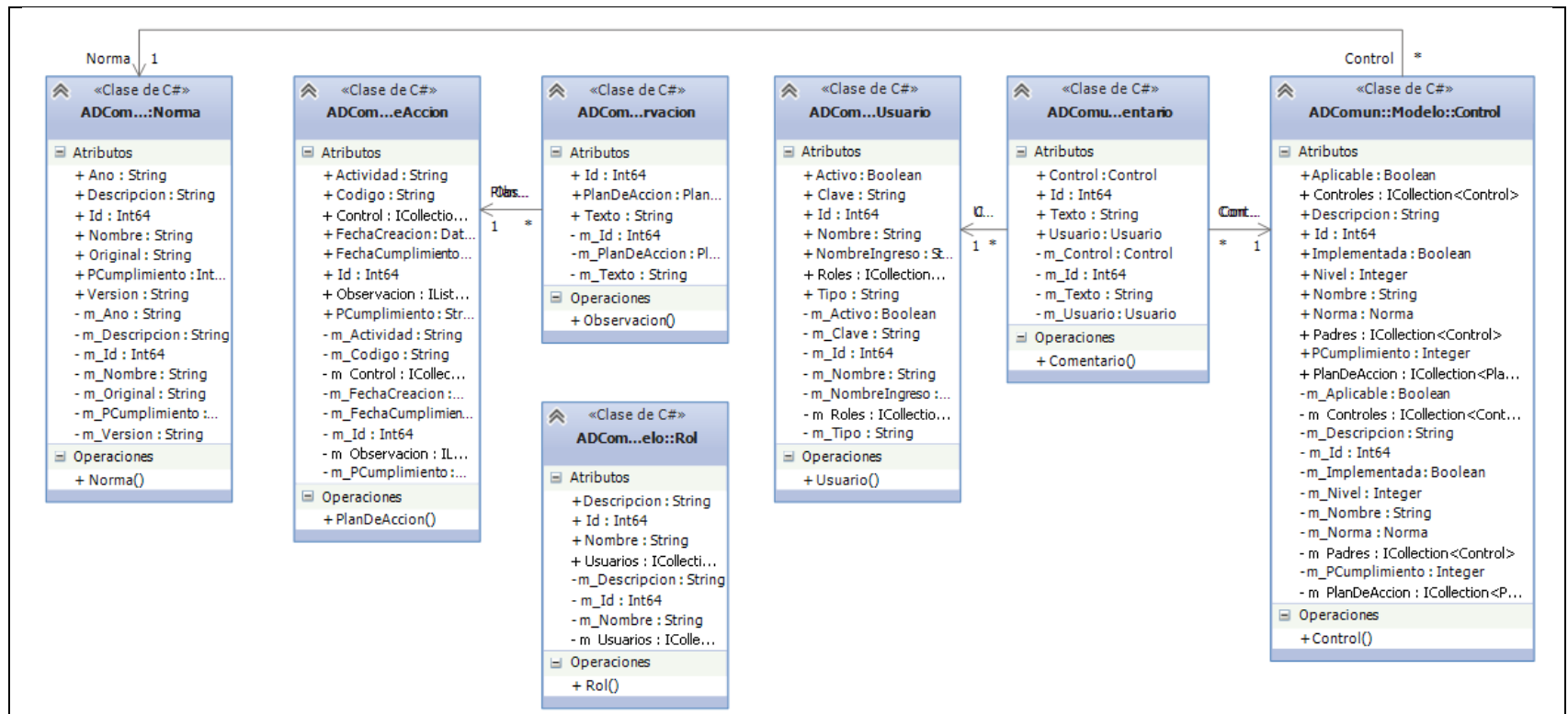
El patrón se implementó en los manejadores de las clases para así tener una única comunicación entre la capa de presentación y la lógica.

5.2.3.3. Diseño de casos de uso reales. A continuación se especifica el comportamiento del sistema de información para un caso de uso, mediante objetos o subsistemas de diseño que interactúan, y determinar las operaciones de las clases e interfaces de los distintos subsistemas de diseño.

5.2.3.3.1 Identificación de Clases Asociadas a Casos de Usos

Las clases asociadas a los casos de uso utilizados en el proyecto se describen a continuación en la figura 55.

Figura 55. Diagrama de clases asociadas a los casos de uso implementados

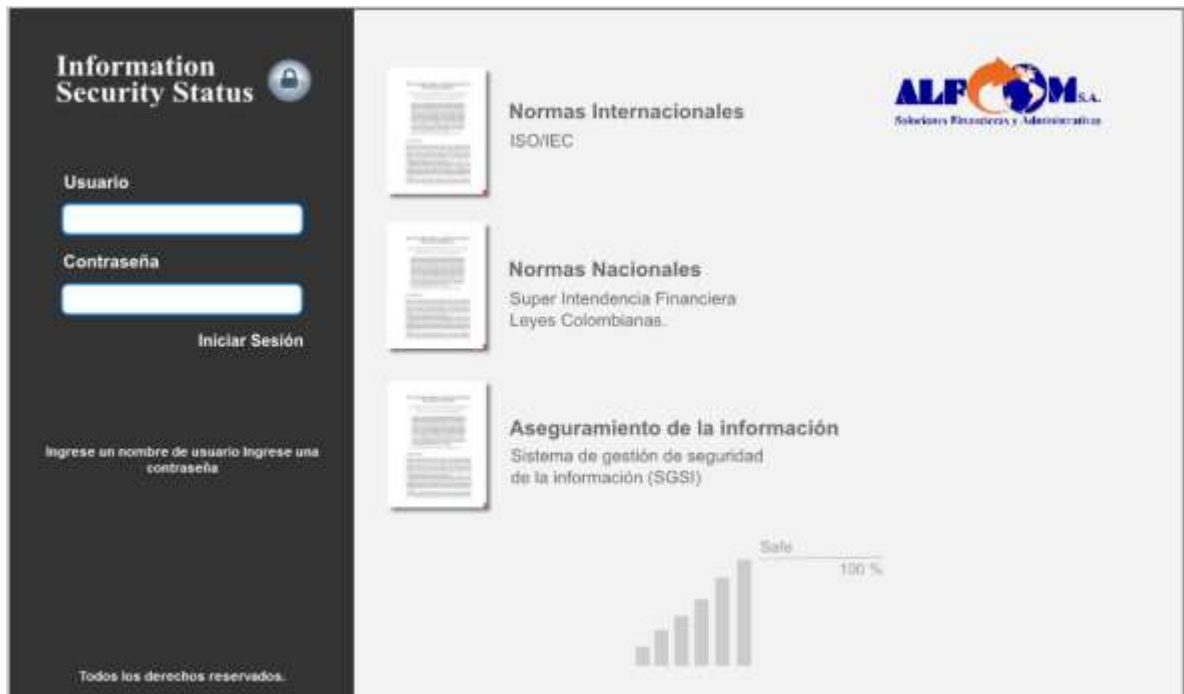


Fuente: Autor

5.2.3.3.2 Revisión de la Interfaz de Usuario

- Formatos Individuales de Interfaz de Pantalla Gráfica. Página de inicio de sesión: Esta página mostrada en la figura 56 permite el ingreso a la aplicación; es necesario contar con una cuenta de usuario creada en el sistema.

Figura 56. Interfaz de usuario – página de inicio de sesión



Fuente: Autor.

Página de principal: Al cargar esta página se observan las opciones otorgadas por el sistema de información para la gestión de normas de seguridad de la información descritas en la figura 57. La pantalla inicial muestra las normas implementadas en el sistema y su descripción. La pestaña norma de la aplicación muestra la misma información que esta página.

Figura 57. Interfaz de usuario – página principal



Fuente: Autor

Página Control: Esta página descrita en la figura 58, muestra los controles creados en la aplicación mostrando información relacionada como: Nombre control, Norma a la que pertenece, Nivel, y opciones de edición, vista, y eliminación.

Figura 58. Interfaz de usuario – página Control



Fuente: Autor

Página Plan de Acción: Esta página muestra los planes de acción creados en la aplicación, los cuales están asociados a un control de la norma. También es posible crear, editar, ver y modificar controles. La figura 59 muestra esta página.

Figura 59. Interfaz de usuario – página Plan de Acción

Código	Actividad	Fecha de creación	Fecha de cumplimiento	Opciones
2.1 Criterios de seguridad de la información	Listar controles asociados	07/08/2014 12:00:00 a.m.	12/08/2014 12:00:00 a.m.	[Iconos]
2.3 Canales de distribución de servicios financieros	Actividad 1	07/08/2014 12:00:00 a.m.	21/08/2014 12:00:00 a.m.	[Iconos]
1. Ámbito de aplicación	No aplica	09/08/2014 12:00:00 a.m.	07/08/2014 12:00:00 a.m.	[Iconos]

Fuente: Autor

Página Comentario: Esta página, descrita en la figura 60, muestra los comentarios creados a los controles de las normas. Esta actividad es realizada por usuarios con el rol “Consultor”.

Figura 60. Interfaz de usuario – página Comentarios

Texto	Control	Norma	Usuario	Opciones
Facil de implementar,	2.1 Criterios de seguridad de la información	Circular 052	admin	[Iconos]

Fuente: Autor

Página Observación: Esta página definida en la figura 61, muestra las observaciones creadas a los controles de las normas. Esta actividad es realizada por usuarios con el rol “Auditor”.

Figura 61. Interfaz de usuario – página Observación



Fuente: Autor

Página Usuario: Esta página descrita en la figura 62, muestra los usuarios creados en la aplicación. Permite igualmente editar, ver y crear usuarios. Esta tarea es realizada por usuarios con el rol “Administrador”.

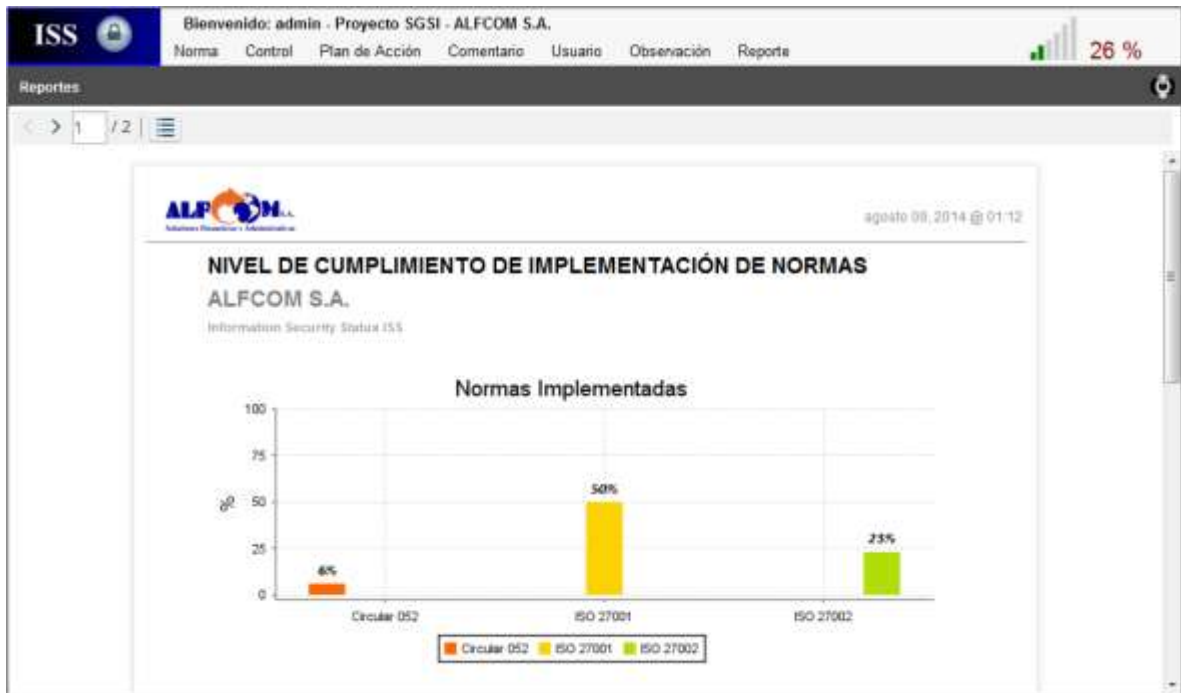
Figura 62. Interfaz de usuario – página Usuarios



Fuente: Autor

Página Reporte: Esta página descrita en la figura 63, muestra un reporte de la implementación de las normas incluidas en el sistema, junto con su lista de controles asociados. Los reportes generados pueden ser exportados bajo los formatos HTML, PDF, Excel, CSV, RTF y en formato TXT.




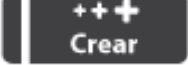



Figura 63. Interfaz de usuario – página Reporte



Fuente: Reporte

Catálogo de controles y elementos de diseño de la Interfaz de pantalla gráfica definidos en el cuadro 84.

Cuadro 84. Controles y elementos de diseño de la interfaz gráfica.

Elemento de diseño	Descripción
	Permite Consultar los elementos accesibles a través de la interfaz.
	Permite Editar los elementos accesibles a través de la interfaz.
	Permite Eliminar los elementos accesibles a través de la interfaz.
	Permite Crear nuevos elementos accesibles a través de la interfaz.
	Permite realizar búsquedas sobre los registros de la aplicación.
	Permite Guardar elementos creados o modificados a través de la interfaz.
	Permite Cancelar operaciones que el usuario no desea realizar.

Fuente: Autor

- Modelo de Navegación de Interfaz de Pantalla Gráfica. La navegación de la aplicación es realizada a través del menú ubicado en la parte superior de la aplicación descrito en la figura 64, en él se encuentran todas las opciones ofrecidas por la aplicación.

Figura 64. Modelo de navegación de interfaz de pantalla gráfica.



Fuente: Autor

5.2.3.4. Diseño de clases

5.2.3.4.1 Identificación de Clases Adicionales. El diseño original de clases se mantuvo, por consiguiente no se crearon clases adicionales.

5.2.3.4.2 Diseño de Asociaciones y Agregaciones. Las asociaciones de las clases del proyecto se evidencian en la *Figura 2. Diagrama de clases inicial*, ya que se tienen las clase con sus respectivas asociaciones entre si.

5.2.3.4.3 Identificación de Atributos de las Clases. Los atributos de la clase se exponen en la Figura 2. Diagrama de clases inicial, en la cual pueden ser consultadas por sus clases correspondientes.

5.2.3.4.4 Identificación de Operaciones de las Clases. Gracias a la implementación del patrón fachada, todas las operaciones son llamadas desde la misma, por consiguiente se adjunto el código de las llamadas de la fachada a los manejadores a través del Anexo 6.

5.2.3.4.5 Diseño de la Jerarquía. La jerarquía de las clases implementadas comienza desde la norma, la cual tiene controles asociados, luego se despliegan los planes de acción sobre cada control de último nivel.

5.2.3.4.6 Descripción de Métodos de las Operaciones. Los métodos de las operaciones se describen en el Anexo 3, Código fuente de la aplicación, con su respectiva documentación dentro de cada una de las clases.

5.2.3.4.7 Especificación de Necesidades de Migración y Carga Inicial de Datos. La carga inicial de datos es realizada por el usuario con el rol de administrador, teniendo como objetivo agregar la norma, y posteriormente la creación e implementación de controles.

5.2.3.5. Diseño físico de datos

5.2.3.5.1 Diseño del Modelo Físico de Datos. Se describen a continuación a través de las figuras 65, 66, 67, 68, 69, 70 y 71 las clases transformadas en tablas debido a que la información es almacenada en el SGBD Microsoft SQL Server.

Figura 65. Tabla base de datos – Clase Norma

Norma	
normalId	
nombre	
version	
ano	
original	
descripcion	
pCumplimiento	

«Clase de C#» ADComun::Modelo::Norma	
Atributos	
+ Ano	: String
+ Descripcion	: String
+ Id	: Int64
+ Nombre	: String
+ Original	: String
+ PCumplimiento	: Integer
+ Version	: String

Fuente: Autor

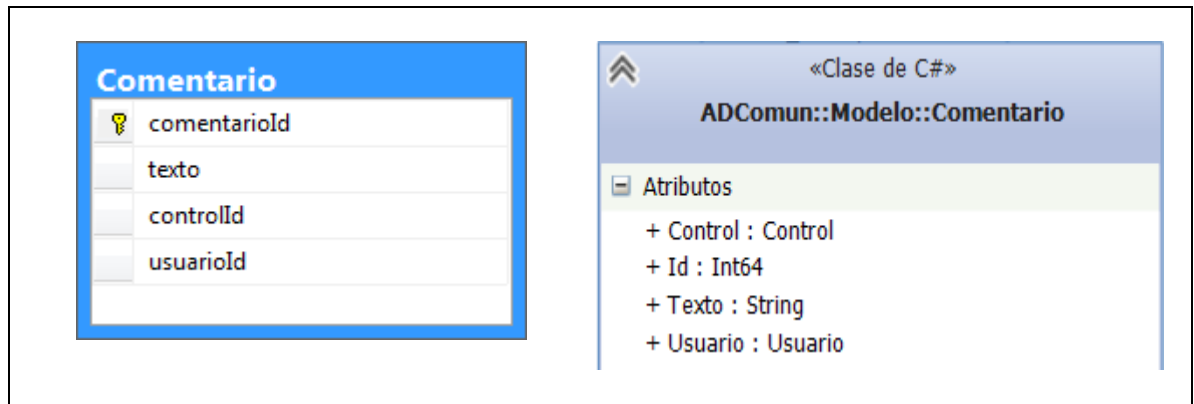
Figura 66. Tabla base de datos – Clase Control

Control	
controlId	
aplicable	
implementada	
nombre	
nivel	
descripcion	
normalId	
PCumplimiento	

«Clase de C#» ADComun::Modelo::Control	
Atributos	
+ Aplicable	: Boolean
+ Controles	: ICollection<Control>
+ Descripcion	: String
+ Id	: Int64
+ Implementada	: Boolean
+ Nivel	: Integer
+ Nombre	: String
+ Norma	: Norma
+ Padres	: ICollection<Control>
+ PCumplimiento	: Integer
+ PlanDeAccion	: ICollection<PlanDeAccion>

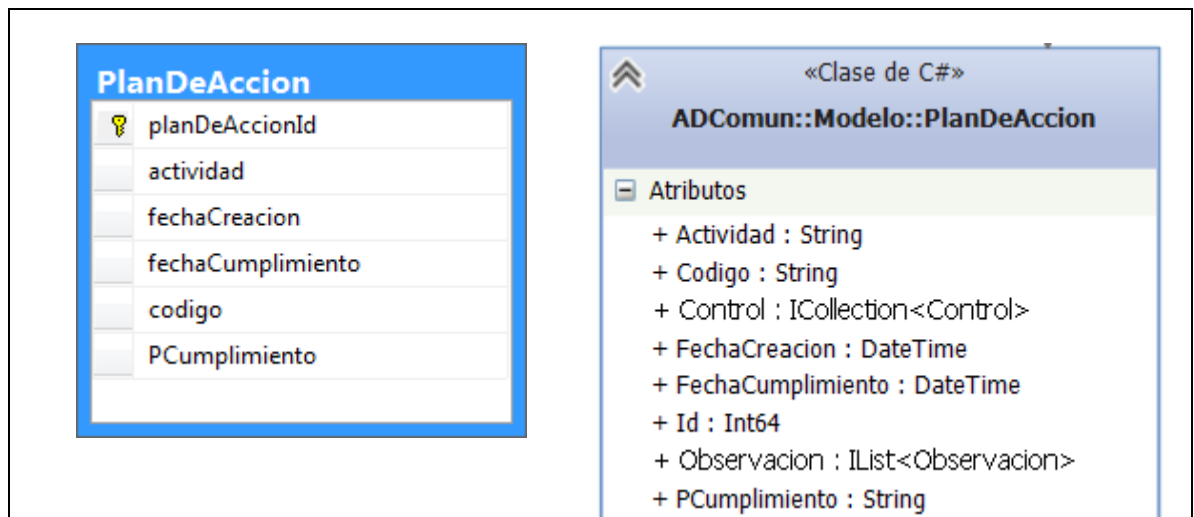
Fuente: Autor

Figura 67. Tabla base de datos – Clase Comentario



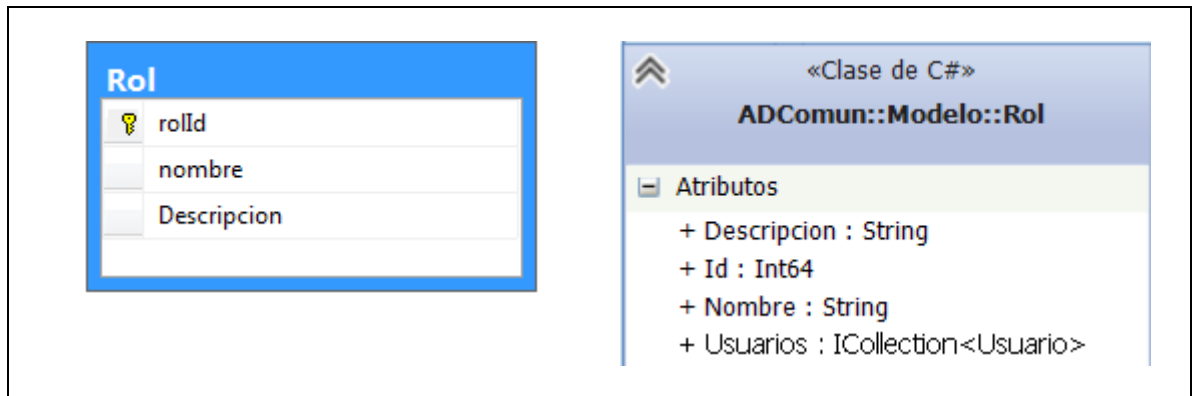
Fuente: Autor

Figura 68. Tabla base de datos – Clase PlanDeAccion



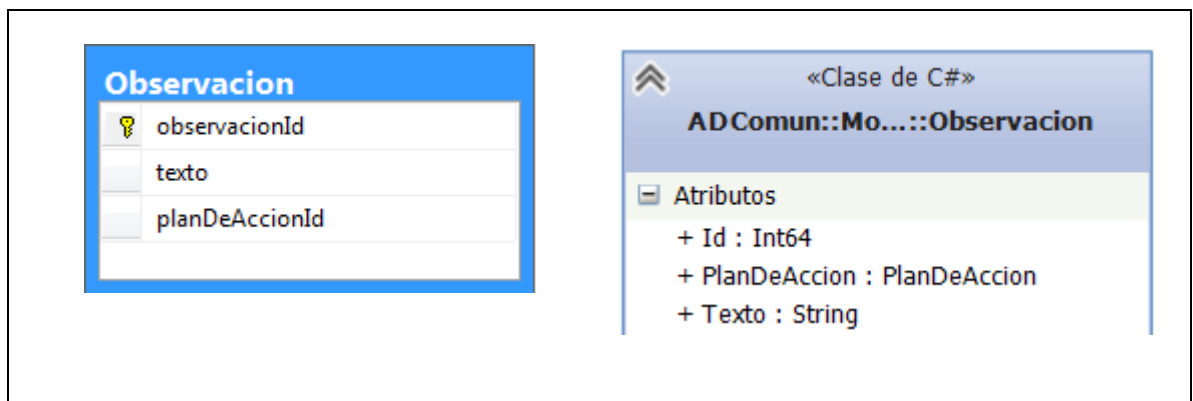
Fuente: Autor

Figura 69. Tabla base de datos – Clase Rol



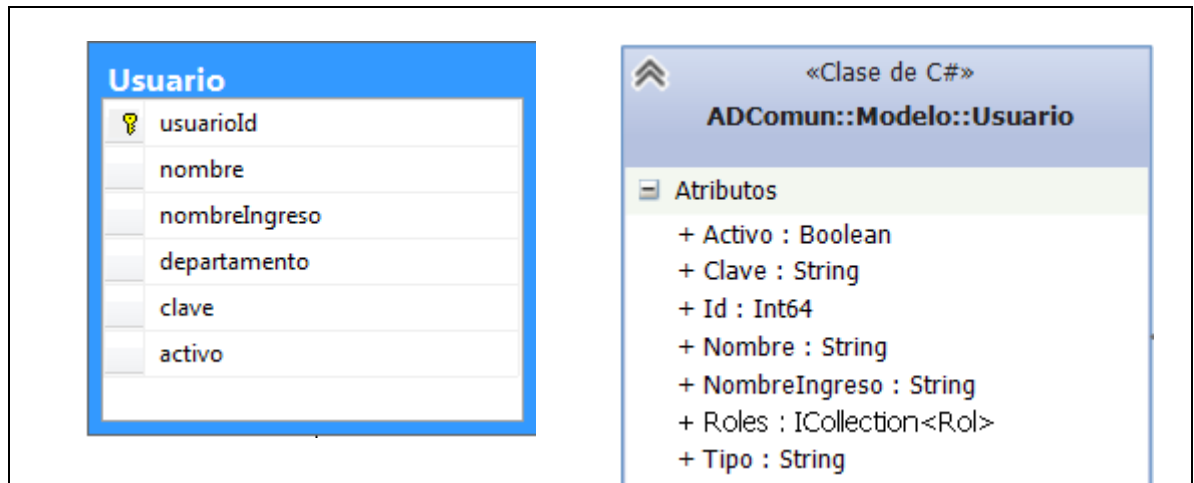
Fuente: Autor

Figura 70. Tabla base de datos – Observacion



Fuente: Autor

Figura 71. Tabla base de datos – Usuario



Fuente: Autor

5.2.3.5.2 Especificación de los Caminos de Acceso a los Datos. A continuación se describen los caminos de acceso a seguir para cada caso de uso especificado, a través de las tablas contenidas en la base de datos “Gestion” del proyecto.

Cuadro 85. Descripción del camino de acceso a los datos.

Acción	Pasos
Agregar Usuario	<ol style="list-style-type: none"> 1. Se accede a la tabla "Usuario" para comprobar que el nombre del usuario es nuevo 2. Si el nombre de usuario es nuevo, se accede a la tabla "Rol" para especificar el rol del usuario.
Modificar Usuario	<ol style="list-style-type: none"> 1. Se accede a la tabla "Usuario" utilizando el ID del usuario a modificar. De ser necesario la modificación del rol del Usuario, se accede a la tabla "Rol".
Eliminar Usuario	<ol style="list-style-type: none"> 1. Se elimina el usuario utilizando como valor único el ID del usuario al acceder a la tabla "Usuario".
Agregar Norma	<ol style="list-style-type: none"> 1. Se accede a la tabla "Norma" para comprobar que el nombre de la norma es nueva, si el nombre es nuevo se crea la nueva norma en la tabla.
Actualizar Norma	<ol style="list-style-type: none"> 1. Se accede a la tabla "Norma" utilizando el ID de la norma a modificar.
Eliminar Norma	<ol style="list-style-type: none"> 1. Se elimina la norma utilizando como valor único el ID de la norma al acceder a la tabla "Norma". Se verifica previamente si la norma tiene controles asociados.
Consultar Norma	<ol style="list-style-type: none"> 1. Se toma la información de la tabla "Norma" de acuerdo al ID de la norma a consultar.
Implementar Control	<ol style="list-style-type: none"> 1. Se accede a la tabla "Control" para crear nuevo control tomando como referencia el campo "Aplicable" de la misma tabla. 2. Se selecciona la norma a la que está asociado el control accediendo a la tabla "Norma"
Actualizar Control	<ol style="list-style-type: none"> 1. Se accede a la tabla "Control" utilizando el ID del control a modificar. 2. En caso de requerir la modificación de la norma asociada al control, se accede a la tabla "Norma".
Ver Control	<ol style="list-style-type: none"> 1. Se obtienen los datos de la tabla "Control" de acuerdo al ID del control a consultar.
Agregar Comentario	<ol style="list-style-type: none"> 1. Se accede a la tabla "Comentario" para agregar un comentario nuevo. 2. Se registra el usuario que ingresa el comentario accediendo a la base de datos "Usuario". 3. Se asocia el control al que va asociado el comentario creado accediendo a la tabla "Control".
Ver Comentario	<ol style="list-style-type: none"> 1. Se obtienen los datos de la tabla "Comentario" de acuerdo al ID del comentario a consultar.

Cuadro 85. (Continuación)

Acción	Pasos
Agregar Observación	<ol style="list-style-type: none"> 1. Se accede a la tabla "Observación" para agregar una nueva observación. 2. Se registra el usuario que ingresa la observación accediendo a la base de datos "Usuario". 3. Se asocia el control al que va asociada la observación creada accediendo a la tabla "Control".
Ver Observación	<ol style="list-style-type: none"> 1. Se obtienen los datos de la tabla "Observacion" de acuerdo al ID de la observación a consultar.
Agregar Plan de Acción	<ol style="list-style-type: none"> 1. Se accede a la tabla "PlanDeAccion" para crear un nuevo Plan de acción. 2. Se selecciona el control al que está asociado el plan de acción accediendo a la tabla "Control".
Modificar Plan de Acción	<ol style="list-style-type: none"> 1. Se accede a la tabla "PlanDeAcción" utilizando el ID del plan de acción a modificar.
Eliminar Plan de Acción	<ol style="list-style-type: none"> 1. Se elimina el registro por medio de su ID, el cual está contenido en la tabla "PlanDeAccion".

Fuente: Autor

En La siguiente matriz descrita en el cuadro 86, se especifica la relación Acciones – Tablas acerca de los caminos de acceso a los datos.

Cuadro 86. Matriz de especificación de acceso a los datos, relación Tablas – Casos

Acción	Tablas de la Base de datos "Gestion" de la Aplicación ISS						
	Usuario	Rol	Norma	Control	Comentario	Observacion	PlanDeAccion
Agregar Usuario							
Modificar Usuario							
Eliminar Usuario							
Agregar Norma							
Actualizar Norma							
Eliminar Norma							
Consultar Norma							
Implementar Control							
Actualizar Control							
Ver Control							
Agregar Comentario							
Ver Comentario							
Agregar Observación							
Ver Observación							
Agregar Plan de Acción							
Modificar Plan de Acción							
Eliminar Plan de Acción							

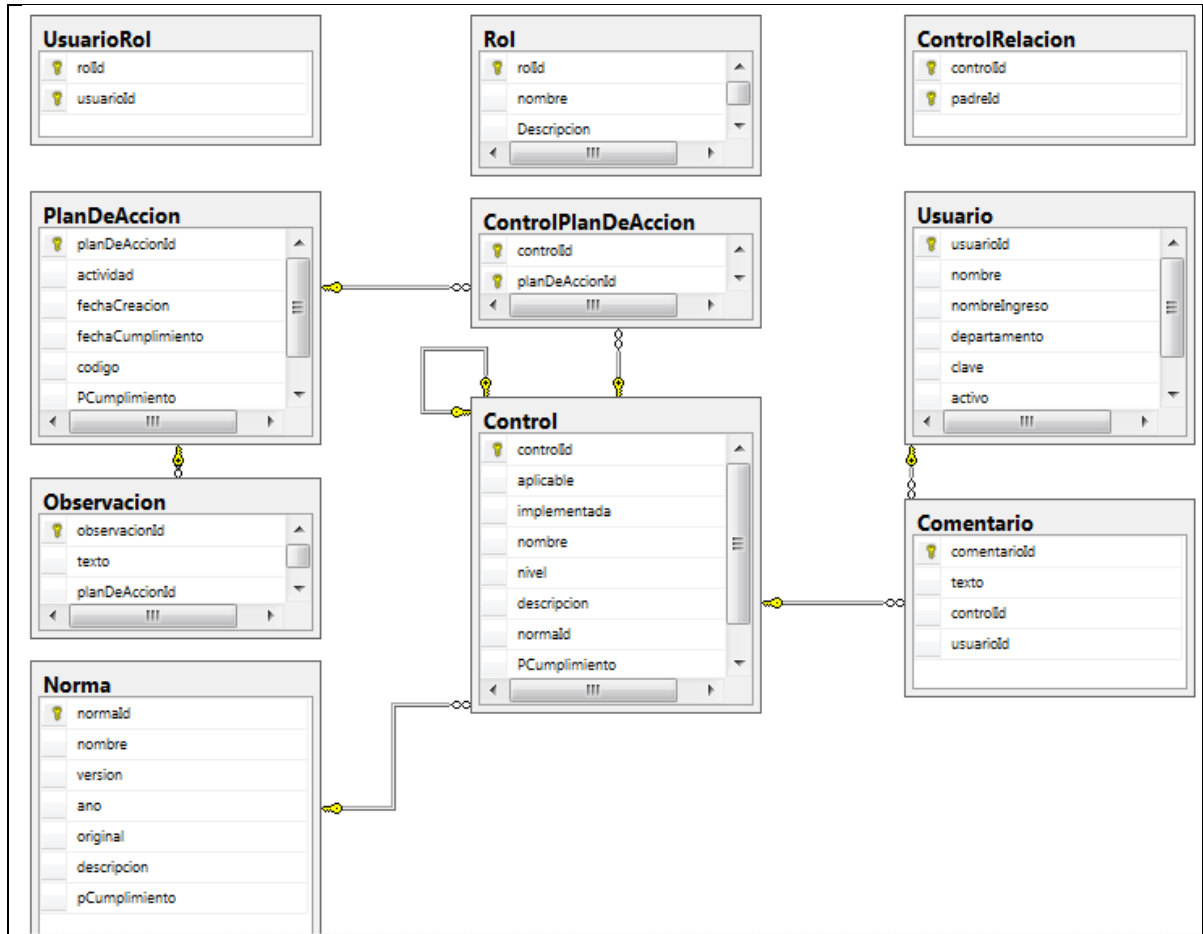
Fuente: Autor

5.2.3.5.3 Optimización del Modelo Físico de Datos. Se determina mantener los caminos de acceso a los datos definidos en el apartado anterior dado las siguientes observaciones:

- Las interacciones entre las tablas no pasan de 3 para los casos de uso.
- No existe comunicación con otras bases de datos.
- La información almacenada en su mayoría corresponde a texto plano.
- La aplicación está pensada para el acceso simultáneo de 5 usuarios.

5.2.3.5.4 Especificación de la Distribución de Datos. A continuación en la figura 72 se describe la distribución de datos a través del diagrama entidad relación de la base de datos utilizada por el sistema de información. La información es almacenada en el sistema gestor de bases de datos escogido para el proyecto Microsoft SQL Server 2012 Express.

Figura 72. Diagrama entidad relación, distribución de datos.



Fuente: Autor

5.2.3.6. Verificación y aceptación de la arquitectura del sistema

5.2.3.6.1 Verificación de las Especificaciones de Diseño. Se identificaron algunas inconsistencias respecto al cumplimiento de los requisitos definidos para la aplicación, los cuales se encuentran corregidos actualmente y son descritos a continuación:

- Presentación de los menús de navegación en la interfaz gráfica de usuario.
- Presentación de los elementos principales en la interfaz gráfica de usuario.

- Asociación de casos de uso relacionados al momento de usar la aplicación.
- Relaciones no definidas en el modelo físico de datos.
- Validación de campos de acuerdo a la información suministrada en la aplicación.
- Consistencia funcional de subsistemas de la aplicación.

5.2.3.6.2 Análisis de Consistencia de las Especificaciones de Diseño. El objetivo de esta tarea es asegurar que las especificaciones del diseño son coherentes entre sí, comprobando la falta de ambigüedades o duplicación de información. Esta consistencia se asegura entre especificaciones de diseño, y con respecto a los modelos del análisis. Las comprobaciones realizadas se hacen a través de matrices de elementos comunes en los siguientes modelos:

- Análisis de Clases vs Casos de uso. Sobre esta matriz se describe las diferentes clases de la aplicación que están relacionadas con los casos de uso estipulados.

Cuadro 87. Análisis de Clases vs Casos de uso.

Clases asociadas	Casos de uso					
	Gestión de Usuarios	Gestión de plan de acción	Gestión de normas	Gestión de controles	Gestión de comentarios	Gestión de observaciones
ADComun:Modelo:Comentario					x	
ADComun:Modelo:Norma		x	x	x	x	x
ADComun:Modelo:Usuario	x					
ADComun:Modelo:Rol	x					
ADComun:Modelo:Observación						x
ADComun:Modelo:Control		x	x	x	x	x
ADComun:Modelo:PlanDeAccion		x				

Fuente: Autor

5.2.3.6.3 Aceptación de la Arquitectura del Sistema. El objetivo de esta tarea es obtener la aceptación, por parte de las áreas de explotación y sistemas, de la arquitectura del sistema de información y de los requisitos de operación y seguridad, con el fin de poder valorar su impacto en la instalación.

De acuerdo a consulta con el personal encargado del desarrollo de la aplicación por parte de la empresa ALFCOM S.A., se determinará la aceptación de la arquitectura del sistema propuesto para el presente proyecto.

5.2.3.7. Generación de especificaciones de construcción

5.2.3.7.1 Especificación del Entorno de Construcción. El objetivo de esta tarea es la definición detallada y completa del entorno necesario para la construcción de los componentes del sistema de información.

Se propone que la especificación del entorno se realice según los siguientes conceptos:

- Entorno tecnológico: hardware, software y comunicaciones.
- Herramientas de construcción, generadores de código, compiladores, etc.
- Restricciones técnicas del entorno.
- Planificación de capacidades previstas, o la información que estime oportuno el departamento de sistemas para efectuar dicha planificación.
- Requisitos de operación y seguridad del entorno de construcción.

A continuación se describen las especificaciones del entorno de construcción.

Cuadro 88. Especificaciones del entorno de construcción.

Concepto	Descripción
Entorno tecnológico: hardware	<ul style="list-style-type: none">• Servidor Dedicado con procesador de 2 GHz, Memoria 4GB y Disco Duro 40 GB
Entorno tecnológico: software	<ul style="list-style-type: none">• Microsoft Windows Server 2008• Microsoft Internet Information Services IIS 7• Motor de bases de datos Microsoft Windows• Nhibernate versión 3.3.3• Navegador Web
Entorno tecnológico: Comunicaciones	<ul style="list-style-type: none">• Red Lan FastEthernet 100 Mbps
Herramientas de construcción, generadores de código, compiladores, etc.	<ul style="list-style-type: none">• Microsoft Visual Studio 2012 Express• Pentaho Business Analytics
Restricciones técnicas del entorno	<ul style="list-style-type: none">• Sin restricciones

Cuadro 88. (Continuación)

Concepto	Descripción
Planificación de capacidades previstas, o la información que estime oportuno el departamento de sistemas para efectuar dicha planificación.	<ul style="list-style-type: none"> • Infraestructura actual utilizada por la empresa ALFCOM S.A.
Requisitos de operación y seguridad del entorno de construcción	<ul style="list-style-type: none"> • No se observan requisitos.

Fuente: Autor

5.2.3.7.2 Definición de Componentes y Subsistemas de Construcción. Los componentes se definen mediante la agrupación de elementos del diseño de detalle de cada subsistema de diseño. En principio, cada módulo o clase y cada formato individual de interfaz se corresponden con un componente.

La tabla descrita a continuación se encuentra detallada de acuerdo a las dependencias expuestas en el numeral 3.2.5.1.5, Identificación de Subsistemas de Diseño.

Cuadro 89. Relación de clases con los componentes asociados.

Clase	Componentes asociados
ADComun:Modelo:Comentario	<ul style="list-style-type: none"> • ADNucleo.Fachada.m_ManejadorComentario • ADNucleo.Manejadores.ManejadorComentario
ADComun:Modelo:Norma	<ul style="list-style-type: none"> • ADNucleo.Fachada.m_ManejadorNorma • ADNucleo.Manejadores.ManejadorNorma
ADComun:Modelo:Usuario	<ul style="list-style-type: none"> • ADNucleo.Fachada.m_ManejadorUsuario • ADNucleo.Manejadores.ManejadorUsuario
ADComun:Modelo:Rol	<ul style="list-style-type: none"> • ADNucleo.Fachada.m_ManejadorRol • ADNucleo.Manejadores.ManejadorRol
ADComun:Modelo:Observacion	<ul style="list-style-type: none"> • ADNucleo.Fachada.m_ManejadorObservacion • ADNucleo.Manejadores.ManejadorObservacion

Cuadro 89. (Continuación)

Clase	Componentes asociados
ADComun:Modelo:Control	<ul style="list-style-type: none"> • ADNucleo.Fachada.m_ManejadorControl • ADNucleo.Manejadores.ManejadorControl
ADComun:Modelo:PlanDeAccion	<ul style="list-style-type: none"> • ADNucleo.Fachada.m_ManejadorPlanDeAccion • ADNucleo.Manejadores.ManejadorPlanDeAccion

Fuente: Autor

5.2.3.7.3 Elaboración de Especificaciones del Modelo Físico de Datos. En esta tarea se generan las especificaciones necesarias para la definición y creación de los elementos del modelo físico de datos, mediante el lenguaje de definición de datos del correspondiente gestor de base de datos o sistema de ficheros, teniendo en cuenta el entorno tecnológico, las normas y estándares de la organización y características intrínsecas del gestor o sistema de ficheros a utilizar.

A continuación se describe la constitución de las tablas utilizadas en el proyecto.

Cuadro 90. Base de datos Gestión, tabla Comentario

Tabla: Comentario	
Atributo	Descripción
Texto [String]	Almacena el detalle del comentario ingresado por el usuario
controlId [bigint]	Asociación con el control al que se le hace el comentario
usuarioId [bigint]	Asociación con el usuario que realiza el comentario
Clave Primaria	comentariold [bigint]

Fuente: Autor

Cuadro 91. Base de datos Gestión, tabla Control

Tabla: Control	
Atributo	Descripción
aplicable [bit]	Campo para definir si el control es aplicable o no
implementada [bit]	Campo para definir si el control esta implementado o no, utilizado para identificar si un control tiene asignado un plan de acción.

Cuadro 91. (Continuación)

Tabla: Control	
Atributo	Descripción
nombre [varchar(255)]	Nombre del control
nivel [int]	Define el nivel del control dentro de la estructura de controles, Ejemplo: Control, sub-control, etc.
descripcion [varchar(255)]	Descripción del control
normald [bigint]	Norma asociada al control
PCumplimiento [integer]	Porcentaje de cumplimiento del control. Es asignado automáticamente a partir del porcentaje de cumplimiento definido en los planes de acción o de los controles hijos.
Clave Primaria	controlld [bigint]

Fuente: Autor

Cuadro 92. Base de datos Gestión, tabla Norma

Tabla: Norma	
Atributo	Descripción
nombre [varchar(255)]	Nombre de la norma
version [varchar(50)]	Permite describir la versión de la norma
ano [varchar(50)]	Año de creación de la norma
original varchar(255)	Campo para describir la adquisición o legalidad de la norma
descripcion varchar(255)	Descripción de la norma.
PCumplimiento [integer]	Porcentaje de cumplimiento de la norma. Es asignado automáticamente a partir del porcentaje de cumplimiento definido en los planes de acción o de los controles hijos.
Clave Primaria	normald [bigint]

Fuente: Autor

Cuadro 93. Base de datos Gestión, tabla Observación

Tabla: Observacion	
Atributo	Descripción
Texto [varchar(255)]	Almacena el detalle de la observación ingresado por el usuario
planDeAccionId [bigint]	Asociación con el plan de acción al que se le hace la observación
Clave Primaria	observacionId [bigint]

Fuente: Autor

Cuadro 94. Base de datos Gestión, tabla PlanDeAccion

Tabla: PlanDeAccion	
Atributo	Descripción
actividad [String]	Registra la actividad del plan de acción
fechaCreacion [datetime]	Fecha de creación del plan de acción
fechaCumplimiento [datetime]	Fecha de cumplimiento del plan de acción
Código [varchar(50)]	Código plan de acción
PCumplimiento [nvarchar(50)]	Porcentaje de cumplimiento del plan de acción.

Fuente: Autor

Cuadro 95. Base de datos Gestión, tabla Rol

Tabla: Rol	
Atributo	Descripción
nombre [varchar(50)]	Nombre del rol: Administrador, Auditor, Consultor
Descripcion [String]	Descripción del rol dentro de la aplicación.
Clave Primaria	rolId [bigint]

Fuente: Autor

Cuadro 96. Base de datos Gestión, tabla Usuario

Tabla: Usuario	
Atributo	Descripción
nombre [String]	Nombre del usuario
nombreIngreso [String]	Nombre de la cuenta de usuario
Clave [String]	Clave del usuario
activo [Boolean]	Valida si el usuario está activo o no
Clave Primaria	usuarioID [bigint]

Fuente: Autor

5.2.3.8. Especificación técnica del plan de pruebas

5.2.3.8.1 Especificación del Entorno de Pruebas. El objetivo de esta tarea es la definición detallada y completa del entorno necesario para la realización de las pruebas del sistema: unitarias, de integración, de implantación y de aceptación. Para las pruebas realizadas no se realizan cambios a los requisitos del entorno de pruebas definido en la sección “Definición de Requisitos del Entorno de Pruebas” en la Especificación del plan de Pruebas descrito en el Análisis del sistema de información.

5.2.3.8.2 Especificación Técnica de Niveles de Prueba. El objetivo de esta tarea es el diseño detallado de los distintos niveles de prueba, especificados en el plan de pruebas elaborado en el proceso Análisis del Sistema de Información.

Los tipos de pruebas de aceptación del sistema a realizar son:

- Pruebas unitarias. Se determina realizar las pruebas unitarias para el componente “Implementar Control”, con el fin de validar la capacidad del sistema para realizar mediciones del porcentaje de cumplimiento al momento de implementar un control por parte del usuario “Consultor” y validar como este se refleja en la implementación general de la norma.

Cuadro 97. Pruebas unitarias del sistema

Prueba	Componente	Objetivo
PU-1	Implementar Control	Validar que el porcentaje de cumplimiento ingresado por el usuario Consultor al momento de implementar el control se asigne correctamente.
PU-2	Implementar Control	Validar que de forma automática la aplicación calcule el promedio de los sub-controles asociados a un mismo control, reflejando el porcentaje de cumplimiento del control padre.
PU-3	Implementar Control	Validar que de forma automática la aplicación calcule el promedio de los controles principales (De nivel 1) de la norma implementada, reflejando el porcentaje de cumplimiento general.

Fuente: Autor

De acuerdo a las pruebas descritas anteriormente, se describen los detalles de las mismas a continuación:

Cuadro 98. Detalle prueba unitaria PU-1.

Prueba	PU-1
Objetivo	Validar que el porcentaje de cumplimiento ingresado por el usuario Consultor al momento de implementar el control se asigne correctamente.
Entrada	Porcentaje de cumplimiento asignado manualmente por el usuario "Consultor" al momento de implementar un control a través de la creación de un plan de acción.
Salida	<ul style="list-style-type: none"> El control debe tomar como valor de cumplimiento del control el valor asignado por el usuario consultor en un plan de acción asociado.
Condiciones	Ninguna
Pre-requisitos	Debe existir una norma. Debe existir un control creado el cual debe estar asociado a la norma.

Cuadro 98. (Continuación)

Tareas a realizar	
<ol style="list-style-type: none"> 1. Se abre la aplicación desde una cuenta con el rol consultor. 2. Se selecciona el menú “Plan de acción” de la aplicación. 3. Se crea un plan de acción y se asocia a un control creado previamente. 4. Se escribe el porcentaje de cumplimiento a criterio del usuario consultor. 5. Se guardan los cambios realizados sobre el control. 	

Fuente: Autor

Cuadro 99. Detalle prueba unitaria PU-2.

Prueba	PU-2
Objetivo	Validar que de forma automática la aplicación calcule el promedio de los sub-controles asociados a un mismo control, reflejando el porcentaje de cumplimiento del control padre.
Entrada	Porcentaje de cumplimiento asignado manualmente por el usuario “Consultor” al momento de implementar un control.
Salida	El control principal de nivel superior del que dependen sub-controles asociados, debe calcular de forma automática su valor de cumplimiento de acuerdo al valor de cumplimiento asignado a cada uno de los sub-controles que dependen de él.
Condiciones	El control principal de nivel superior debe tener sub-controles de nivel inferior asociados.
Pre-requisitos	Cada sub-control de último nivel debe tener asignado un porcentaje de cumplimiento definido a través de su plan de acción asociado.
Tareas a realizar	
<ol style="list-style-type: none"> 1. Se abre la aplicación desde una cuenta con el rol consultor. 2. Se selecciona la norma a trabajar 3. Se selecciona el control. 4. El porcentaje de cumplimiento del control de nivel superior debe ser calculado automáticamente por la aplicación teniendo en cuenta las modificaciones realizadas en los pasos anteriores. El cálculo automático realiza un promedio de los sub-controles que cuentan con un porcentaje de cumplimiento ya definido a través de un plan de acción asociado. 	

Fuente: Autor

Cuadro 100. Detalle prueba unitaria PU-3.

Prueba	PU-3
Objetivo	Validar que de forma automática la aplicación calcule el promedio de los controles principales (De nivel 1) de la norma implementada, reflejando el porcentaje de cumplimiento general.
Entrada	Porcentaje de cumplimiento asignado manualmente por el usuario "Consultor" al momento de implementar plan de acción asociado un control de último nivel.
Salida	La norma implementada debe calcular de forma automática su valor de cumplimiento de acuerdo al valor de cumplimiento de cada control de nivel superior asociados a la norma.
Condiciones	Los controles de nivel uno de la norma deben tener asignado un valor de cumplimiento.
Pre-requisitos	Cada sub-control de nivel inferior debe tener asignado un porcentaje de cumplimiento.
Tareas a realizar	
<ol style="list-style-type: none"> 1. Se abre la aplicación desde una cuenta con el rol consultor. 2. Se selecciona la norma a consultar. 3. El porcentaje de cumplimiento del control de nivel superior debe ser calculado automáticamente por la aplicación teniendo en cuenta las modificaciones realizadas en los pasos anteriores. El cálculo automático realiza un promedio de los sub-controles que cuentan con un porcentaje de cumplimiento ya definido. 4. La norma implementada debe calcular de forma automática su valor de cumplimiento de acuerdo al valor de cumplimiento de cada control de nivel superior asociados a la norma. 	

Fuente: Autor

- Pruebas del sistema. Para las pruebas del sistema se busca validar las diferencias entre requerimientos y la solución desarrollada. A continuación se describen los controles a implementar con base en las siguientes estrategias:

1. Probar la navegación, ingreso de datos, procesamiento y recuperación.
2. Metodología de caja negra, orientado a pruebas desde la GUI.

A continuación se describen las pruebas de sistema a realizar.

Cuadro 101. Pruebas del sistema

Prueba	Casos de uso	Objetivo
PS-1	Gestión de usuarios Gestión de normas Gestión de controles	Ingresar norma en el sistema junto con sus controles.
PS-2	Gestión de usuarios Gestión de normas Gestión de controles	Implementar norma, controles y sub-controles de la norma seleccionada.
PS-3	Gestión de usuarios Gestión de plan de acción Gestión de normas Gestión de controles	Validar el cálculo automático de implementación general de la norma seleccionada.
PS-4	Gestión de usuarios Gestión de normas Gestión de controles	Consulta de normas ingresadas en el sistema.
PS-5	Gestión de usuarios Gestión de plan de acción Gestión de normas Gestión de controles Gestión de comentarios	Ingreso de comentarios a los controles implementados desde el rol "Consultor"
PS-6	Gestión de usuarios Gestión de plan de acción Gestión de normas Gestión de controles Gestión de observaciones	Ingreso de observaciones a los controles implementados desde el rol "Auditor"

Fuente: Autor

De acuerdo a las pruebas descritas anteriormente, se describen los detalles de las mismas a continuación:

Cuadro 102. Detalle prueba de sistema PS-1.

Prueba	PS-1
Objetivo	Ingresar norma en el sistema junto con sus controles.
Entrada	Nombre, versión, año, licencia, y descripción de la norma. Controles de la norma
Salida	<ul style="list-style-type: none"> El sistema permite consultar, modificar e implementar la norma de acuerdo al rol del usuario.
Condiciones	<ul style="list-style-type: none"> El usuario administrador puede agregar y modificar la norma.
Pre-requisitos	<ul style="list-style-type: none"> No hay pre-requisitos.
Tareas a realizar	
<ol style="list-style-type: none"> Se abre la aplicación desde una cuenta con el rol administrador. Se selecciona el menú norma. Se presiona el botón Crear para crear norma. Se describen los detalles de la norma. Se guarda la norma. Se selecciona el menú Control. Se selecciona el botón Crear para cada control nuevo. Se describen los detalles del control. Se asocia el control a la norma. Si el control es de nivel 2 o superior se debe seleccionar el control asociado. Se guardan los cambios de cada control. 	

Fuente: Autor

Cuadro 103. Detalle prueba de sistema PS-2.

Prueba	PS-2
Objetivo	Implementar norma, controles y sub-controles de la norma seleccionada.
Entrada	Selección de la norma a implementar. Implementación de controles y definición de porcentajes de cumplimiento.
Salida	<ul style="list-style-type: none"> La norma es implementada y se refleja el porcentaje de cumplimiento de acuerdo al valor de cumplimiento asignado por el usuario consultor.
Condiciones	<ul style="list-style-type: none"> El usuario consultor puede implementar y consultar la norma.

Cuadro 103. (Continuación)

Pre-requisitos	<ul style="list-style-type: none"> La norma y sus controles deben ser ingresados por el usuario administrador.
Tareas a realizar	
<ol style="list-style-type: none"> Se abre la aplicación desde una cuenta con el rol administrador. Se selecciona el menú plan de acción para empezar a implementar la norma. Se crea el plan de acción, fecha de cumplimiento del plan de acción, y control asociado. El paso anterior debe realizarse para cada control de la norma. Se selecciona el botón Guardar para plan de acción definido. 	

Fuente: Autor

Cuadro 104. Detalle prueba de sistema PS-3.

Prueba	PS-3
Objetivo	Consulta de normas ingresadas en el sistema.
Entrada	Selección de la norma a consultar. Selección de controles a consultar.
Salida	<ul style="list-style-type: none"> La norma y sus controles son mostrados en pantalla en modo consulta.
Condiciones	<ul style="list-style-type: none"> El usuario administrador, consultor y auditor puede consultar la norma.
Pre-requisitos	<ul style="list-style-type: none"> La norma y sus controles deben ser ingresados por el usuario administrador.
Tareas a realizar	
<ol style="list-style-type: none"> Se abre la aplicación desde una cuenta con rol administrador, consultor y auditor. Se selecciona el menú norma. Se selecciona el botón ver detalle. Se consulta los controles de la norma. 	

Fuente: Autor

Cuadro 105. Detalle prueba de sistema PS-4.

Prueba	PS-4
Objetivo	Validar el cálculo automático de implementación general de la norma seleccionada.
Entrada	Selección de la norma a consultar. Selección de controles a implementar y asignar el valor de implementación a criterio del usuario consultor.
Salida	El porcentaje con implementación de controles de nivel superior y el porcentaje general de implementación de la norma es actualizado automáticamente.
Condiciones	El cálculo automático de implementación de los controles de nivel superior depende del porcentaje de cumplimiento definido por el usuario consultor en los controles de nivel inferior a través de su plan de acción asociado. El cálculo automático de implementación general de la norma depende del porcentaje de cumplimiento definido por el usuario consultor en los controles de nivel inferior y del cálculo automático generado por la aplicación en los controles de nivel superior.
Pre-requisitos	La norma y sus controles deben ser ingresados por el usuario administrador, y los valores de implementación de los controles de nivel inferior ser diligenciados por el usuario consultor.
Tareas a realizar	
<ol style="list-style-type: none"> 1. Se abre la aplicación desde una cuenta con rol consultor. 2. Se selecciona el menú norma. 3. Se selecciona el botón ver detalle de la norma a consultar. 4. Se consulta el campo Porcentaje de cumplimiento. 	

Fuente: Autor

Cuadro 106. Detalle prueba de sistema PS-5.

Prueba	PS-5
Objetivo	Ingreso de comentarios a los controles implementados desde el rol "Consultor".
Entrada	Selección de la norma a implementar. Selección de controles a implementar y asignar comentario a criterio del usuario consultor.
Salida	El control tiene asociado un comentario.
Condiciones	El control a comentar ya debió ser implementado por el usuario consultor.

Cuadro 106. (Continuación)

Pre-requisitos	La norma y sus controles deben ser ingresados por el usuario administrador, y los valores de implementación de los controles de nivel inferior ser diligenciados por el usuario consultor.
Tareas a realizar	
<ol style="list-style-type: none"> 1. Se abre la aplicación desde una cuenta con rol consultor. 2. Se selecciona el menú comentario. 3. Se crea comentario seleccionando la norma y control asociado. 4. Se guarda comentario. 5. Se observa la creación del comentario en la gestión de comentarios. 	

Fuente: Autor

Cuadro 107. Detalle prueba de sistema PS-6.

Prueba	PS-6
Objetivo	Ingreso de observaciones a los controles implementados desde el rol "Auditor".
Entrada	Selección de la norma a implementar. Selección de controles a consultar y asignar comentario a criterio del usuario auditor.
Salida	El control tiene asociado una observación a través de su plan de acción.
Condiciones	El control a comentar ya debió ser implementado por el usuario consultor.
Pre-requisitos	La norma y sus controles deben ser ingresados por el usuario administrador, y los valores de implementación de los controles de nivel inferior ser diligenciados por el usuario consultor.
Tareas a realizar	
<ol style="list-style-type: none"> 1. Se abre la aplicación desde una cuenta con rol auditor. 2. Se selecciona el menú Observación. 3. Se selecciona el botón Crear. 4. Se escribe texto y se asocia al plan de acción. 5. Se guardan los cambios de la observación. 6. Se lista las observaciones en la gestión de observaciones. 	

Fuente: Autor

Las pruebas adicionales que puedan surgir, son realizadas de acuerdo a lo especificado en la sección de Análisis del sistema de información – Definición de las pruebas de aceptación del sistema.

5.2.3.8.3 Revisión de la Planificación de Pruebas. Las pruebas son realizadas a medida que se va desarrollando el software y se ajustan los componentes del mismo de acuerdo a los requerimientos y caso de uso establecidos.

5.2.3.9. Aprobación del diseño del sistema de información

5.2.3.9.1 Presentación y Aprobación del Diseño del Sistema de Información. El Diseño del Sistema de Información es aprobado por la empresa ALFCOM, dado que se cubren las estrategias esperadas para continuar con el desarrollo de la aplicación.

5.3. CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)

5.3.1.1. Preparación del entorno de generación y construcción. En esta actividad se busca asegurar la disponibilidad de todos los medios y facilidades para llevar a cabo la construcción del sistema de información.

5.3.1.1.1 Implantación de la Base de Datos Física o Ficheros. En el siguiente cuadro se describen los elementos de implementación de la base de datos:

Cuadro 108. Elementos de implementación de la base de datos física.

Componente	Detalle
Espacio de almacenamiento	Capacidad en disco mínima para la base de datos: 20 GB
Dispositivos físicos a emplear	Servidor con aplicación Microsoft SQL Server 2012 Express instalada.
Tipo de registro físico	Las bases de datos son almacenadas por el SGBD a través del lenguaje de consulta estructurado SQL.
Zona de desbordamiento	Capacidad de espacio libre mínimo del 10%.

Fuente: Autor

Se debe cargar inicialmente una copia de la base de datos a través del menú importar del SGBD Microsoft SQL Server 2012. Se anexa copia de la base de datos que debe ser cargada antes de iniciar la aplicación Anexo 2.

5.3.1.1.2 Preparación del Entorno de Construcción. En esta tarea se prepara el entorno de construcción de los componentes del sistema de información. Esta tarea se encuentra detallada en la anterior sección: Identificación del entorno tecnológico – Entorno de producción. Se describen los componentes necesarios a través de la siguiente tabla:

Cuadro 109. Componentes del entorno de construcción.

Componente	Detalle
Librerías	NHibernate.dll Nhibernate.Mapping.Attributes.dll System.dll Mscorlib.dll System.web.dll System.data.dll System.core.dll System.Configuration.dll

Cuadro 109. (Continuación)

Componente	Detalle
Herramientas	Microsoft Visual Studio 2012 Express Microsoft SQL Server 2012 Express Navegador Web Microsoft Internet Information Services 7 Pentaho Report Designer Pentaho Business Analytics
Puestos de trabajo	2 Computadores portátiles para los ingenieros encargados del desarrollo de la aplicación. Estos computadores permiten simular la ejecución de la aplicación, como si este fuera implementado en el servidor de producción.

Fuente: Autor

No se identifican cambios al entorno de construcción de la aplicación inicialmente planteado.

5.3.1.2. Generación del código de los componentes y procedimientos. En esta tarea se realiza la codificación de los componentes del sistema de información con base en las especificaciones de construcción obtenidas en el proceso de Diseño del Sistema de Información.

5.3.1.2.1 Generación del Código de Componentes. El código desarrollado para la aplicación propuesta se encuentra detallado en el Anexo 3, el cual se encuentra definido bajo las siguientes Librerías:

Cuadro 110. Código de componentes

Librería	Código
VSComun	ManejadoresDatos NHibernateUtils Excepciones Modelo Enums

Cuadro 110. (Continuación)

Librería	Código
ADComun	Enums Constantes Excepciones Modelo
ADNucleo	Fachada Manejadores

Fuente: Autor

5.3.1.2.2 Generación del Código de los Procedimientos de Operación y Seguridad. Los manuales técnicos de operación, seguridad y administración del sistema son descritos en el Anexo 4.

5.3.1.3. Ejecución de las pruebas unitarias. A través de esta actividad se realizan las pruebas unitarias del componente “implementar control” del sistema de información, una vez codificado, con el objeto de comprobar que su estructura es correcta y que se ajustan a la funcionalidad establecida.

5.3.1.3.1 Preparación del Entorno de las Pruebas Unitarias. Para el desarrollo de las pruebas unitarias se trabaja sobre un entorno virtualizado sobre los equipos portátiles de los desarrolladores del proyecto. El entorno cumple con los requisitos descritos anteriormente en la Definición de Requisitos del Entorno de Pruebas.

5.3.1.3.2 Resultado de la Realización de las Pruebas Unitarias. A continuación se describe el resultado de realizar 3 intentos de las pruebas unitarias de acuerdo a las condiciones definidas en cada una de ellas.

Se realizan las pruebas unitarias para el componente “Implementar Control”, con el fin de validar la capacidad del sistema para realizar mediciones del porcentaje de cumplimiento al momento de implementar un control por parte del usuario “Consultor” y validar como este se refleja en la implementación general de la norma.

Cuadro 111. Resultado de la realización de pruebas unitarias.

Prueba	Componente	Resultado	Valoración
PU-1	Implementar Control	El porcentaje de cumplimiento ingresado por el usuario Consultor al momento de implementar el control se asigna correctamente.	Exitoso
PU-2	Implementar Control	De forma automática la aplicación calcula el promedio de los sub-controles asociados a un mismo control, reflejando el porcentaje de cumplimiento del control padre.	Exitoso
PU-3	Implementar Control	De forma automática la aplicación calcula el promedio de los controles principales (De nivel 1) de la norma implementada, reflejando el porcentaje de cumplimiento general.	Exitoso

Fuente: Autor

5.3.1.4. Ejecución de las pruebas del sistema. A través de esta actividad se realizan las pruebas del sistema y así comprobar la integración del sistema de información globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de sistemas de información con los que se comunica.

5.3.1.4.1 Preparación del Entorno de las Pruebas del Sistema. Para el desarrollo de las pruebas unitarias se trabaja sobre un entorno virtualizado sobre los equipos portátiles de los desarrolladores del proyecto. El entorno cumple con los requisitos descritos anteriormente en la Definición de Requisitos del Entorno de Pruebas.

5.3.1.4.2 Resultado de la Realización de las Pruebas del Sistema. A continuación se describe el resultado de realizar 3 intentos de las pruebas del sistema de acuerdo a las condiciones definidas en cada una de ellas.

Se validan las diferencias entre requerimientos y la solución desarrollada. A continuación se describen los controles implementados con base en las siguientes estrategias:

- Probar la navegación, ingreso de datos, procesamiento y recuperación.
- Metodología de caja negra, orientado a pruebas desde la GUI.

Cuadro 112. Resultado de la realización de pruebas unitarias.

Prueba	Casos de uso	Resultado	Valoración
PS-1	Gestión de usuarios Gestión de normas Gestión de controles	Ingreso de norma en el sistema junto con sus controles.	Exitoso
PS-2	Gestión de usuarios Gestión de normas Gestión de controles	Implementación de norma, controles y sub-controles de la norma seleccionada.	Exitoso
PS-3	Gestión de usuarios Gestión de plan de acción Gestión de normas Gestión de controles	Se validó el cálculo automático de implementación general de la norma seleccionada.	Exitoso
PS-4	Gestión de usuarios Gestión de normas Gestión de controles	Se consulta las normas ingresadas en el sistema.	Exitoso
PS-5	Gestión de usuarios Gestión de plan de acción Gestión de normas Gestión de controles Gestión de comentarios	Se Ingresaron comentarios a los controles implementados desde el rol "Consultor"	Exitoso
PS-6	Gestión de usuarios Gestión de plan de acción Gestión de normas Gestión de controles Gestión de observaciones	Se Ingresaron observaciones a los controles implementados desde el rol "Auditor"	Exitoso

Fuente: Autor

5.3.1.5. Elaboración de los manuales de usuario. A través de esta actividad se elabora la documentación de usuario final de acuerdo a los requisitos establecidos en la tarea Especificación de Requisitos de Documentación de Usuario (DSI 11.1), y recogidos en el catálogo de requisitos.

5.3.1.5.1 Elaboración de los Manuales de Usuario. El manual de usuario de la aplicación Information Security Status ISS se encuentra descrito en el Anexo 5 del presente documento.

6. DESARROLLO, RESULTADOS Y APORTES

La aplicación *Information Security Status – ISS* está instalada en la empresa Alcom S.A. como parte de una mejora corporativa y de calidad, se encuentra desarrollada con tecnología de última generación y con los mejores estándares de desarrollo de aplicaciones WEB, para su creación se utilizaron herramientas como Visual studio 2013, MS SQL server 2012, NHibernate, Pentaho, entre otros, y como lenguajes de programación principalmente, C Sharp (C#), HTML, JavaScript y HQL.

Después de realizar el sistema de información planteado, se describen los resultados obtenidos a continuación:

1. Se creó la aplicación denominada Information Security Status (ISS) enfocada en la implementación de normas de seguridad de la información, orientada a la medición automática y en tiempo real de un Sistema de Gestión de Seguridad de la Información SGSI.
2. Se realiza instalación de la aplicación en la empresa Alcom S.A., con el fin de iniciar la implementación de los controles definidos por la Superintendencia Financiera de Colombia bajo su circular 052 de 2007, los cuales están dirigidos a empresas "intermediarias" que no están sujetas actualmente a la vigilancia de la superintendencia financiera, pero que en desarrollo de sus actividades, tienen acceso a información confidencial de las entidades financieras y de sus clientes.
3. Se plantea un enfoque de gestión de normas de seguridad de la información a través de una aplicación web, que busca centralizar las actividades asociadas con este proceso permitiendo dar a conocer en todo momento los resultados obtenidos a la alta gerencia de la organización.
4. La aplicación desarrollada tiene un enfoque dirigido a la implementación de cualquier tipo de norma, permitiendo a futuro convertirse en un sistema centralizado de administración de estándares de seguridad de la información.
5. La medición en la implementación de diferentes normas de seguridad de la información se ve reflejada en los reportes generados por la aplicación de forma automática, garantizando información oportuna a la alta gerencia de la organización para la toma de decisiones.

A través del desarrollo del presente proyecto se espera una mejora en el control de la implementación de sistemas de gestión de seguridad de la información en la

empresa Alfcom S.A., además de una reducción de $\frac{2}{3}$ en los costos que este proceso le cuesta a la empresa, donde actualmente se tiene establecido un presupuesto mensual de \$ 1.000.000 (un millón de pesos) mensuales. Teniendo en cuenta los costos que requiere la puesta en marcha del presente sistema de información en la empresa Alfcom S.A., se estima un retorno de la inversión para el 2 año de operación del mismo.

7. CONCLUSIONES

En el presente proyecto se desarrolló un sistema de información orientado a la implementación de normas de gestión de seguridad de la información como ISO 27001, Circular externa 052 de 2007, etc. La función principal del sistema permite medir constantemente el nivel de implementación de cada una de las normas incluidas en el sistema, garantizando que la alta gerencia conozca en todo momento el nivel de madurez de este proceso.

El sistema de información fue creado inicialmente para ser utilizado en la empresa colombiana de desarrollo de software financiero ALFCOM S.A., donde se observó la necesidad de llevar a cabo la implementación de un Sistema de Gestión de Seguridad de la Información (SGSI), teniendo en cuenta las diferentes exigencias promulgadas por la Superintendencia Financiera de Colombia bajo su Circular 052 del año 2007, dirigidas a empresas "intermediarias" que no están sujetas actualmente a la vigilancia de la superintendencia financiera, pero que en desarrollo de sus actividades, tienen acceso a información confidencial de las entidades financieras y de sus clientes.

Para cumplir con la meta planteada se optó por utilizar las herramientas de desarrollo propuestas por la empresa ALFCOM S.A., dado que esta cuenta con el licenciamiento y los equipos necesarios para planear, desarrollar, implementar y poner en marcha el sistema de información propuesto.

Entre las ventajas identificadas en el desarrollo de este proyecto, se registra un considerable beneficio tangible con relación a los recursos necesarios para llevar a cabo a la implementación de un Sistema de Gestión de seguridad de la información en la empresa ALFCOM S.A., donde inicialmente se tenía planteado un presupuesto mensual de \$ 1.000.000 (un millón de pesos colombianos) como parte de los honorarios destinados al personal encargado de la implementación, logrando reducir estos costos a una tercera parte. De esta manera los costos por mes respecto al salario designado para esta actividad se reducen a \$333.333 (treientos treinta y tres mil treientos treinta y tres pesos colombianos), permitiendo utilizar las 2 terceras partes del tiempo del personal designado en otras tareas del negocio. (Ver Estudio de la inversión).

Teniendo en cuenta lo anterior y descontando los costos asociados a la creación e implementación del presente proyecto en la empresa ALFCOM S.A., se proyecta un retorno de la inversión en el segundo año de la implementación del sistema de información, con un beneficio total de \$8.500.000 (ocho millones quinientos mil pesos colombianos).

Adicionalmente, se destacan los beneficios intangibles identificados en el presente proyecto, los cuales permitirán a la empresa ALFCOM S.A.: i) medir la implementación de una norma a medida que se realiza, ii) utilizar la herramienta como mecanismo centralizado de información con un concepto de medición constante, iii) permitir la toma de decisiones de forma eficaz y eficiente, y iv) la información será mostrada a los organismos de auditoría de forma centralizada.

Enfoques futuros:

Respecto al desarrollo de la presente aplicación, se ha determinado diferentes enfoques que pueden tomarse respecto a la mejora y puesta en venta del producto aquí planteado bajo las siguientes premisas:

- El soporte de la aplicación está planteado ofrecerse a las empresas interesadas con un costo anual de \$ 1.000.000 (un millón de pesos colombianos), teniendo en cuenta que la estrategia inicial de distribución de la aplicación será de forma gratuita, permitiendo abarcar un mercado más amplio y por tanto la venta equitativa de su soporte técnico.
- El sistema permite ser adaptable a diferentes normas, por lo que permitirá la creación de contenido que puede ofrecerse a las empresas interesadas como parte del soporte a un precio adicional.

Finalmente se puede concluir que el presente proyecto permitió a sus autores identificar varias necesidades presentadas por las empresas al momento de implementar un sistema de gestión de seguridad de la información, logrando definir una solución adsequible y de fácil uso que contribuirá al desarrollo profesional de sus creadores, como a la cultura que necesitan las empresas de hoy en día para cumplir sus objetivos, teniendo en cuenta uno de los activos considerados actualmente como de los más importantes, la información.

8. BIBLIOGRAFÍA

COLOMBIA, SUPERINTENDENCIA FINANCIERA DE COLOMBIA. Circular Externa 042 de 2012 (24, octubre, 2012). Por medio de la cual se incorporan algunas modificaciones al Capítulo Décimo Segundo del Título Primero de la Circular Básica Jurídica, en materia de requerimientos mínimos de seguridad y calidad para la realización de operaciones. 2012. P 1-3.

COLOMBIA, SUPERINTENDENCIA FINANCIERA DE COLOMBIA. Circular Externa 007 de 1996 (19, enero, 1996). Esta Circular reúne las diferentes instrucciones en materia jurídica emitidas por la Superintendencia Bancaria y que a la fecha se encuentran vigentes. 1996. P 1-3.

COLOMBIA, SUPERINTENDENCIA FINANCIERA DE COLOMBIA. Circular Externa 052 de 2007 (25, octubre, 2007). Requerimientos mínimos de seguridad y calidad en el manejo de información a través de medios y canales de distribución de productos y servicios para clientes y usuarios. 2007. P 1-3.

COLOMBIA, Congreso de Colombia. Ley 1273 de 2009 (05, enero, 2009). Por medio de la cual se modifica el Código Penal, se crea un nuevo bien jurídico tutelado - denominado "de la protección de la información y de los datos"- y se preservan integralmente los sistemas que utilicen las tecnologías de la información y las comunicaciones, entre otras disposiciones. 2009. P 1-3.

COLOMBIA, Congreso de Colombia. Ley 1266 de 2008 (31, diciembre, 2008). Por la cual se dictan las disposiciones generales del hábeas data y se regula el manejo de la información contenida en bases de datos personales, en especial la financiera, crediticia, comercial, de servicios y la proveniente de terceros países y se dictan otras disposiciones. 2008. P 1-3.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. Tecnología de la información - Técnicas de seguridad - Sistemas de gestión de seguridad de la información - Requerimientos. ISO/IEC 27001. 1 ed. ISO, 2005. 9 p.

INSTITUTO COLOMBIANO DE NORMALIZACIÓN Y CERTIFICACIÓN. Presentación de tesis, trabajos de grado y otros trabajos de investigación. NTC-1486. Bogotá D.C.: El instituto, 2008. 1-41 p.

INSTITUTO COLOMBIANO DE NORMALIZACIÓN Y CERTIFICACIÓN. Referencias documentales para fuentes de información electrónicas. NTC-4490. Bogotá D.C.: El instituto, 1998. 1-27 p.

INSTITUTO COLOMBIANO DE NORMALIZACIÓN Y CERTIFICACIÓN. Referencias bibliográficas, contenido, forma y estructura. NTC-5613. Bogotá D.C.: El instituto, 2008. 1-38 p.

PYTEL, Pablo. Método de estimación de esfuerzo para proyectos de explotación de información, herramienta para su validación. Buenos Aires: Universidad Politécnica de Madrid Instituto Tecnológico de Buenos Aires. Ingeniería de Software, 2011, 2-491.

CEEI, Fundación de Albacete. Securia SGSI [Online]. España: CEEI SEC, 2013 [citado 15 Diciembre 2013]. Disponible en internet: <<http://www.securia.es>>.

AUDISEC. GlobalSGSI [Online]. España: Audisec, 2014 [citado 2 Agosto 2014]. Disponible en internet: <<http://news.audisec.es/es/productos-globalsuite/globalsgsi.html>>.

ISO27000.es. ISO 27000 [Online]. España: ISO27000.es, 2014 [citado 12 Mayo 2014]. Disponible en internet :< <http://www.iso27000.es/iso27000.html> >.

COVINOC. Circular 052 y su impacto sobre la tecnología de las Cobranzas en su empresa [Online]. Colombia: Financial Systems Company, 2014 [citado 08 Enero 2014]. Disponible en internet :<http://www.covinoc.com/recursos/pdf_oscarangel.pdf>.

MICROSOFT. Microsoft SQL server [Online]. Microsoft Technet, 2014 [citado 02 Agosto 2014]. Disponible en internet: <<http://technet.microsoft.com/es-es/library/bb545450.aspx>>.

SUPERINTENDENCIA FINANCIERA DE COLOMBIA [Online]. Superintendencia Financiera de Colombia, 2013 [citado 15 Noviembre de 2014]. Disponible en internet:
<<http://www.superfinanciera.gov.co/Normativa/PrincipalesPublicaciones/boletinejboletin3311/seguridadycalidad.html> >.

9. ANEXOS

Anexo 1. Lista de pruebas.

# RF	Descripción	Elemento Afetado	Nivel Severidad	Fecha Identificación	Fecha Corrección	Estado
1	Dentro de las opciones falta el menú "Observación", que es lo que el auditor comenta en cada plan de acción.	Caso de uso	4	7/06/2014	1/07/2014	OK
2	Error al agregar norma con una descripción extensa (más de dos renglones)	Base de datos	3	7/06/2014	9/06/2014	OK
3	El plan de acción debe tener la presentación de la imagen propuesta.	Modulo Interfaz	4	7/06/2014	12/09/2014	OK
4	EL plan de acción debe permitir escribir el porcentaje de cumplimiento del mismo	Caso de uso	4	7/06/2014	12/09/2014	OK
5	Error al agregar Nuevo usuario a la aplicación, la aplicación no vuelve a ejecutarse y es necesario restaurar la base de datos para continuar con la ejecución de la aplicación.	Nodo	1	8/06/2014	12/09/2014	OK
6	Error al eliminar plan de acción	Caso de uso	3	27/06/2014	12/09/2014	OK
7	La clase Rol sigue sin estar relacionada en el diagrama de clases	Caso de uso	4	27/06/2014	12/09/2014	OK
8	Es necesario modificar la relación de la tabla "Observacion" de la base de datos. Dado que esta tiene que	Caso de uso	4	30/06/2014	12/09/2014	OK

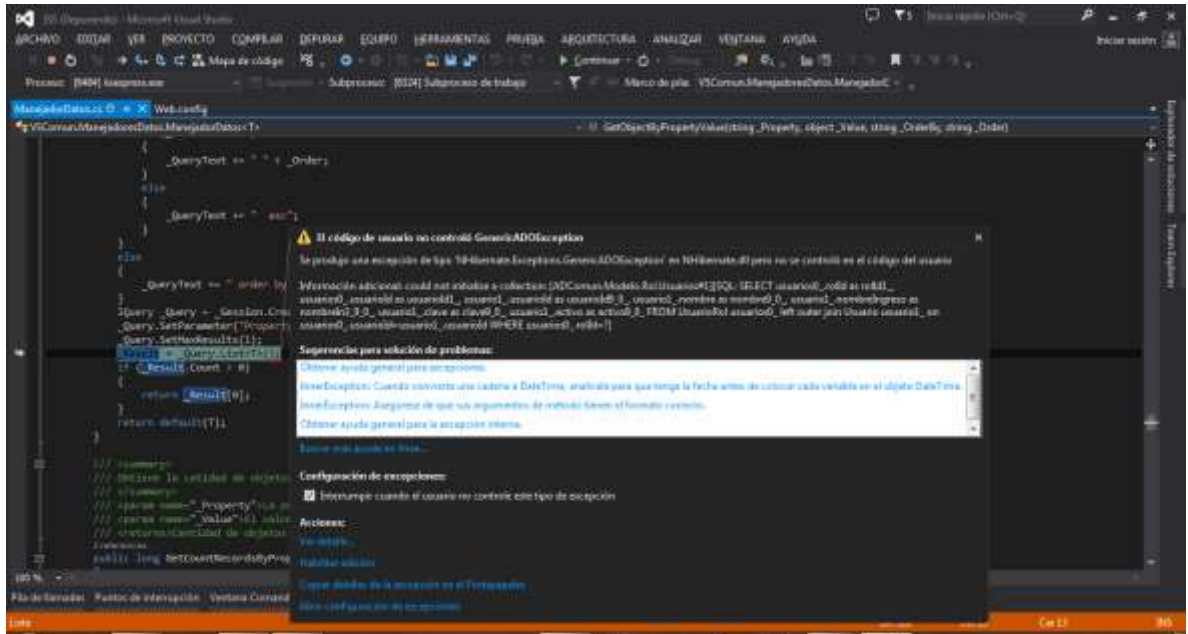
	ser similar a la que tiene la tabla "Comentario", dado que tanto comentarios como observaciones deben ser realizados sobre un control.					
9	Es necesario que al agregar un nuevo plan de acción, este permita asociarse con un control ya implementado. Para realizar esta tarea es necesario: 1. Crear un campo en la tabla "PlanDeAccion" que permita asociar el control. 2. La interfaz gráfica de "Control" debe tener un campo que permita crear un plan de acción quedando asociado automáticamente al control seleccionado.	Caso de uso	4	30/06/2014	12/09/2014	OK
10	Error al agregar control con una descripción extensa	Base de datos	3	30/06/2014	13/07/2014	OK
11	Aumentar los caracteres ingresados en el campo nombre de la tabla Usuario a 255	Base de datos	4	6/07/2014	10/07/2014	OK
12	Las tablas de control y observación deben tener un campo fecha para registrar el momento en que es ingresado el valor del campo	Base de datos	4	10/07/2014	12/09/2014	OK
13	Error al editar un control ya creado. No aparece la información del control seleccionado.	Caso de uso	4	16/07/2014	12/09/2014	OK
14	Error al crear un plan de acción cuando la norma seleccionada no tiene controles creados asociados a la norma.	Caso de uso	4	4/08/2014	12/09/2014	OK

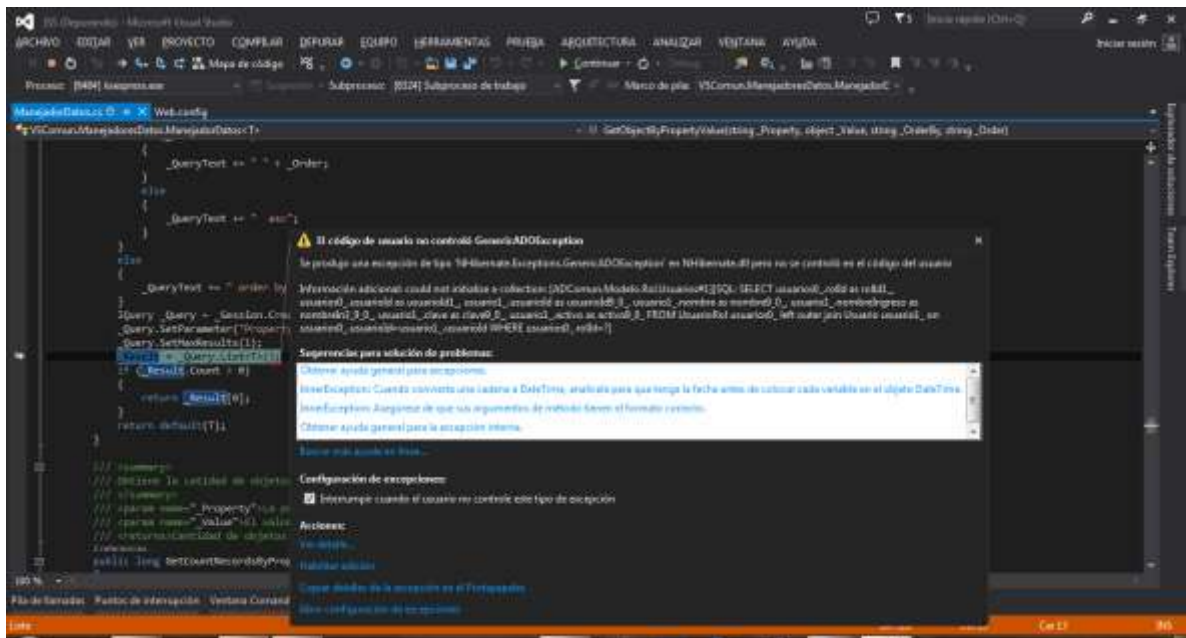
15	Error al actualizar una norma con una descripción mayor a 1 renglon	Caso de uso	4	4/08/2014	12/09/2014	OK
16	Error al agregar un código de varios caracteres no es permitido en el plan de acción, es necesario modificar el tipo de valor del campo código por uno de más caracteres.	Caso de uso	4	7/08/2014	12/09/2014	OK
17	Error al agregar un plan de acción sin seleccionar la fecha de cumplimiento, es necesario cambiar el error mostrado o asignar una fecha por defecto si no se selecciona esta.	Caso de uso	4	9/08/2014	12/09/2014	OK
18	Error al eliminar un plan de acción, no es posible eliminarlo.	Caso de uso	4	9/08/2014	12/09/2014	OK

Pruebas realizadas sobre la aplicación

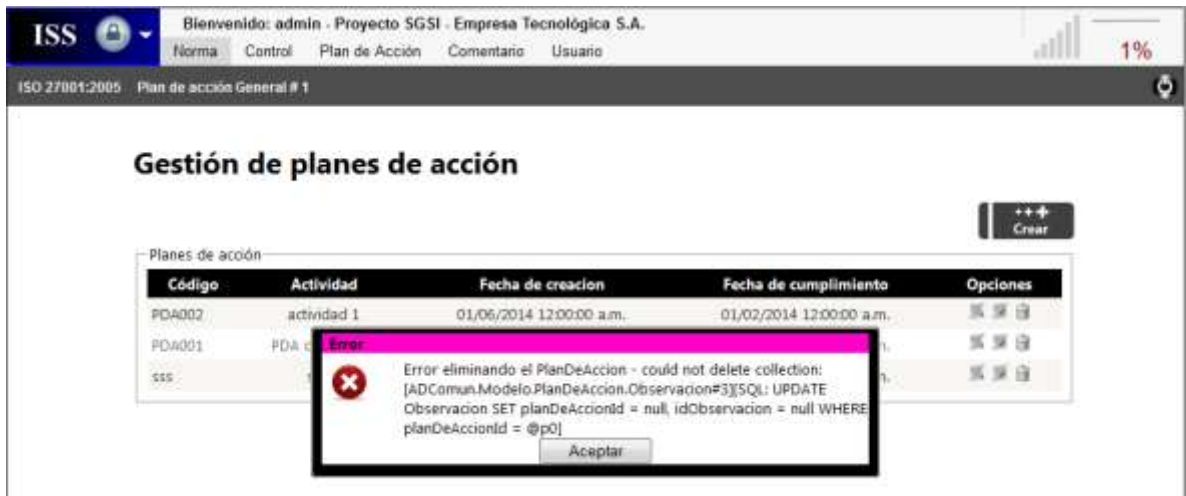
Nivel de Severidad	Descripción	Ejemplo
1	Se detiene la ejecución de la aplicación y no es posible volver utilizarla.	Error crítico que impide ejecutar nuevamente la aplicación.
2	La función a realizar no es ejecutada y la aplicación es detenida.	La función ejecutada genera errores generando que la aplicación se cierre. La aplicación puede ser abierta nuevamente.
3	Las funciones no pueden ser ejecutadas, pero la aplicación sigue en funcionamiento.	Se agregan datos a la aplicación y se reciben mensajes de error desconocidos. La aplicación sigue funcionando.
4	Modificación Menor	Errores menores de uso de la aplicación, presentación inconsistente de interfaz gráfica, etc.

Requisito faltante No.:	5
Severidad: 1	Se detiene la ejecución de la aplicación y no es posible continuar con su ejecución.
Descripción:	Error al agregar Nuevo usuario a la aplicación, la aplicación no vuelve a ejecutarse y es necesario restaurar la base de datos para continuar con la ejecución de la aplicación.

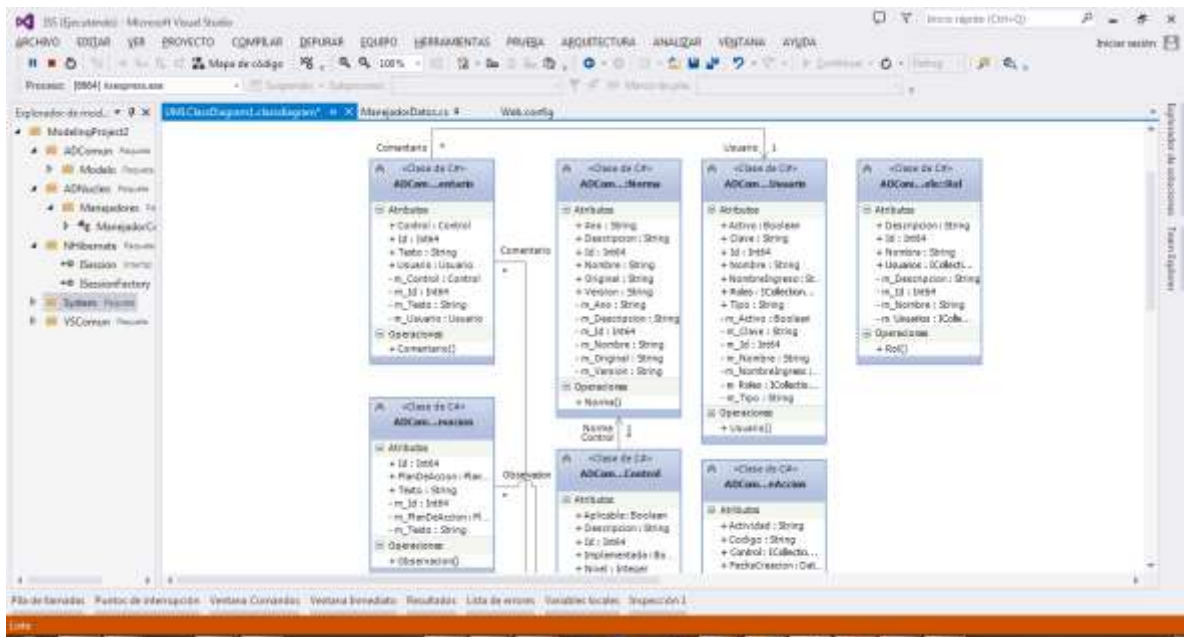




Requisito faltante No.:	6
Severidad: 3	Las funciones no pueden ser ejecutadas, pero la aplicación sigue en funcionamiento.
Descripción:	No es posible eliminar un plan de acción.



Requisito faltante No.:	7
Severidad: 3	Errores menores de uso de la aplicación, presentación inconsistente de interfaz gráfica, etc.
Descripción:	La clase Rol sigue sin estar relacionada en el diagrama de clases



Requisito faltante No.:	8
Severidad: 4	Errores menores de uso de la aplicación, presentación inconsistente de interfaz gráfica, etc.
Descripción:	Es necesario modificar la relación de la tabla "Observacion" de la base de datos. Dado que esta tiene que ser similar a la que tiene la tabla "Comentario", dado que tanto comentarios como observaciones deben ser realizadas sobre un control.



Requisito faltante No.:	9
Severidad: 4	Errores menores de uso de la aplicación, presentación inconsistente de interfaz gráfica, etc.
Descripción:	<p>Es necesario que al agregar un nuevo plan de acción, este permita asociarse con un control ya implementado.</p> <p>Para realizar esta tarea es necesario:</p> <ol style="list-style-type: none"> 1. Crear un campo en la tabla "PlanDeAccion" que permita asociar el control. 2. La interfaz gráfica de "Control" debe tener un campo que permita crear un plan de acción quedando asociado automáticamente al control seleccionado.

ISS  Bienvenido: admin - Proyecto SGSI - Empresa Tecnológica S.A. 1%

Norma Control Plan de Acción Comentario Usuario

ISO 27001:2005 Plan de acción General # 1 

Gestión de Controles

Filtro

Nombre: 

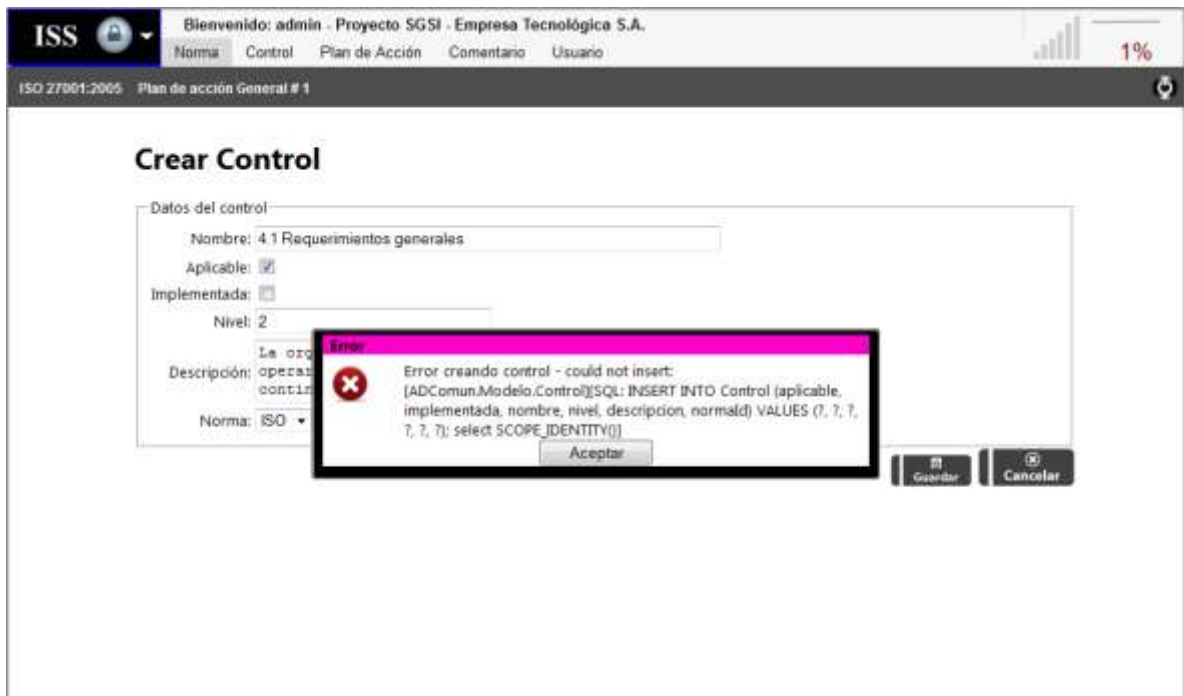
 Crear

Controles

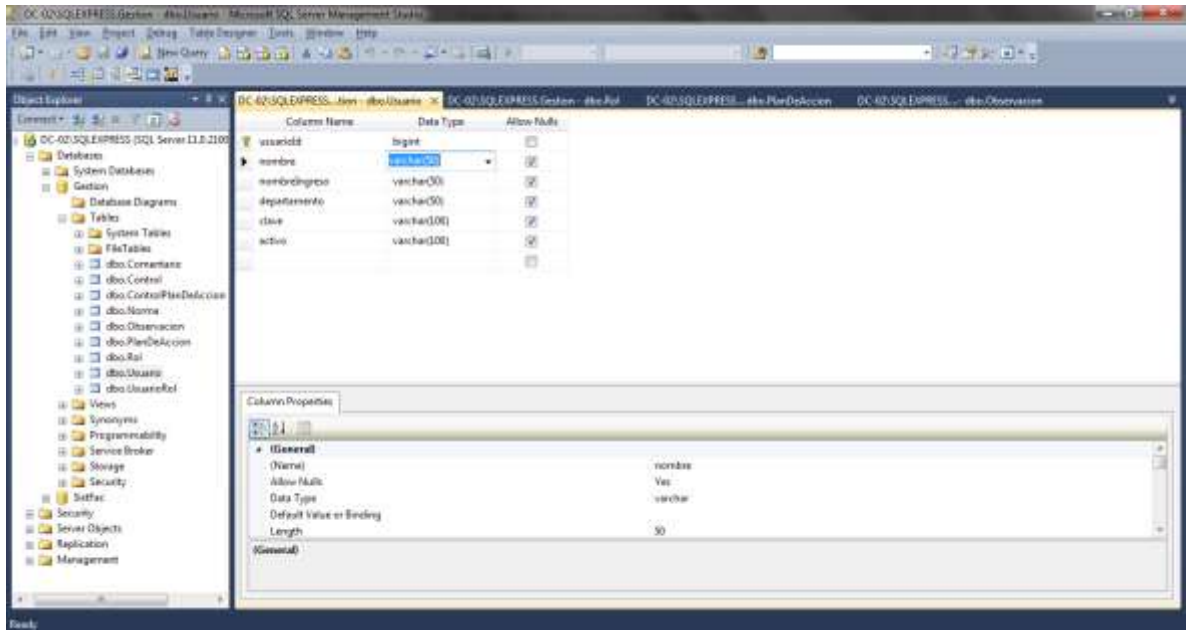
Nombre	Norma	Aplicable	Implementado	Nivel	Descripción	Opciones
c1	ISO	False	True	12	hola	  


 Campo para crear un plan de acción el cual quedará asociado automáticamente a este control, registrando este valor en la tabla PlanDeAccion


Requisito faltante No.:	10
Severidad: 4	Errores menores de uso de la aplicación, presentación inconsistente de interfaz gráfica, etc.
Descripción:	Error al agregar control con una descripción extensa (más de dos renglones)



Requisito faltante No.:	11
Severidad: 4	Errores menores de uso de la aplicación, presentación inconsistente de interfaz gráfica, etc.
Descripción:	Aumentar los caracteres ingresados en el campo nombre de la tabla Usuario a 255




Requisito faltante No.:	12
Severidad: 4	Errores menores de uso de la aplicación, presentación inconsistente de interfaz gráfica, etc.
Descripción:	Las tablas de control y observación deben tener un campo fecha para registrar el momento en que es ingresado el valor del campo

	Column Name	Data Type	Allow Nulls
	comentarioId	bigint	<input type="checkbox"/>
	texto	varchar(255)	<input type="checkbox"/>
	controlId	bigint	<input type="checkbox"/>
	usuarioId	bigint	<input type="checkbox"/>
			<input type="checkbox"/>

Requisito faltante No.:	13
Severidad: 4	Errores menores de uso de la aplicación, presentación inconsistente de interfaz gráfica, etc.
Descripción:	Error al editar un control ya creado. No aparece la información del control seleccionado.

ISS  Bienvenido: admin - Proyecto SGSI - Empresa Tecnológica S.A.

Norma Control Plan de Acción Comentario Usuario Observación  1%

ISO 27001:2005 - Plan de acción General # 1 

Editar Control

Datos del control:

Nombre: xxxxx

Aplicable:

Implementada:

Nivel: 1

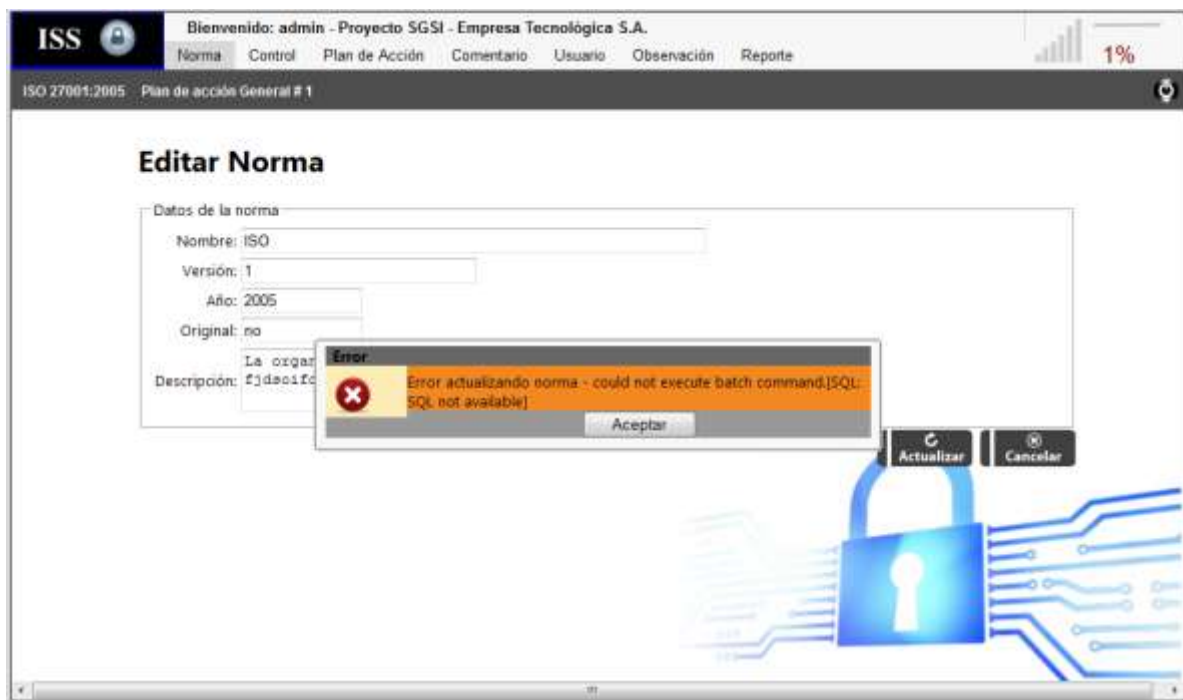
Descripción: xxxxx

Norma: ISO 27001

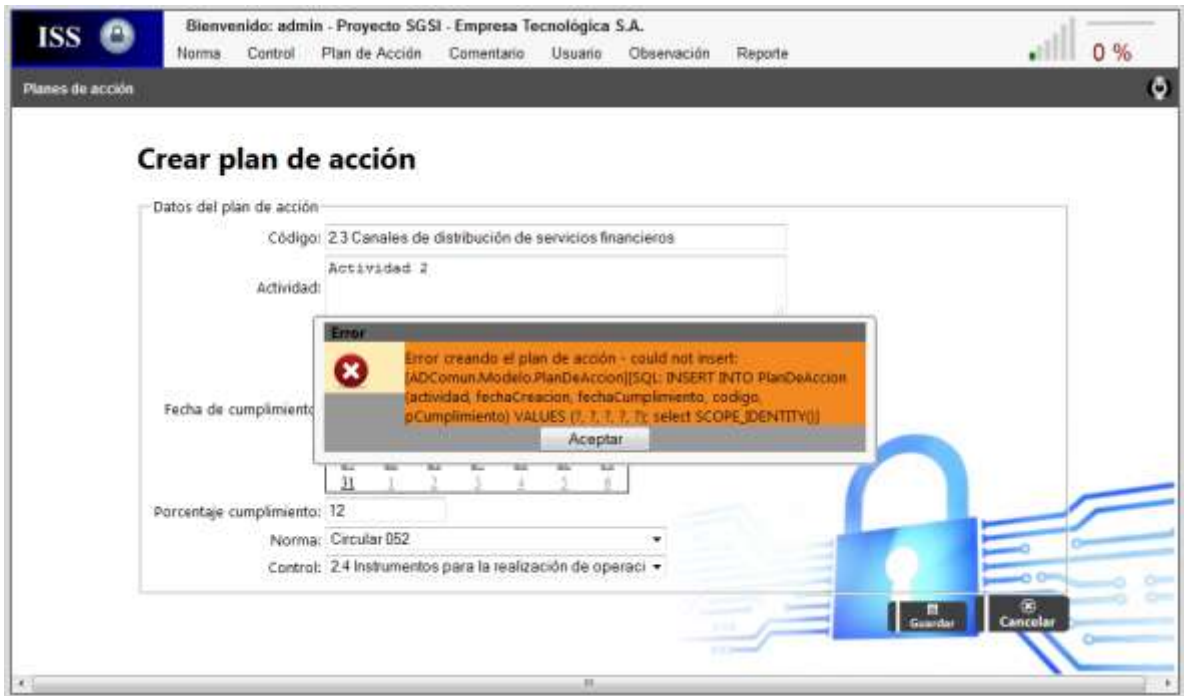
Requisito faltante No.:	14
Severidad: 4	Errores menores de uso de la aplicación, presentación inconsistente de interfaz gráfica, etc.
Descripción:	Error al crear un plan de acción cuando la norma seleccionada no tiene controles creados asociados a la norma. (Es necesario validar que las normas tengan siempre como mínimo un control asociado.)



Requisito faltante No.:	15
Severidad: 4	Errores menores de uso de la aplicación, presentación inconsistente de interfaz gráfica, etc.
Descripción:	Error al actualizar una norma con una descripción mayor a 1 renglón




Requisito faltante No.:	16
Severidad: 4	Errores menores de uso de la aplicación, presentación inconsistente de interfaz gráfica, etc.
Descripción:	Error al agregar un código de varios caracteres no es permitido en el plan de acción, es necesario modificar el tipo de valor del campo código.





Requisito faltante No.:	17
Severidad: 4	Errores menores de uso de la aplicación, presentación inconsistente de interfaz gráfica, etc.
Descripción:	Error al agregar un plan de acción sin seleccionar la fecha de cumplimiento, es necesario cambiar el error mostrado o asignar una fecha por defecto si no se selecciona esta.






Requisito faltante No.:	18
Severidad: 4	Errores menores de uso de la aplicación, presentación inconsistente de interfaz gráfica, etc.
Descripción:	Error al eliminar un plan de acción, no es posible eliminarlo.

ISS  Bienvenido: admin - Proyecto SGSI - ALFCOM S.A.
















Norma Control Plan de Acción Comentario Usuario Observación Reporte  28 %

Planes de acción 

Gestión de planes de acción

   **Crear**

Planes de acción

Código	Actividad	Fecha de creación	Fecha de cumplimiento	Opciones
2.1 Criterios de seguridad información		Error Error eliminando el PlanDeAccion - could not delete collection: [ADComun.Modelo.PlanDeAccion.Observacion#10002][SQL: UPDATE Observacion SET planDeAccionId = null, idObservacion = null WHERE planDeAccionId = @p0]		00:00  
2.3 Canales de distribución financiero				00:00  
1. Ámbito de aplicación				00:00   <input type="button" value="Aceptar"/>
2.2 Criterios de calidad de la información	Actividad 1	07/08/2014 12:00:00 a.m.	29/08/2014 12:00:00 a.m.	 
2.4 Instrumentos para la realización de operación	Actividad 1	07/08/2014 12:00:00 a.m.	16/08/2014 12:00:00 a.m.	 
1.2 Aplicación	Actividad 1	07/08/2014 12:00:00 a.m.	21/08/2014 12:00:00 a.m.	 
1. Aplicación 27002	Actividad 1	09/08/2014 12:00:00 a.m.	20/08/2014 12:00:00 a.m.	 

Anexo 2. Base de datos inicial de la aplicación.

```
CREATE TABLE [dbo].[Comentario](
    [comentarioId] [bigint] IDENTITY(1,1) NOT NULL,
    [texto] [text] NULL,
    [controlId] [bigint] NOT NULL,
    [usuarioId] [bigint] NOT NULL,
    CONSTRAINT [PK_ComentarioId] PRIMARY KEY CLUSTERED
(
    [comentarioId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
/***** Object: Table [dbo].[Control]      Script Date: 05/08/2014 22:06:36
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Control](
    [controlId] [bigint] IDENTITY(1,1) NOT NULL,
    [aplicable] [bit] NULL,
    [implementada] [bit] NULL,
    [nombre] [varchar](50) NULL,
    [nivel] [int] NULL,
    [descripcion] [text] NULL,
    [normaId] [bigint] NOT NULL,
    [PCumplimiento] [int] NULL,
    CONSTRAINT [PK_ControlId] PRIMARY KEY CLUSTERED
(
    [controlId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[ControlPlanDeAccion]      Script Date: 05/08/2014
22:06:36 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ControlPlanDeAccion](
    [controlId] [bigint] NOT NULL,
    [planDeAccionId] [bigint] NOT NULL,
    CONSTRAINT [PK ControlPlanDeAccionId] PRIMARY KEY CLUSTERED
```

```

(
    [planDeAccionId] ASC,
    [controlId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object: Table [dbo].[ControlRelacion]    Script Date: 05/08/2014
22:06:36 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ControlRelacion](
    [controlId] [bigint] NOT NULL,
    [padreId] [bigint] NOT NULL,
    CONSTRAINT [PK_controlId_padreId] PRIMARY KEY CLUSTERED
(
    [controlId] ASC,
    [padreId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object: Table [dbo].[Norma]    Script Date: 05/08/2014 22:06:36
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Norma](
    [normaId] [bigint] IDENTITY(1,1) NOT NULL,
    [nombre] [varchar](50) NULL,
    [version] [varchar](50) NULL,
    [ano] [varchar](50) NULL,
    [original] [varchar](50) NULL,
    [descripcion] [text] NULL,
    [pCumplimiento] [int] NULL,
    CONSTRAINT [PK_NormaId] PRIMARY KEY CLUSTERED
(
    [normaId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Observacion]    Script Date: 05/08/2014
22:06:36 *****/
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Observacion](
    [observacionId] [bigint] IDENTITY(1,1) NOT NULL,
    [texto] [text] NULL,
    [planDeAccionId] [bigint] NULL,
    CONSTRAINT [PK_ObservacionId] PRIMARY KEY CLUSTERED
(
    [observacionId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
/***** Object: Table [dbo].[PlanDeAccion]    Script Date: 05/08/2014
22:06:36 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[PlanDeAccion](
    [planDeAccionId] [bigint] IDENTITY(1,1) NOT NULL,
    [actividad] [varchar](50) NULL,
    [fechaCreacion] [datetime] NOT NULL,
    [fechaCumplimiento] [datetime] NOT NULL,
    [codigo] [varchar](50) NULL,
    [PCumplimiento] [nvarchar](50) NULL,
    CONSTRAINT [PK_PlanDeAccionId] PRIMARY KEY CLUSTERED
(
    [planDeAccionId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Rol]    Script Date: 05/08/2014 22:06:36
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Rol](
    [rolId] [bigint] IDENTITY(1,1) NOT NULL,
    [nombre] [varchar](50) NULL,
    [Descripcion] [varchar](50) NULL,
    CONSTRAINT [PK_RolId] PRIMARY KEY CLUSTERED
(
    [rolId] ASC

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Usuario]    Script Date: 05/08/2014 22:06:36
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Usuario](
    [usuarioId] [bigint] IDENTITY(1,1) NOT NULL,
    [nombre] [varchar](255) NULL,
    [nombreIngreso] [varchar](50) NULL,
    [departamento] [varchar](50) NULL,
    [clave] [varchar](100) NULL,
    [activo] [bit] NULL,
    CONSTRAINT [PK_UsuarioId] PRIMARY KEY CLUSTERED
(
    [usuarioId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [UK_NOMBRE_USUARIO] UNIQUE NONCLUSTERED
(
    [nombre] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [UK_NOMBREINGRESO_USUARIO] UNIQUE NONCLUSTERED
(
    [nombreIngreso] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[UsuarioRol]    Script Date: 05/08/2014 22:06:36
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[UsuarioRol](
    [rolId] [bigint] NOT NULL,
    [usuarioId] [bigint] NOT NULL,
    CONSTRAINT [PK_UsuarioRolId] PRIMARY KEY CLUSTERED
(
    [usuarioId] ASC,
    [rolId] ASC

```

```

) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
ALTER TABLE [dbo].[Comentario] WITH CHECK ADD CONSTRAINT
[FK_Comentario_Control] FOREIGN KEY([controlId])
REFERENCES [dbo].[Control] ([controlId])
GO
ALTER TABLE [dbo].[Comentario] CHECK CONSTRAINT [FK_Comentario_Control]
GO
ALTER TABLE [dbo].[Comentario] WITH CHECK ADD CONSTRAINT
[FK_Comentario_Usuario] FOREIGN KEY([usuarioId])
REFERENCES [dbo].[Usuario] ([usuarioId])
GO
ALTER TABLE [dbo].[Comentario] CHECK CONSTRAINT [FK_Comentario_Usuario]
GO
ALTER TABLE [dbo].[Control] WITH CHECK ADD CONSTRAINT
[FK_Control_ControlControl] FOREIGN KEY([controlId])
REFERENCES [dbo].[Control] ([controlId])
GO
ALTER TABLE [dbo].[Control] CHECK CONSTRAINT [FK_Control_ControlControl]
GO
ALTER TABLE [dbo].[Control] WITH CHECK ADD CONSTRAINT [FK_Control_Norma]
FOREIGN KEY([normaId])
REFERENCES [dbo].[Norma] ([normaId])
GO
ALTER TABLE [dbo].[Control] CHECK CONSTRAINT [FK_Control_Norma]
GO
ALTER TABLE [dbo].[ControlPlanDeAccion] WITH CHECK ADD CONSTRAINT
[FK_ControlPlanDeAccion_Cotrol] FOREIGN KEY([controlId])
REFERENCES [dbo].[Control] ([controlId])
GO
ALTER TABLE [dbo].[ControlPlanDeAccion] CHECK CONSTRAINT
[FK_ControlPlanDeAccion_Cotrol]
GO
ALTER TABLE [dbo].[ControlPlanDeAccion] WITH CHECK ADD CONSTRAINT
[FK_ControlPlanDeAccion_PlanDeAccion] FOREIGN KEY([planDeAccionId])
REFERENCES [dbo].[PlanDeAccion] ([planDeAccionId])
GO
ALTER TABLE [dbo].[ControlPlanDeAccion] CHECK CONSTRAINT
[FK_ControlPlanDeAccion_PlanDeAccion]
GO
ALTER TABLE [dbo].[Observacion] WITH CHECK ADD CONSTRAINT
[FK_Observacion_PlanDeAccion] FOREIGN KEY([planDeAccionId])
REFERENCES [dbo].[PlanDeAccion] ([planDeAccionId])
GO
ALTER TABLE [dbo].[Observacion] CHECK CONSTRAINT
[FK_Observacion_PlanDeAccion]
GO
USE [master]
GO
ALTER DATABASE [Gestion] SET READ_WRITE
GO

```

Anexo 3. Código fuente de la aplicación.

```
Comentario.cs
using ADComun.Modelo;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using VSComun.Modelo;
using NHMA = NHibernate.Mapping.Attributes;

namespace ADComun.Modelo
{
    /// <summary>
    /// Clase Comentario
    /// </summary>
    [NHMA.Class(Table = "Comentario", Lazy = false, NameType = typeof(Comentario))]
    public class Comentario : IObjetoPersistente
    {
        #region Atributos

        /// <summary>
        /// Identificador de la clase Comentario
        /// </summary>
        private long m_Id;

        /// <summary>
        /// Campo Texto del comentario
        /// </summary>
        private string m_Texto;

        /// <summary>
        /// Relación con la clase Control
        /// </summary>
        private Control m_Control;

        /// <summary>
        /// Relación con la clase Usuario
        /// </summary>
        private Usuario m_Usuario;

        #endregion

        #region Constructor

        /// <summary>
        /// Constructor de la clase comentario.
        /// </summary>
        public Comentario()
        {
        }

    }
}
```



```
#endregion

#region Propiedades

/// <summary>
/// Obtiene o establece el identificador de la clase Comentario.
/// </summary>
[NHMA.Id(-1, Column = "comentarioid", TypeType = typeof(long), Name = "Id")]
[NHMA.Generator(Class = "identity")]
public long Id
{
    get
    {
        return m_Id;
    }
    set
    {
        m_Id = value;
    }
}

/// <summary>
/// Obtiene o establece el texto de la clase Comentario.
/// </summary>
[NHMA.Property(-1, TypeType = typeof(string), NotNull = true)]
[NHMA.Column(Name = "texto", SqlType = "varchar(50)")]
public string Texto
{
    get
    {
        return m_Texto;
    }
    set
    {
        m_Texto = value;
    }
}

/// <summary>
/// Obtiene o establece la relación de muchos a uno entre Control y la clase Comentario.
/// </summary>
[NHMA.ManyToOne(Column = "controlId", NotNull = true)]
public Control Control
{
    get
    {
        return m_Control;
    }
    set
    {
        m_Control = value;
    }
}
}
```

```

    /// <summary>
    /// Obtiene o establece la relación de muchos a uno entre usuario y la clase Comentario.
    /// </summary>
    [NHMA.ManyToOne(Column = "usuariold", NotNull = true)]
    public Usuario Usuario
    {
        get
        {
            return m_Usuario;
        }
        set
        {
            m_Usuario = value;
        }
    }
    #endregion
}
}

```

```

Control.cs
using ADComun.Modelo;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using VSComun.Modelo;
using NHMA = NHibernate.Mapping.Attributes;

namespace ADComun.Modelo
{
    /// <summary>
    /// Clase Control.
    /// </summary>
    [NHMA.Class(Table = "Control", Lazy = false, NameType = typeof(Control))]
    public class Control : IObjetoPersistente
    {
        #region Atributos

        /// <summary>
        /// Identificador de la clase control.
        /// </summary>
        private long m_Id;

        /// <summary>
        /// Atributo para determinar si es aplicable o no el control
        /// </summary>
        private bool m_Aplicable;

        /// <summary>
        /// Atributo para determinar si está o no implementado el control.

```

```

    /// </summary>
    private bool m_Implementada;

    /// <summary>
    /// Nombre del control.
    /// </summary>
    private string m_Nombre;

    /// <summary>
    /// Nivel del control.
    /// </summary>
    private int m_Nivel;

    /// <summary>
    /// porcentaje de cumplimiento del control.
    /// </summary>
    private int m_PCumplimiento;

    /// <summary>
    /// Descripción del control.
    /// </summary>
    private string m_Descripcion;

    /// <summary>
    /// relación con el objeto Norma.
    /// </summary>
    private Norma m_Norma;

    /// <summary>
    /// Colección de tipo PlanDeAccion
    /// </summary>
    private ICollection<PlanDeAccion> m_PlanDeAccion;

    /// <summary>
    /// Colección de tipo Control.
    /// </summary>
    private ICollection<Control> m_Controles;

    /// <summary>
    /// Colección de tipo rol.
    /// </summary>
    private ICollection<Control> m_Padres;

#endregion

#region Constructor

    /// <summary>
    /// Constructor de la clase Control.
    /// </summary>
    public Control()
    {
        m_Controles = new HashSet<Control>();
        m_Padres = new HashSet<Control>();
    }

```

```
}  
  
#endregion  
  
#region Propiedades  
  
[NHMA.Id(-1, Column = "controlId", TypeType = typeof(long), Name = "Id")]  
[NHMA.Generator(Class = "identity")]  
/// <summary>  
/// Obtiene o establece el identificador del control.  
/// </summary>  
public long Id  
{  
    get  
    {  
        return m_Id;  
    }  
    set  
    {  
        m_Id = value;  
    }  
}  
  
/// <summary>  
/// Obtiene o establece si es aplicable el control.  
/// </summary>  
[NHMA.Property(Column = "aplicable", TypeType = typeof(bool), NotNull = false)]  
public bool Aplicable  
{  
    get  
    {  
        return m_Aplicable;  
    }  
    set  
    {  
        m_Aplicable = value;  
    }  
}  
  
/// <summary>  
/// Obtiene o establece si es implementado el control.  
/// </summary>  
[NHMA.Property(Column = "implementada", TypeType = typeof(bool), NotNull = false)]  
public bool Implementada  
{  
    get  
    {  
        return m_Implementada;  
    }  
    set  
    {  
        m_Implementada = value;  
    }  
}  
}
```

```

    /// <summary>
    /// Obtiene o establece el nombre del control.
    /// </summary>
    [NHMA.Property(-1, TypeType = typeof(string), NotNull = true)]
    [NHMA.Column(Name = "nombre", SqlType = "varchar(50)")]
    public string Nombre
    {
        get
        {
            return m_Nombre;
        }
        set
        {
            m_Nombre = value;
        }
    }

```

```

    /// <summary>
    /// Obtiene o establece el nivel del control.
    /// </summary>
    [NHMA.Property(Column = "nivel", TypeType = typeof(int), NotNull = false)]
    public int Nivel
    {
        get
        {
            return m_Nivel;
        }
        set
        {
            m_Nivel = value;
        }
    }

```

```

    /// <summary>
    /// Obtiene o establece el % de cumplimiento del control.
    /// </summary>
    [NHMA.Property(Column = "pCumplimiento", TypeType = typeof(int), NotNull = false)]
    public int PCumplimiento
    {
        get
        {
            return m_PCumplimiento;
        }
        set
        {
            m_PCumplimiento = value;
        }
    }

```

```

    /// <summary>
    /// Obtiene o establece la descripción del control.
    /// </summary>

```

```

[NHMA.Property(-1, TypeType = typeof(string), NotNull = true)]
[NHMA.Column(Name = "descripcion", SqlType = "varchar(255)")]
public string Descripcion
{
    get
    {
        return m_Descripcion;
    }
    set
    {
        m_Descripcion = value;
    }
}

```

```

/// <summary>
/// Obtiene o establece la relación de muchos a uno entre la norma y el control.
/// </summary>
[NHMA.ManyToOne(Column = "normald", NotNull = true)]
public Norma Norma
{
    get
    {
        return m_Norma;
    }
    set
    {
        m_Norma = value;
    }
}

```

```

/// <summary>
/// Obtiene o establece la relación de muchos a muchos entre el PlanDeAccion y el control.
/// </summary>
[NHMA.Set(-2, Table = "ControlPlanDeAccion", Lazy = NHMA.CollectionLazy.False)]
[NHMA.Key(-1, Column = "controlId")]
[NHMA.ManyToMany(ClassType = typeof(PlanDeAccion), Column = "planDeAccionId")]
public ICollection<PlanDeAccion> PlanDeAccion
{
    get
    {
        return m_PlanDeAccion;
    }
    set
    {
        m_PlanDeAccion = value;
    }
}

```

```

/// <summary>
/// Obtiene o establece la relación del rol con el usuario.
/// Relación de muchos a muchos.
/// </summary>
[NHMA.Set(-2, Table = "ControlRelacion", Lazy = NHMA.CollectionLazy.False)]
[NHMA.Key(-1, Column = "padreId")]

```

```

[NHMA.ManyToMany(ClassType = typeof(Control), Column = "controlId")]
public ICollection<Control> Controles
{
    get
    {
        return m_Controles;
    }
    set
    {
        m_Controles = value;
    }
}

/// <summary>
/// Obtiene o establece la collection del roles que se relacionan con el usuario.
/// Relación de muchos a muchos.
/// </summary>
[NHMA.Set(-2, Table = "ControlRelacion", Lazy = NHMA.CollectionLazy.False)]
[NHMA.Key(-1, Column = "controlId")]
[NHMA.ManyToMany(ClassType = typeof(Control), Column = "padreId")]
public ICollection<Control> Padres
{
    get
    {
        return m_Padres;
    }
    set
    {
        m_Padres = value;
    }
}

#endregion
}
}

```

```

Norma.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using VSComun.Modelo;
using NHMA = NHibernate.Mapping.Attributes;

namespace ADComun.Modelo
{
    /// <summary>
    /// Clase Norma.
    /// </summary>
    [NHMA.Class(Table = "Norma", Lazy = false, NameType = typeof(Norma))]

```

```
public class Norma : IObjetoPersistente
{
    #region Atributos

    /// <summary>
    /// Identificador de la norma.
    /// </summary>
    private long m_Id;

    /// <summary>
    /// Nombre de la norma.
    /// </summary>
    private string m_Nombre;

    /// <summary>
    /// Versión de la norma.
    /// </summary>
    private string m_Version;

    /// <summary>
    /// Año de la norma.
    /// </summary>
    private string m_Ano;

    /// <summary>
    /// Se cuenta con la norma en original o con licencia.
    /// </summary>
    private string m_Original;

    /// <summary>
    /// porcentaje de cumplimiento de la norma.
    /// </summary>
    private int m_PCumplimiento;

    /// <summary>
    /// Descripción de la norma.
    /// </summary>
    private string m_Descripcion;

    #endregion

    #region Constructor

    /// <summary>
    /// Constructor de la clase norma.
    /// </summary>
    public Norma()
    {
    }

    #endregion

    #region Propiedades
```



```

    /// <summary>
    /// Obtiene o establece el identificador de la norma.
    /// </summary>
    [NHMA.Id(-1, Column = "normald", TypeType = typeof(long), Name = "Id")]
    [NHMA.Generator(Class = "identity")]
    public long Id
    {
        get
        {
            return m_Id;
        }
        set
        {
            m_Id = value;
        }
    }

    /// <summary>
    /// Obtiene o establece el nombre de la norma.
    /// </summary>
    [NHMA.Property(-1, TypeType = typeof(string), NotNull = true)]
    [NHMA.Column(Name = "nombre", SqlType = "varchar(50)")]
    public string Nombre
    {
        get
        {
            return m_Nombre;
        }
        set
        {
            m_Nombre = value;
        }
    }

    /// <summary>
    /// Obtiene o establece el % de cumplimiento de la norma.
    /// </summary>
    [NHMA.Property(Column = "pCumplimiento", TypeType = typeof(int), NotNull = false)]
    public int PCumplimiento
    {
        get
        {
            return m_PCumplimiento;
        }
        set
        {
            m_PCumplimiento = value;
        }
    }

    /// <summary>
    /// Obtiene o establece la versión de la norma.
    /// </summary>

```

```

[NHMA.Property(-1, TypeType = typeof(string), NotNull = true)]
[NHMA.Column(Name = "version", SqlType = "varchar(50)")]
public string Version
{
    get
    {
        return m_Version;
    }
    set
    {
        m_Version = value;
    }
}

```

```

/// <summary>
/// Obtiene o establece el año de la norma.
/// </summary>
[NHMA.Property(-1, TypeType = typeof(string), NotNull = true)]
[NHMA.Column(Name = "ano", SqlType = "varchar(50)")]
public string Ano
{
    get
    {
        return m_Ano;
    }
    set
    {
        m_Ano = value;
    }
}

```

```

/// <summary>
/// Obtiene o establece el estado de legalidad de la norma.
/// </summary>
[NHMA.Property(-1, TypeType = typeof(string), NotNull = true)]
[NHMA.Column(Name = "original", SqlType = "varchar(50)")]
public string Original
{
    get
    {
        return m_Original;
    }
    set
    {
        m_Original = value;
    }
}

```

```

/// <summary>
/// Obtiene o establece la descripción de la norma.
/// </summary>
[NHMA.Property(-1, TypeType = typeof(string), NotNull = true)]
[NHMA.Column(Name = "descripcion", SqlType = "varchar(50)")]
public string Descripcion

```

```
    {
        get
        {
            return m_Descripcion;
        }
        set
        {
            m_Descripcion = value;
        }
    }
}
#endregion
}
```

```
Observacion.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using VSComun.Modelo;
using NHMA = NHibernate.Mapping.Attributes;

namespace ADComun.Modelo
{
    /// <summary>
    /// Clase Observación.
    /// </summary>
    [NHMA.Class(Table = "Observacion", Lazy = false, NameType = typeof(Observacion))]
    public class Observacion : IObjetoPersistente
    {
        #region Atributos

        /// <summary>
        /// Identificador de la Observación.
        /// </summary>
        private long m_Id;

        /// <summary>
        /// Texto de la observación.
        /// </summary>
        private string m_Texto;

        /// <summary>
        /// Relación con la clase PlanDeAcción
        /// </summary>
        private PlanDeAccion m_PlanDeAccion;

        #endregion

        #region Constructor
```

```
/// <summary>
/// Constructor de la clase observación.
/// </summary>
public Observacion()
{
}

#endregion

#region Propiedades

/// <summary>
/// Obtiene o establece el identificador de la observación.
/// </summary>
[NHMA.Id(-1, Column = "observacionId", TypeType = typeof(long), Name = "Id")]
[NHMA.Generator(Class = "identity")]
public long Id
{
    get
    {
        return m_Id;
    }
    set
    {
        m_Id = value;
    }
}

/// <summary>
/// Obtiene o establece el texto de la Observación.
/// </summary>
[NHMA.Property(-1, TypeType = typeof(string), NotNull = true)]
[NHMA.Column(Name = "texto", SqlType = "varchar(50)")]
public string Texto
{
    get
    {
        return m_Texto;
    }
    set
    {
        m_Texto = value;
    }
}

/// <summary>
/// Obtiene o establece la relación de muchos a uno entre el planDeAccion y la norma.
/// </summary>
[NHMA.ManyToOne(Column = "planDeAccionId", NotNull = false)]
public PlanDeAccion PlanDeAccion
{
    get

```

```
        return m_PlanDeAccion;
    }
    set
    {
        m_PlanDeAccion = value;
    }
}
#endregion
}
```

```
PlanDeAccion.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using VSComun.Modelo;
using NHMA = NHibernate.Mapping.Attributes;

namespace ADComun.Modelo
{
    /// <summary>
    /// clase del plan de acción.
    /// </summary>
    [NHMA.Class(Table = "PlanDeAccion", Lazy = false, NameType = typeof(PlanDeAccion))]
    public class PlanDeAccion : IObjetoPersistente
    {
        #region Atributos

        /// <summary>
        /// Identificador del plan de acción.
        /// </summary>
        private long m_Id;

        /// <summary>
        /// Actividad del plan de acción.
        /// </summary>
        private string m_Actividad;

        /// <summary>
        /// Fecha de creación del plan de acción.
        /// </summary>
        private DateTime m_FechaCreacion;

        /// <summary>
        /// Fecha de cumplimiento del plan de acción.
        /// </summary>
        private DateTime m_FechaCumplimiento;
    }
}
```

```
/// <summary>
/// Relación entre control y el plan de acción.
/// </summary>
private ICollection<Control> m_Control;

/// <summary>
/// Relación entre observación y el plan de acción.
/// </summary>
private IList<Observacion> m_Observacion;

/// <summary>
/// Código del plan de acción.
/// </summary>
private string m_Codigo;

/// <summary>
///
/// </summary>
private string m_PCumplimiento;

#endregion

#region Constructor

/// <summary>
/// Constructor de la clase plan de acción.
/// </summary>
public PlanDeAccion()
{
}

#endregion

#region Propiedades

/// <summary>
/// Obtiene o establece el identificador del plan de acción.
/// </summary>
[NHMA.Id(-1, Column = "planDeAccionId", TypeType = typeof(long), Name = "Id")]
[NHMA.Generator(Class = "identity")]
public long Id
{
    get
    {
        return m_Id;
    }
    set
    {
        m_Id = value;
    }
}

/// <summary>
```

```

/// Obtiene o establece la actividad del plan de acción.
/// </summary>
[NHMA.Property(-1, TypeType = typeof(string), NotNull = true)]
[NHMA.Column(Name = "actividad", SqlType = "varchar(50)")]
public string Actividad
{
    get
    {
        return m_Actividad;
    }
    set
    {
        m_Actividad = value;
    }
}

```

```

/// <summary>
/// Obtiene o establece la fecha de creación del plan de acción.
/// </summary>
[NHMA.Property(Column = "fechaCreacion", TypeType = typeof(DateTime), NotNull = true)]
public DateTime FechaCreacion
{
    get
    {
        return m_FechaCreacion;
    }
    set
    {
        m_FechaCreacion = value;
    }
}

```

```

/// <summary>
/// Obtiene o establece la fecha de cumplimiento del plan de acción.
/// </summary>
[NHMA.Property(Column = "fechaCumplimiento", TypeType = typeof(DateTime), NotNull = true)]
public DateTime FechaCumplimiento
{
    get
    {
        return m_FechaCumplimiento;
    }
    set
    {
        m_FechaCumplimiento = value;
    }
}

```

```

/// <summary>
/// Obtiene o establece la relación muchos a muchos entre el control y el plan de acción.
/// </summary>
[NHMA.Set(-2, Table = "ControlPlanDeAccion", Lazy = NHMA.CollectionLazy.False)]
[NHMA.Key(-1, Column = "PlanDeAccionId")]
[NHMA.ManyToMany(ClassType = typeof(Control), Column = "controlId")]

```

```

public ICollection<Control> Control
{
    get
    {
        return m_Control;
    }
    set
    {
        m_Control = value;
    }
}

/// <summary>
/// Obtiene o establece la relación de uno a muchos entre la observación y el plan de acción.
/// </summary>
[NHMA.List(-3, Cascade = "save-update")]
[NHMA.Key(-2, ForeignKey = "FK_PlanDeAccion", Column = "planDeAccionId")]
[NHMA.Index(-1, Column = "idObservacion")]
[NHMA.OneToOne(ClassType = typeof(Observacion))]
public IList<Observacion> Observacion
{
    get
    {
        return m_Observacion;
    }
    set
    {
        m_Observacion = value;
    }
}

/// <summary>
/// Obtiene o establece el código del plan de acción.
/// </summary>
[NHMA.Property(-1, TypeType = typeof(string), NotNull = true)]
[NHMA.Column(Name = "codigo", SqlType = "varchar(50)")]
public string Codigo
{
    get
    {
        return m_Codigo;
    }
    set
    {
        m_Codigo = value;
    }
}

/// <summary>
/// Obtiene o establece el porcensaje de cumplimiento del plan de acción.
/// </summary>
[NHMA.Property(-1, TypeType = typeof(string), NotNull = true)]
[NHMA.Column(Name = "pCumplimiento", SqlType = "varchar(50)")]
public string PCumplimiento

```



```
    {
        get
        {
            return m_PCumplimiento;
        }
        set
        {
            m_PCumplimiento = value;
        }
    }

    #endregion
}
```

```
Rol.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using NHMA = NHibernate.Mapping.Attributes;
using VSComun.Modelo;

namespace ADComun.Modelo
{
    /// <summary>
    /// Clase Rol - tabla rol
    /// </summary>
    [NHMA.Class(Table = "Rol", Lazy = false, NameType = typeof(Rol))]
    public class Rol : IObjetoPersistente
    {
        #region Atributos

        /// <summary>
        /// Identificador del Rol.
        /// </summary>
        private long m_Id;

        /// <summary>
        /// Nombre del rol.
        /// </summary>
        private string m_Nombre;

        /// <summary>
        /// Descripción del rol.
        /// </summary>
        private string m_Descripcion;

        /// <summary>
        /// Colección de tipo usuario.

```

```

    /// </summary>
    private ICollection<Usuario> m_Usuarios;

#endregion

#region Constructores

    /// <summary>
    /// Constructor de la clase rol.
    /// </summary>
    public Rol()
    {
        m_Usuarios = new HashSet<Usuario>();
    }

#endregion

#region Propiedades

    /// <summary>
    /// Obtiene o establece el identificador del rol.
    /// </summary>
    [NHMA.Id(-1, Column = "rolId", TypeType = typeof(long), Name = "Id")]
    [NHMA.Generator(Class = "identity")]
    public long Id
    {
        get
        {
            return m_Id;
        }
        set
        {
            m_Id = value;
        }
    }

    /// <summary>
    /// Obtiene o establece el nombre del rol.
    /// </summary>
    [NHMA.Property(-1, TypeType = typeof(string), NotNull = true)]
    [NHMA.Column(Name = "nombre", SqlType = "varchar(100)")]
    public string Nombre
    {
        get
        {
            return m_Nombre;
        }
        set
        {
            m_Nombre = value;
        }
    }
}

```

```
/// <summary>
/// Obtiene o establece la descripción del rol.
/// </summary>
[NHMA.Property(-1, TypeType = typeof(string), NotNull = false)]
[NHMA.Column(Name = "descripcion", SqlType = "varchar(255)")]
public string Descripcion
{
    get
    {
        return m_Descripcion;
    }
    set
    {
        m_Descripcion = value;
    }
}

/// <summary>
/// Obtiene o establece la relación del rol con el usuario.
/// Relación de muchos a muchos.
/// </summary>
[NHMA.Set(-2, Table = "UsuarioRol", Lazy = NHMA.CollectionLazy.False)]
[NHMA.Key(-1, Column = "roleId")]
[NHMA.ManyToMany(ClassType = typeof(Usuario), Column = "usuariold")]
public ICollection<Usuario> Usuarios
{
    get
    {
        return m_Usuarios;
    }
    set
    {
        m_Usuarios = value;
    }
}
#endregion
}
```

```
Usuario.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using NHMA = NHibernate.Mapping.Attributes;
using VSComun.Modelo;

namespace ADComun.Modelo
{
    /// <summary>
```

```

/// Clase Usuario - tabla usuario
/// </summary>
[NHMA.Class(Table = "Usuario", Lazy = false, NameType = typeof(Usuario))]
public class Usuario : IObjetoPersistente
{
    #region Atributos

    /// <summary>
    /// Identificador del usuario
    /// </summary>
    private long m_Id;

    /// <summary>
    /// Nombre del usuario.
    /// </summary>
    private string m_Nombre;

    /// <summary>
    /// Nombre de ingreso del usuario.
    /// </summary>
    private string m_NombreIngreso;

    /// <summary>
    /// Colección de tipo rol.
    /// </summary>
    private ICollection<Rol> m_Roles;

    /// <summary>
    /// Clave del usuario.
    /// </summary>
    private string m_Clave;

    /// <summary>
    ///
    /// </summary>
    private string m_Tipo;

    /// <summary>
    ///
    /// </summary>
    private bool m_Activo;

    #endregion

    #region Constructor

    public Usuario()
    {
        m_Roles = new HashSet<Rol>();
        // m_Casos = new HashSet<Caso>();
    }

    #endregion
}

```

```
#region Propiedades

/// <summary>
/// Obtiene o establece el identificador del usuario.
/// </summary>
[NHMA.Id(-1, Column = "usuariId", TypeType = typeof(long), Name = "Id")]
[NHMA.Generator(Class = "identity")]
public long Id
{
    get
    {
        return m_Id;
    }
    set
    {
        m_Id = value;
    }
}

/// <summary>
/// Obtiene o establece el nombre del usuario.
/// </summary>
[NHMA.Property(-1, TypeType = typeof(string), NotNull = true)]
[NHMA.Column(Name = "nombre", SqlType = "varchar(255)")]
public string Nombre
{
    get
    {
        return m_Nombre;
    }
    set
    {
        m_Nombre = value;
    }
}

/// <summary>
/// Obtiene o establece el nombre de ingreso del usuario.
/// </summary>
[NHMA.Property(-1, TypeType = typeof(string), NotNull = true)]
[NHMA.Column(Name = "nombreIngreso", SqlType = "varchar(100)", UniqueKey =
"UK_NOMBREINGRESO_USUARIO")]
public string NombreIngreso
{
    get
    {
        return m_NombreIngreso;
    }
    set
    {
        m_NombreIngreso = value;
    }
}

}
```

```

/// <summary>
/// Obtiene o establece la clave del usuario.
/// </summary>
[NHMA.Property(-1, TypeType = typeof(string), NotNull = false)]
[NHMA.Column(Name = "clave", SqlType = "varchar(255)")]
public string Clave
{
    get
    {
        return m_Clave;
    }
    set
    {
        m_Clave = value;
    }
}

```

```

/// <summary>
/// Obtiene o establece el tipo del usuario.
/// </summary>
public string Tipo
{
    get
    {
        return m_Tipo;
    }
    set
    {
        m_Tipo = value;
    }
}

```

```

/// <summary>
/// Obtiene o establece la collection del roles que se relacionan con el usuario.
/// Relación de muchos a muchos.
/// </summary>
[NHMA.Set(-2, Table = "UsuarioRol", Lazy = NHMA.CollectionLazy.False)]
[NHMA.Key(-1, Column = "usuariold")]
[NHMA.ManyToMany(ClassType = typeof(Rol), Column = "rolld")]
public ICollection<Rol> Roles
{
    get
    {
        return m_Roles;
    }
    set
    {
        m_Roles = value;
    }
}

```

```

/// <summary>
/// Obtiene o establece el valor de activo del usuario.
/// </summary>

```

```
[NHMA.Property(Column = "activo", TypeType = typeof(bool), NotNull = true)]
public bool Activo
{
    get
    {
        return m_Activo;
    }
    set
    {
        m_Activo = value;
    }
}
#endregion
}
```

```
UsuarioConNombreIngresoRepetidoExcepcion.cs
using ADComun.Enums;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using VSComun.Excepciones;

namespace ADComun.Excepciones
{
    /// <summary>
    /// Excepción cuando se intenta ingresar un usuario con un nombre repetido.
    /// </summary>
    public class UsuarioConNombreIngresoRepetidoExcepcion : AplicacionExcepcion
    {
        /// <summary>
        /// Implementación de la clase.
        /// </summary>
        /// <param name="_Nombre"></param>
        public UsuarioConNombreIngresoRepetidoExcepcion(string _Nombre)
            : base(CodigosError.UsuarioConNombreIngresoRepetido, "Ya existe un usuario con el nombre
de ingreso " + _Nombre)
        {
        }
    }
}
```

```
Constantes.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```

namespace ADComun.Constantes
{
    /// <summary>
    /// Clase que contiene las constantes globales de la aplicación.
    /// </summary>
    public class Constantes
    {

        /// <summary>
        /// Identificador de la aplicación.
        /// </summary>
        public const string APLICACION = "AD";

    }
}

```

ManejadorComentario.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using VSComun.ManejadoresDatos;
using ADComun.Modelo;
using ADComun.Constantes;

namespace ADNucleo.Manejadores
{
    class ManejadorComentario
    {
        #region Atributos

        /// <summary>
        /// Manejador de datos de tipo comentario.
        /// </summary>
        private ManejadorDatos<Comentario> m_ManejadorDatosComentario;

        #endregion

        #region Constructores

        /// <summary>
        /// Constructor del manejador del comentario.
        /// </summary>
        public ManejadorComentario()
        {
            m_ManejadorDatosComentario = new ManejadorDatos<Comentario>(Constantes.APLICACION);
        }

        #endregion
    }
}

```



```
#region Métodos

/// <summary>
/// Agrega un Comentario en la tabla correspondiente.
/// </summary>
/// <param name="_Comentario">Comentario a insertar</param>
/// <returns>Retorna el identificador del Comentario.</returns>
internal long AgregarComentario(Comentario _Comentario)
{
    long Comentariold;
    try
    {
        Comentariold = m_ManejadorDatosComentario.InsertOP(_Comentario);
        return Comentariold;
    }
    catch (Exception exception)
    {
        throw exception;
    }
}

/// <summary>
/// Actualiza un Comentario en la tabla correspondiente.
/// </summary>
/// <param name="_Comentario">Comentario a insertar</param>
internal void ActualizarComentario(Comentario _Comentario)
{
    try
    {
        m_ManejadorDatosComentario.UpdateOP(_Comentario);
    }
    catch (Exception exception)
    {
        throw exception;
    }
}

/// <summary>
/// Lee un Comentario de la base de datos.
/// </summary>
/// <param name="_Comentariold">Identificador del Comentario que se quiere leer.</param>
/// <returns>Comentario leida.</returns>
internal Comentario LeerComentario(long _Comentariold)
{
    return m_ManejadorDatosComentario.GetOP(_Comentariold);
}

/// <summary>
/// Borra el Comentario de la base de datos.
/// </summary>
/// <param name="_Comentariold">Identificador del Comentario a borrar.</param>
internal void EliminarComentario(long _Comentariold)
{
}
```

```

        try
        {
            m_ManejadorDatosComentario.DeleteOP(_Comentarioid);
        }
        catch (Exception exception)
        {
            throw exception;
        }
    }

    /// <summary>
    /// Retorna una colección con todos los Comentario por la propiedad id de forma ascendente.
    /// </summary>
    /// <returns>Lista de todos los Comentario ordenados por id.</returns>
    internal IList<Comentario> LeerTodosComentarios()
    {
        return m_ManejadorDatosComentario.GetAllOP("comentarioid", "asc");
    }

    #endregion
}
}

```

```

ManejadorControl.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using VSComun.ManejadoresDatos;
using ADComun.Modelo;
using ADComun.Constantes;
//using GC = ADComun.Constantes;

namespace ADNucleo.Manejadores
{
    class ManejadorControl
    {
        #region Atributos

        /// <summary>
        /// Manejador de datos de tipo control.
        /// </summary>
        private ManejadorDatos<Control> m_ManejadorDatosControl;

        #endregion

        #region Constructores

        /// <summary>

```

```

    /// Contructor de la clase manejador control.
    /// </summary>
    public ManejadorControl()
    {
        m_ManejadorDatosControl = new ManejadorDatos<Control>(Constantes.APLICACION);
    }

#endregion

#region Métodos

    /// <summary>
    /// Agrega un Control en la tabla correspondiente.
    /// </summary>
    /// <param name="_Control">Control a insertar</param>
    /// <returns>Retorna el identificador del Control.</returns>
    internal long AgregarControl(Control _Control)
    {
        long ControlId;
        try
        {
            ControlId = m_ManejadorDatosControl.InsertOP(_Control);
            return ControlId;
        }
        catch (Exception exception)
        {
            throw exception;
        }
    }

    /// <summary>
    /// Actualiza un Control en la tabla correspondiente.
    /// </summary>
    /// <param name="_Control">Control a insertar</param>
    internal void ActualizarControl(Control _Control)
    {
        try
        {
            m_ManejadorDatosControl.UpdateOP(_Control);
        }
        catch (Exception exception)
        {
            //if (exception.InnerException != null)
            // {
            //     if (exception.InnerException.Message.IndexOf(CONSTRAINT_UNICO_CODIGO_CASO)
            // != -1)
            //     {
            //         throw new CasoConCodigoRepetidoExcepcion(_Caso.Codigo);
            //     }
            // }
            throw exception;
        }
    }
}

```

```

    /// <summary>
    /// Lee un Control de la base de datos.
    /// </summary>
    /// <param name="_ControllId">Identificador del Control que se quiere leer.</param>
    /// <returns>Control leído.</returns>
    internal Control LeerControl(long _ControllId)
    {
        return m_ManejadorDatosControl.GetOP(_ControllId);
    }

    /// <summary>
    /// Borra el Control de la base de datos.
    /// </summary>
    /// <param name="_ControllId">Identificador del Control a borrar.</param>
    internal void EliminarControl(long _ControllId)
    {
        try
        {
            m_ManejadorDatosControl.DeleteOP(_ControllId);
        }
        catch (Exception exception)
        {
            //if (exception.InnerException != null)
            //if
            (exception.InnerException.Message.IndexOf(CONSTRAINT_LLAVEFORANEA_IMGFORENSE) != -1)
            // {
            //     throw new MedioConImgForenseRelacionadaExcepcion(_Casold);
            // }
            //if
            throw exception;
        }
    }

    /// <summary>
    /// Calcula el porcentaje de cumplimiento del control.
    /// </summary>
    internal void CalcularPorcentajeControl()
    {
        IList<Control> _Controles = LeerTodosControles();
        foreach (Control control in _Controles)
        {
            if (control.Controles.Count() == 0)
            {
                if (control.PlanDeAccion.Count > 0)
                {
                    if (control.PlanDeAccion.First().PCumplimiento != null)
                    {
                        control.PCumplimiento = int.Parse(control.PlanDeAccion.First().PCumplimiento);
                    }
                    else
                    {
                        control.PCumplimiento = 0;
                    }
                }
            }
        }
    }

```

```

    }
    else
    {
        control.PCumplimiento = 0;
    }
    ActualizarControl(control);
}

foreach (Control control in _Controles)
{
    if (control.Controles.Count() != 0)
    {
        int count = 0;
        int porcentaje = 0;
        foreach (Control controlHijo in control.Controles)
        {
            porcentaje += controlHijo.PCumplimiento;
            count++;
        }
        int PorcentajePadre;
        PorcentajePadre = porcentaje / count;
        control.PCumplimiento = PorcentajePadre;
        ActualizarControl(control);
    }
}

/// <summary>
/// Retorna una colección con todos los controles por la propiedad id de forma ascendente.
/// </summary>
/// <returns>Lista de todos los controles ordenados por id.</returns>
internal IList<Control> LeerTodosControles()
{
    return m_ManejadorDatosControl.GetAllOP("controlId", "asc");
}

/// <summary>
/// Retorna una lista de controles que tengan relación con la norma.
/// </summary>
/// <param name="_Normald">Identificador de la norma.</param>
/// <returns>Lista de todos los controles ordenados por Id.</returns>
internal IList<Control> BuscarControlesPorNorma(long _Normald)
{
    return m_ManejadorDatosControl.FindObjectsByPropertyValue("Norma", _Normald, "Id", null);
}

/// <summary>
/// Busca los controles por nivel.
/// </summary>
/// <param name="_Nivel">Nivel superior a buscar.</param>
/// <returns>Lista de controles</returns>
internal IList<Control> BuscarControlesPorNivel(int _Nivel) // Nuevo +
{

```

```

string hql;
List<string> paramNames = new List<string>();
List<object> values = new List<object>();

hql = "from " + typeof(Control).Name + " as po";
if (_Nivel != null && _Nivel != 0)
{
    hql += " where po.Nivel < :Nivel";
    paramNames.Add("Nivel");
    values.Add(_Nivel);
}

return m_ManejadorDatosControl.FindObjectsByQuery(hql, paramNames.ToArray<string>(),
values.ToArray<object>());
}

/// <summary>
/// Busca los controles por norma y estado de implementación.
/// </summary>
/// <param name="_Normald">Id de la norma.</param>
/// <param name="_Implementado">Estado de implementación.</param>
/// <returns>Lista de controles.</returns>
internal IList<Control> BuscarControlesPorNormaElImplementado(long _Normald, bool
Implementado) // Nuevo +
{
    string hql;
    bool hasWhere = false;
    List<string> paramNames = new List<string>();
    List<object> values = new List<object>();

    hql = "from " + typeof(Control).Name + " as po";
    if (_Normald != null && _Normald != 0)
    {
        hasWhere = true;
        hql += " where po.Norma.Id = :Normald";
        paramNames.Add("Normald");
        values.Add(_Normald);
    }

    if (_Implementado != null)
    {
        if (!hasWhere)
        {
            hql += " where";
        }
        else
        {
            hql += " and";
        }
        hql += " po.Implementada = :Implementada";
        paramNames.Add("Implementada");
        values.Add(_Implementado);
    }
}

```

```

        return m_ManejadorDatosControl.FindObjectsByQuery(hql, paramNames.ToArray<string>()),
        values.ToArray<object>());
    }

    #endregion
}
}
}

```

ManejadorNorma.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using VSComun.ManejadoresDatos;
using ADComun.Modelo;
using ADComun.Constantes;

namespace ADNucleo.Manejadores
{
    /// <summary>
    /// Clase manejador norma.
    /// </summary>
    class ManejadorNorma
    {
        #region Atributos

        /// <summary>
        /// Manejador de datos tipo norma.
        /// </summary>
        private ManejadorDatos<Norma> m_ManejadorDatosNorma;

        #endregion

        #region Constructores

        /// <summary>
        /// Constructor de la clase.
        /// </summary>
        public ManejadorNorma()
        {
            m_ManejadorDatosNorma = new ManejadorDatos<Norma>(Constantes.APLICACION);
        }

        #endregion

        #region Métodos

        /// <summary>

```

```

/// Agrega una norma en la tabla correspondiente.
/// </summary>
/// <param name="_Norma">Norma a insertar</param>
/// <returns>Retorna el identificador de la norma.</returns>
internal long AgregarNorma(Norma _Norma)
{
    long normald;
    try
    {
        normald = m_ManejadorDatosNorma.InsertOP(_Norma);
        return normald;
    }
    catch (Exception exception)
    {
        throw exception;
    }
}

```

```

/// <summary>
/// Actualiza una norma en la tabla correspondiente.
/// </summary>
/// <param name="_Norma">norma a insertar</param>
internal void ActualizarNorma(Norma _Norma)
{
    try
    {
        m_ManejadorDatosNorma.UpdateOP(_Norma);
    }
    catch (Exception exception)
    {
        throw exception;
    }
}

```

```

/// <summary>
/// Lee una norma de la base de datos.
/// </summary>
/// <param name="_Normald">Identificador de la norma que se quiere leer.</param>
/// <returns>norma leida.</returns>
internal Norma LeerNorma(long _Normald)
{
    return m_ManejadorDatosNorma.GetOP(_Normald);
}

```

```

/// <summary>
/// Borra la norma de la base de datos.
/// </summary>
/// <param name="_Normald">Identificador de la norma a borrar.</param>
internal void EliminarNorma(long _Normald)
{
    try
    {
        m_ManejadorDatosNorma.DeleteOP(_Normald);
    }
}

```



```

        catch (Exception exception)
        {
            throw exception;
        }
    }

    /// <summary>
    /// Retorna una colección con todos las normas por la propiedad id de forma ascendente.
    /// </summary>
    /// <returns>Lista de todos las normas ordenados por id.</returns>
    internal IList<Norma> LeerTodasNormas()
    {
        return m_ManejadorDatosNorma.GetAllOP("normald", "asc");
    }

    /// <summary>
    /// Calcula el porcentaje de la norma.
    /// </summary>
    internal void CalcularPorcentajeNorma()
    {
        IList<Norma> _Normas = LeerTodasNormas();

        foreach (Norma norma in _Normas)
        {
            IList<Control> _Controles =
                Fachada.FachadaAD.Instancia.BuscarControlesPorNorma(norma.Id);
            int count = 0;
            int porcentaje = 0;
            foreach (Control control in _Controles)
            {
                if (control.Nivel == 1)
                {
                    porcentaje += control.PCumplimiento;
                    count++;
                }
            }
            int PorcentajeNorma;
            if (count != 0)
            {
                PorcentajeNorma = porcentaje / count;
                norma.PCumplimiento = PorcentajeNorma;
                ActualizarNorma(norma);
            }
        }
    }

    #endregion
}
}

```

ManejadorObservacion.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using VSComun.ManejadoresDatos;
using ADComun.Modelo;
using ADComun.Constantes;

namespace ADNucleo.Manejadores
{
    /// <summary>
    /// Clase de tipo manejador observación.
    /// </summary>
    class ManejadorObservacion
    {
        #region Atributos

        /// <summary>
        /// manejador de datos de tipo observación.
        /// </summary>
        private ManejadorDatos<Observacion> m_ManejadorDatosObservacion;

        #endregion

        #region Constructores

        /// <summary>
        /// Constructor de la clase.
        /// </summary>
        public ManejadorObservacion()
        {
            m_ManejadorDatosObservacion = new
            ManejadorDatos<Observacion>(Constantes.APLICACION);
        }

        #endregion

        #region Métodos

        /// <summary>
        /// Agrega una Observacion en la tabla correspondiente.
        /// </summary>
        /// <param name="_Observacion">Observacion a insertar</param>
        /// <returns>Retorna el identificador de la Observacion.</returns>
        internal long AgregarObservacion(Observacion _Observacion)
        {
            long observacionId;
            try
            {
                observacionId = m_ManejadorDatosObservacion.InsertOP(_Observacion);
            }
            catch { }
            return observacionId;
        }
    }
}
```

```

    }
    catch (Exception exception)
    {
        throw exception;
    }
}

/// <summary>
/// Actualiza una Observacion en la tabla correspondiente.
/// </summary>
/// <param name="_Observacion">Observacion a insertar</param>
internal void ActualizarObservacion(Observacion _Observacion)
{
    try
    {
        m_ManejadorDatosObservacion.UpdateOP(_Observacion);
    }
    catch (Exception exception)
    {
        throw exception;
    }
}

/// <summary>
/// Lee una Observacion de la base de datos.
/// </summary>
/// <param name="_observacionId">Identificador de la Observacion que se quiere leer.</param>
/// <returns>Observacion leida.</returns>
internal Observacion LeerObservacion(long _observacionId)
{
    return m_ManejadorDatosObservacion.GetOP(_observacionId);
}

/// <summary>
/// Borra la Observacion de la base de datos.
/// </summary>
/// <param name="_observacionId">Identificador de la Observacion a borrar.</param>
internal void EliminarObservacion(long _observacionId)
{
    try
    {
        m_ManejadorDatosObservacion.DeleteOP(_observacionId);
    }
    catch (Exception exception)
    {
        throw exception;
    }
}

/// <summary>
/// Retorna una colección con todas las Observaciones por la propiedad id de forma ascendente.
/// </summary>
/// <returns>Lista de todas las Observaciones ordenadas por id.</returns>
internal IList<Observacion> LeerTodasObservaciones()

```

```
        return m_ManejadorDatosObservacion.GetAllOP("observacionId", "asc");
    }

    #endregion
}
```

```
ManejadorPlanDeAccion.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using VSComun.ManejadoresDatos;
using ADComun.Modelo;
using GC = ADComun.Constantes;

namespace ADNucleo.Manejadores
{
    /// <summary>
    /// Clase para el manejador plan de acción.
    /// </summary>
    class ManejadorPlanDeAccion
    {
        #region Atributos

        /// <summary>
        /// Manejador de datos tipo plan de acción.
        /// </summary>
        private ManejadorDatos<PlanDeAccion> m_ManejadorDatosPlanDeAccion;

        #endregion

        #region Constructores

        /// <summary>
        /// Constructor de la clase.
        /// </summary>
        public ManejadorPlanDeAccion()
        {
            m_ManejadorDatosPlanDeAccion = new
            ManejadorDatos<PlanDeAccion>(GC.Constantes.APLICACION);
        }

        #endregion

        #region Métodos

        /// <summary>
        /// Agrega un PlanDeAccion en la tabla correspondiente.
```

```

/// </summary>
/// <param name="_PlanDeAccion">PlanDeAccion a insertar</param>
/// <returns>Retorna el identificador del PlanDeAccion.</returns>
internal long AgregarPlanDeAccion(PlanDeAccion _PlanDeAccion)
{
    long PlanDeAccionId;
    try
    {
        PlanDeAccionId = m_ManejadorDatosPlanDeAccion.InsertOP(_PlanDeAccion);
        return PlanDeAccionId;
    }
    catch (Exception exception)
    {
        throw exception;
    }
}

```

```

/// <summary>
/// Actualiza un PlanDeAccion en la tabla correspondiente.
/// </summary>
/// <param name="_PlanDeAccion">PlanDeAccion a insertar</param>
internal void ActualizarPlanDeAccion(PlanDeAccion _PlanDeAccion)
{
    try
    {
        m_ManejadorDatosPlanDeAccion.UpdateOP(_PlanDeAccion);
    }
    catch (Exception exception)
    {
        throw exception;
    }
}

```

```

/// <summary>
/// Lee un PlanDeAccion de la base de datos.
/// </summary>
/// <param name="_PlanDeAccionId">Identificador del PlanDeAccion que se quiere leer.</param>
/// <returns>PlanDeAccion leida.</returns>
internal PlanDeAccion LeerPlanDeAccion(long _PlanDeAccionId)
{
    return m_ManejadorDatosPlanDeAccion.GetOP(_PlanDeAccionId);
}

```

```

/// <summary>
/// Borra el PlanDeAccion de la base de datos.
/// </summary>
/// <param name="_PlanDeAccionId">Identificador del PlanDeAccion a borrar.</param>
internal void EliminarPlanDeAccion(long _PlanDeAccionId)
{
    try
    {
        m_ManejadorDatosPlanDeAccion.DeleteOP(_PlanDeAccionId);
    }
    catch (Exception exception)
    {
    }
}

```

```

        }
        throw exception;
    }

    /// <summary>
    /// Retorna una colección con todos los PlanDeAccion por la propiedad id de forma ascendente.
    /// </summary>
    /// <returns>Lista de todos los PlanDeAccion ordenados por id.</returns>
    internal IList<PlanDeAccion> LeerTodosPlanesDeAccion()
    {
        return m_ManejadorDatosPlanDeAccion.GetAllOP("planDeAccionId", "asc");
    }
    #endregion
}
}
}

```

```

ManejadorRol.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using VSComun.ManejadoresDatos;
using ADComun.Modelo;
using ADC = ADComun.Constantes;
using ADComun.Excepciones;
using NHibernate;

namespace ADNucleo.Manejadores
{
    /// <summary>
    /// Clase para el manejador Rol.
    /// </summary>
    class ManejadorRol
    {
        #region Atributos

        /// <summary>
        /// Manejador de datos tipo Rol.
        /// </summary>
        private ManejadorDatos<Rol> m_ManejadorDatosRol;

        #endregion

        #region Constructores

        /// <summary>
        /// Constructor de la clase.
        /// </summary>

```

```
public ManejadorRol()
{
    m_ManejadorDatosRol = new ManejadorDatos<Rol>(ADC.Constantes.APLICACION);
}

#endregion

#region Métodos

/// <summary>
/// Lee un rol de la base de datos.
/// </summary>
/// <param name="_RolId">Identificador del rol que se quiere leer.</param>
/// <returns>Rol cargado de la base de datos.</returns>
internal Rol LeerRol(long _RolId)
{
    return m_ManejadorDatosRol.GetOP(_RolId);
}

/// <summary>
/// Retorna una colección con todas los roles por la propiedad id de forma ascendente.
/// </summary>
/// <returns>Lista de todos los roles ordenados por id.</returns>
internal IList<Rol> LeerTodosRoles()
{
    return m_ManejadorDatosRol.GetAllOP("id", "asc");
}

#endregion
}
```

```
ManejadorUsuario.cs
using ADComun.Excepciones;
using ADComun.Modelo;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using VSComun.ManejadoresDatos;
using ADC = ADComun.Constantes;

namespace ADNucleo.Manejadores
{
    /// <summary>
    /// Clase para el manejador de usuarios.
    /// </summary>
    class ManejadorUsuario
    {
        #region Atributos
```

```

    /// <summary>
    /// Atributo para el manejo de excepciones.
    /// </summary>
    private static readonly string CONSTRAINT_UNICO_NOMBRE_USUARIO =
"UK_NOMBREINGRESO_USUARIO";

    /// <summary>
    /// Manjedaro de datos tipo usuario.
    /// </summary>
    private ManejadorDatos<Usuario> m_ManejadorDatosUsuario;

#endregion

#region Constructores

    /// <summary>
    /// Constructor de la clase manejador usuario.
    /// </summary>
    public ManejadorUsuario()
    {
        m_ManejadorDatosUsuario = new ManejadorDatos<Usuario>(ADC.Constantes.APLICACION);
    }

#endregion

#region Métodos

    /// <summary>
    /// Inserta un usuario a la tabla usuario.
    /// </summary>
    /// <param name="_Usuario">Usuario a insertar.</param>
    /// <returns>Retorna el identificador del usuario.</returns>
    internal long AgregarUsuario(Usuario _Usuario)
    {
        long Usuariold;
        try
        {
            Usuariold = m_ManejadorDatosUsuario.InsertOP(_Usuario);
            return Usuariold;
        }
        catch (Exception exception)
        {
            if (exception.InnerException != null)
            {
                if
(exception.InnerException.Message.IndexOf(CONSTRAINT_UNICO_NOMBRE_USUARIO) != -1)
                {
                    throw new UsuarioConNombreIngresoRepetidoExcepcion(_Usuario.NombreIngreso);
                }
            }
            throw exception;
        }
    }
}

```



```

/// <summary>
/// Actualiza un usuario en la tabla usuario.
/// </summary>
/// <param name="_Usuario">Usuario de tipo usuario a actualizar.</param>
internal void ActualizarUsuario(Usuario _Usuario)
{
    try
    {
        m_ManejadorDatosUsuario.UpdateOP(_Usuario);
    }
    catch (Exception excepcion)
    {
        if (excepcion.InnerException != null)
        {
            if
(excepcion.InnerException.Message.IndexOf(CONSTRAINT_UNICO_NOMBRE_USUARIO) != -1)
            {
                throw new UsuarioConNombreIngresoRepetidoExcepcion(_Usuario.NombreIngreso);
            }
        }
        throw excepcion;
    }
}

```

```

/// <summary>
/// Lee un usuario de la base de datos.
/// </summary>
/// <param name="_UsuarioId">Identificador del usuario que se quiere leer.</param>
/// <returns>Usuario cargado de la base de datos.</returns>
internal Usuario LeerUsuario(long _UsuarioId)
{
    return m_ManejadorDatosUsuario.GetOP(_UsuarioId);
}

```

```

/// <summary>
/// Lee un usuario de la base de datos
/// </summary>
/// <param name="_NombreIngreso">Nombre de ingreso del usuario</param>
/// <returns>Usuario cargado de la base de datos</returns>
internal Usuario LeerUsuarioPorNombreIngreso(string _NombreIngreso)
{
    return m_ManejadorDatosUsuario.GetObjectByPropertyValue("NombreIngreso",
_NombreIngreso, null, null);
}

```

```

/// <summary>
/// Borra un usuario de la base de datos.
/// </summary>
/// <param name="_UsuarioId">Identificador del usuario a borrar.</param>
internal void EliminarUsuario(long _UsuarioId)
{
    try
    {
        m_ManejadorDatosUsuario.DeleteOP(_UsuarioId);
    }
}

```

```

    }
    catch (Exception exception)
    {
        if (exception.InnerException != null)
        {
            //TODO restringir la eliminación del usuario ya que tiene relación con caso.
            //ii
            (exception.InnerException.Message.IndexOf(CONSTRAINT_LLAVEFORANEA_USUARIOFUENTEINFO
            RMACION) != -1)
            //i
            // throw new UsuarioConUsuarioFuenteInformacionRelacionadoExcepcion(_UsuarioId);
            //j
        }
        throw exception;
    }
}

/// <summary>
/// Devuelve una colección con todos los usuarios ordenados por la propiedad nombre en forma
ascendente.
/// </summary>
/// <returns>Lista de todos los usuarios ordenados por nombre.</returns>
internal IList<Usuario> LeerTodosUsuarios()
{
    return m_ManejadorDatosUsuario.GetAllOP("NombreIngreso", "asc");
}

/// <summary>
/// Retorna una colección con los usuarios que estén dentro de los límites establecidos.
/// </summary>
/// <param name="_Cuantos">Número máximo de resultados.</param>
/// <param name="_Desde">Primera fila para recibir.</param>
/// <returns>Retorna la lista de usuarios entre los parámetros establecidos.</returns>
internal IList<Usuario> LeerUsuarios(int _Cuantos, int _Desde)
{
    return m_ManejadorDatosUsuario.GetOPs(_Cuantos, _Desde);
}

/// <summary>
/// Devuelve una colección con todos los usuarios activos ordenados por nombre de forma
ascendente.
/// </summary>
/// <returns>Lista de usuarios activos por nombre.</returns>
internal IList<Usuario> BuscarUsuariosActivos()
{
    return m_ManejadorDatosUsuario.FindObjectsByPropertyValue("Activo", true, "NombreIngreso",
"asc");
}

/// <summary>
/// Devuelve una colección con todos los Usuarios inactivos ordenados por la propiedad nombre de
forma ascendente.
/// </summary>
/// <returns>Lista de Usuarios inactivos ordenada por nombre.</returns>

```

```

internal IList<Usuario> BuscarUsuariosInactivos()
{
    return m_ManejadorDatosUsuario.FindObjectsByPropertyValue("Activo", false, "NombreIngreso",
"asc");
}

/// <summary>
/// Devuelve una colección con los Usuarios activos que están en los límites establecidos.
/// </summary>
/// <param name="_Desde">Primera fila para recibir.</param>
/// <param name="_Cuantos">Número máximo de resultados.</param>
/// <returns>Retorna la lista de Usuarios entre los parámetros recibidos.</returns>
internal IList<Usuario> BuscarUsuariosActivos(int _Desde, int _Cuantos)
{
    return m_ManejadorDatosUsuario.FindObjectsByPropertyValue("Activo", true, "NombreIngreso",
"asc", _Cuantos, _Desde);
}

/// <summary>
/// Devuelve una colección con los Usuarios inactivos que están en los límites establecidos.
/// </summary>
/// <param name="_Desde">Primera fila para recibir.</param>
/// <param name="_Cuantos">Número máximo de resultados.</param>
/// <returns>Retorna la lista de Usuarios entre los parámetros recibidos.</returns>
internal IList<Usuario> BuscarUsuariosInactivos(int _Desde, int _Cuantos)
{
    return m_ManejadorDatosUsuario.FindObjectsByPropertyValue("Activo", false, "NombreIngreso",
"asc", _Cuantos, _Desde);
}

/// <summary>
/// Busca los Usuarios por nombre y estado (activo/inactivo).
/// </summary>
/// <param name="_Nombre">Nombre del Usuario. Busca fragmentos de palabra. En caso de no
querer.
/// usar este filtro, se debe enviar <code>null</code>.</param>
/// <param name="_Activo">Indica si el Usuario está activo o no. En caso de no querer.
/// usar este filtro, se debe enviar <code>null</code>.</param>
/// <returns></returns>
internal IList<Usuario> BuscarUsuariosPorNombreYEstado(string _Nombre, bool? _Activo) //
Nuevo +
{
    string hql;
    bool hasWhere = false;
    List<string> paramNames = new List<string>();
    List<object> values = new List<object>();

    hql = "from " + typeof(Usuario).Name + " as po";
    if (_Nombre != null && _Nombre.Trim().Length > 0)
    {
        hasWhere = true;
        hql += " where po.Nombre like :Nombre";
        paramNames.Add("Nombre");
        values.Add("%" + _Nombre + "%");
    }
}

```

```

    }

    if (_Activo != null)
    {
        if (!hasWhere)
        {
            hql += " where";
        }
        hql += " po.Activo = :Activo";
        paramNames.Add("Activo");
        values.Add(_Activo);
    }

    return m_ManejadorDatosUsuario.FindObjectsByQuery(hql, paramNames.ToArray<string>(),
    values.ToArray<object>());
}

/// <summary>
/// Busca los Usuarios por nombre y estado (activo/inactivo).
/// </summary>
/// <param name="_Nombre">Nombre del Usuario. Busca fragmentos de palabra. En caso de no
querer.
/// usar este filtro, se debe enviar <code>null</code>.</param>
/// <param name="_Activo">Indica si el Usuario está activo o no. En caso de no querer.
/// <param name="_Desde">Primera fila para recibir.</param>
/// <param name="_Cuantos">Número máximo de resultados.</param>
/// usar este filtro, se debe enviar <code>null</code>.</param>
/// <returns></returns>
internal IList<Usuario> BuscarUsuariosPorNombreYEstado(string _Nombre, bool? _Activo, int
_Desde, int _Cuantos) // Nuevo +
{
    string hql;
    bool hasWhere = false;
    List<string> paramNames = new List<string>();
    List<object> values = new List<object>();

    hql = "from " + typeof(Usuario).Name + " as po";
    if (_Nombre != null && _Nombre.Trim().Length > 0)
    {
        hasWhere = true;
        //Cambio Nombre por NombreIngreso --jnar--
        hql += " where po.NombreIngreso like :NombreIngreso";
        paramNames.Add("NombreIngreso");
        values.Add("%" + _Nombre + "%");
    }

    if (_Activo != null)
    {
        if (!hasWhere)
        {
            hql += " where";
        }
        else
    }

```

```

        hql += " and";
    }
    hql += " po.Activo = :Activo";
    paramNames.Add("Activo");
    values.Add(_Activo);
}

return m_ManejadorDatosUsuario.FindObjectsByQuery(hql, paramNames.ToArray<string>(),
values.ToArray<object>(), _Cuantos, _Desde);
}

/// <summary>
/// Busca los usuarios que tengan cierto fragmento de palabra en el nombre de ingreso.
/// </summary>
/// <param name="_NombreIngreso">Nombre ingreso(Completo o fragmento) a buscar.</param>
/// <returns>Lista de usuarios.</returns>
internal IList<Usuario> BuscarUsuariosPorNombreIngreso(string _NombreIngreso)
{
    return m_ManejadorDatosUsuario.FindObjectsByPropertyValueNonExact("NombreIngreso", "%"
+ _NombreIngreso + "%", "NombreIngreso", "asc");
}

/// <summary>
/// Devuelve la cantidad de usuarios que existen.
/// </summary>
/// <returns>La cantidad de usuarios.</returns>
internal long ContarTodosUsuarios()
{
    return m_ManejadorDatosUsuario.GetCountAllRecords();
}

/// <summary>
/// Devuelve la cantidad de Usuarios activos.
/// </summary>
/// <returns>Cantidad de Usuarios activos</returns>
internal long ContarUsuariosActivos()
{
    return m_ManejadorDatosUsuario.GetCountRecordsByPropertyValue("Activo", true);
}

/// <summary>
/// Devuelve la cantidad de Usuarios inactivos.
/// </summary>
/// <returns>Cantidad de Usuarios inactivos</returns>
internal long ContarUsuariosInactivos()
{
    return m_ManejadorDatosUsuario.GetCountRecordsByPropertyValue("Activo", false);
}

/// <summary>
/// Cuenta los usuarios por nombre y estado(activo/inactivo).
/// </summary>
/// <param name="_Nombre">Nombre del usuario a contar.</param>
/// <param name="_Activo">Estado del usuario(activo/inactivo).</param>

```

```

/// <returns>Cantidad de usuarios.</returns>
internal long ContarUsuariosPorNombreYEstado(string _Nombre, bool? _Activo) // Nuevo
{
    string where = "";
    bool addAnd = false;
    List<string> paramNames = new List<string>();
    List<object> values = new List<object>();

    if (_Nombre != null && _Nombre.Trim().Length > 0)
    {
        //Cambio de Nombre a NombreIngreso --jnar
        where += " NombreIngreso like :NombreIngreso";
        paramNames.Add("NombreIngreso");
        values.Add("%" + _Nombre + "%");
        addAnd = true;
    }

    if (_Activo != null)
    {
        if (addAnd)
        {
            where += " and";
        }
        where += " Activo = :Activo";
        paramNames.Add("Activo");
        values.Add(_Activo);
    }

    return m_ManejadorDatosUsuario.GetCountRecordsByQuery(where,
paramNames.ToArray<string>(), values.ToArray<object>());
}

#endregion
}
}

```

FachadaAD.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using ADNucleo.Manejadores;
using ADComun.Modelo;
using ADNucleo.Manejadores;
using ADComun.Modelo;
using ADComun.Modelo;

namespace ADNucleo.Fachada
{
    /// <summary>

```

```

/// Clase que implementa el patrón fachada en la aplicación.
/// </summary>
public class FachadaAD
{
    #region Atributos

    /// <summary>
    /// Referencia a la única instancia de la fachada.
    /// </summary>
    private static FachadaAD m_Instancia;

    /// <summary>
    /// Atributo de tipo manejador Comentario.
    /// </summary>
    ManejadorComentario m_ManejadorComentario;

    /// <summary>
    /// Atributo de tipo manejador Control.
    /// </summary>
    ManejadorControl m_ManejadorControl;

    /// <summary>
    /// Atributo de tipo manejador Norma.
    /// </summary>
    ManejadorNorma m_ManejadorNorma;

    /// <summary>
    /// Atributo de tipo manejador Observacion.
    /// </summary>
    ManejadorObservacion m_MenejadorObservacion;

    /// <summary>
    /// Atributo de tipo manejador PlanDeAccion.
    /// </summary>
    ManejadorPlanDeAccion m_ManejadorPlanDeAccion;

    /// <summary>
    /// Atributo de tipo manejador usuario.
    /// </summary>
    ManejadorUsuario m_ManejadorUsuario;

    /// <summary>
    /// Atributo de tipo manejador Rol.
    /// </summary>
    ManejadorRol m_ManejadorRol;

    #endregion

    #region Constructores

    /// <summary>
    /// Constructor de la clase la cual inicia todos los manejadores.
    /// </summary>
    public FachadaAD()

```

```

    m_ManejadorComentario = new ManejadorComentario();
    m_ManejadorControl = new ManejadorControl();
    m_ManejadorNorma = new ManejadorNorma();
    m_ManejadorObservacion = new ManejadorObservacion();
    m_ManejadorPlanDeAccion = new ManejadorPlanDeAccion();
    m_ManejadorUsuario = new ManejadorUsuario();
    m_ManejadorRol = new ManejadorRol();
}

#endregion

#region Propiedades

/// <summary>
/// Implementa una única instancia.
/// </summary>
public static FachadaAD Instancia
{
    get
    {
        if (m_Instancia == null)
        {
            m_Instancia = new FachadaAD();
        }
        return m_Instancia;
    }
}

#endregion Propiedades

#region Métodos

#region Métodos Comentario

/// <summary>
/// Agrega un Comentario en la tabla correspondiente.
/// </summary>
/// <param name="_Comentario">Comentario a insertar</param>
/// <returns>Retorna el identificador del Comentario.</returns>
public long AgregarComentario(Comentario _Comentario)
{
    return m_ManejadorComentario.AgregarComentario(_Comentario);
}

/// <summary>
/// Actualiza un Comentario en la tabla correspondiente.
/// </summary>
/// <param name="_Comentario">Comentario a insertar</param>
public void ActualizarComentario(Comentario _Comentario)
{
    m_ManejadorComentario.ActualizarComentario(_Comentario);
}

/// <summary>

```



```

/// Lee un Comentario de la tabla correspondiente.
/// </summary>
/// <param name="_Comentariold">Identificador del Comentario que se quiere leer.</param>
/// <returns>Comentario leida.</returns>
public Comentario LeerComentario(long _Comentariold)
{
    return m_ManejadorComentario.LeerComentario(_Comentariold);
}

```

```

/// <summary>
/// Borra el Comentario de la base de datos.
/// </summary>
/// <param name="_Comentariold">Identificador del Comentario a borrar.</param>
public void EliminarComentario(long _Comentariold)
{
    m_ManejadorComentario.EliminarComentario(_Comentariold);
}

```

```

/// <summary>
/// Lee todos los comentarios de la base de datos.
/// </summary>
/// <returns>Lista de comentarios.</returns>
public IList<Comentario> LeerTodosComentarios()
{
    return m_ManejadorComentario.LeerTodosComentarios();
}

```

```

#endregion Métodos Comentario

```

```

#region Métodos Control

```

```

/// <summary>
/// Agrega un Control en la tabla correspondiente.
/// </summary>
/// <param name="_Control">Control a insertar</param>
/// <returns>Retorna el identificador del Control.</returns>
public long AgregarControl(Control _Control)
{
    return m_ManejadorControl.AgregarControl(_Control);
}

```

```

/// <summary>
/// Actualiza un Control en la tabla correspondiente.
/// </summary>
/// <param name="_Control">Control a insertar</param>
public void ActualizarControl(Control _Control)
{
    m_ManejadorControl.ActualizarControl(_Control);
}

```

```

/// <summary>
/// Lee un Control de la base de datos.
/// </summary>
/// <param name="_Controlld">Identificador del Control que se quiere leer.</param>

```

```
    /// <returns>Control leído.</returns>
    public Control LeerControl(long _ControllId)
    {
        return m_ManejadorControl.LeerControl(_ControllId);
    }

    /// <summary>
    /// Borra el Control de la base de datos.
    /// </summary>
    /// <param name="_ControllId">Identificador del Control a borrar.</param>
    public void EliminarControl(long _ControllId)
    {
        m_ManejadorControl.EliminarControl(_ControllId);
    }

    /// <summary>
    /// Lee todos los controles de la base de datos.
    /// </summary>
    /// <returns>Lista de los controles leídos.</returns>
    public IList<Control> LeerTodosControles()
    {
        return m_ManejadorControl.LeerTodosControles();
    }

    /// <summary>
    /// Busca los controles por norma.
    /// </summary>
    /// <param name="_NormalId">id de la norma a buscar.</param>
    /// <returns>Lista de controles.</returns>
    public IList<Control> BuscarControlesPorNorma(long _NormalId)
    {
        return m_ManejadorControl.BuscarControlesPorNorma(_NormalId);
    }

    /// <summary>
    /// Busca los controles por norma y estado de implementación.
    /// </summary>
    /// <param name="_NormalId">Id de la norma.</param>
    /// <param name="_Implementado">Estado de implementación.</param>
    /// <returns>Lista de controles.</returns>
    public IList<Control> BuscarControlesPorNormaElImplementado(long _NormalId, bool
    _Implementado) // Nuevo +
    {
        return m_ManejadorControl.BuscarControlesPorNormaElImplementado(_NormalId,
    _Implementado);
    }

    /// <summary>
    /// Busca los controles por nivel.
    /// </summary>
    /// <param name="_Nivel">Nivel superior a buscar.</param>
    /// <returns>Lista de controles</returns>
    public IList<Control> BuscarControlesPorNivel(int _Nivel)
    {

```

```

return m_ManejadorControl.BuscarControlesPorNivel(_Nivel);
}

/// <summary>
/// Calcula el porcentaje de cumplimiento del control.
/// </summary>
public void CalcularPorcentajeControl()
{
    m_ManejadorControl.CalcularPorcentajeControl();
}

#endregion

#region Métodos Norma

/// <summary>
/// Agrega una norma en la tabla correspondiente.
/// </summary>
/// <param name="_Norma">Norma a insertar</param>
/// <returns>Retorna el identificador de la norma.</returns>
public long AgregarNorma(Norma _Norma)
{
    return m_ManejadorNorma.AgregarNorma(_Norma);
}

/// <summary>
/// Actualiza una norma en la tabla correspondiente.
/// </summary>
/// <param name="_Norma">norma a insertar</param>
public void ActualizarNorma(Norma _Norma)
{
    m_ManejadorNorma.ActualizarNorma(_Norma);
}

/// <summary>
/// Lee una norma de la base de datos.
/// </summary>
/// <param name="_Normald">Identificador de la norma que se quiere leer.</param>
/// <returns>norma leida.</returns>
public Norma LeerNorma(long _Normald)
{
    return m_ManejadorNorma.LeerNorma(_Normald);
}

/// <summary>
/// Borra la norma de la base de datos.
/// </summary>
/// <param name="_Normald">Identificador de la norma a borrar.</param>
public void EliminarNorma(long _Normald)
{
    m_ManejadorNorma.EliminarNorma(_Normald);
}

/// <summary>

```

```

    /// Retorna una colección con todos las normas por la propiedad id de forma ascendente.
    /// </summary>
    /// <returns>Lista de todos las normas ordenados por id.</returns>
    public IList<Norma> LeerTodasNormas()
    {
        return m_ManejadorNorma.LeerTodasNormas();
    }

    /// <summary>
    /// Calcula el porcentaje de la norma.
    /// </summary>
    public void CalcularPorcentajeNorma()
    {
        m_ManejadorNorma.CalcularPorcentajeNorma();
    }

#endregion Métodos Norma

#region Métodos Observacion

    /// <summary>
    /// Agrega una Observacion en la tabla correspondiente.
    /// </summary>
    /// <param name="_ Observacion">Observacion a insertar</param>
    /// <returns>Retorna el identificador de la Observacion.</returns>
    public long AgregarObservacion(Observacion _Observacion)
    {
        return m_MenejadorObservacion.AgregarObservacion(_ Observacion);
    }

    /// <summary>
    /// Actualiza una Observacion en la tabla correspondiente.
    /// </summary>
    /// <param name="_ Observacion">Observacion a insertar</param>
    public void ActualizarObservacion(Observacion _Observacion)
    {
        m_MenejadorObservacion.ActualizarObservacion(_ Observacion);
    }

    /// <summary>
    /// Lee una Observacion de la base de datos.
    /// </summary>
    /// <param name="_ observacionId">Identificador de la Observacion que se quiere leer.</param>
    /// <returns>Observacion leida.</returns>
    public Observacion LeerObservacion(long _observacionId)
    {
        return m_MenejadorObservacion.LeerObservacion(_observacionId);
    }

    /// <summary>
    /// Borra la Observacion de la base de datos.
    /// </summary>
    /// <param name="_ observacionId">Identificador de la Observacion a borrar.</param>
    public void EliminarObservacion(long _observacionId)

```

```

    {
        m_ManejadorObservacion.EliminarObservacion(_observacionId);
    }

    /// <summary>
    /// Retorna una colección con todos las Observaciones por la propiedad id de forma ascendente.
    /// </summary>
    /// <returns>Lista de todos las Observaciones ordenadas por id.</returns>
    public IList<Observacion> LeerTodasObservaciones()
    {
        return m_ManejadorObservacion.LeerTodasObservaciones();
    }

    #endregion Métodos Observacion

    #region Métodos Plan de Acción

    /// <summary>
    /// Agrega un PlanDeAccion en la tabla correspondiente.
    /// </summary>
    /// <param name="_PlanDeAccion">PlanDeAccion a insertar</param>
    /// <returns>Retorna el identificador del PlanDeAccion.</returns>
    public long AgregarPlanDeAccion(PlanDeAccion _PlanDeAccion)
    {
        return m_ManejadorPlanDeAccion.AgregarPlanDeAccion(_PlanDeAccion);
    }

    /// <summary>
    /// Agrega un PlanDeAccion en la tabla correspondiente.
    /// </summary>
    /// <param name="_PlanDeAccion">PlanDeAccion a insertar</param>
    public void ActualizarPlanDeAccion(PlanDeAccion _PlanDeAccion)
    {
        m_ManejadorPlanDeAccion.ActualizarPlanDeAccion(_PlanDeAccion);
    }

    /// <summary>
    /// Lee un PlanDeAccion de la base de datos.
    /// </summary>
    /// <param name="_PlanDeAccionId">Identificador del PlanDeAccion que se quiere leer.</param>
    /// <returns>PlanDeAccion leida.</returns>
    public PlanDeAccion LeerPlanDeAccion(long _PlanDeAccionId)
    {
        return m_ManejadorPlanDeAccion.LeerPlanDeAccion(_PlanDeAccionId);
    }

    /// <summary>
    /// Borra el PlanDeAccion de la base de datos.
    /// </summary>
    /// <param name="_PlanDeAccionId">Identificador del PlanDeAccion a borrar.</param>
    public void EliminarPlanDeAccion(long _PlanDeAccionId)
    {
        m_ManejadorPlanDeAccion.EliminarPlanDeAccion(_PlanDeAccionId);
    }
}

```

```
/// <summary>
/// Retorna una colección con todos los PlanDeAccion por la propiedad id de forma ascendente.
/// </summary>
/// <returns>Lista de todos los PlanDeAccion ordenados por id.</returns>
public IList<PlanDeAccion> LeerTodosPlanesDeAccion()
{
    return m_ManejadorPlanDeAccion.LeerTodosPlanesDeAccion();
}

#endregion Métodos Plan de Acción

#region Usuario

/// <summary>
/// Inserta un usuario a la tabla usuario.
/// </summary>
/// <param name="_Usuario">Usuario a insertar.</param>
/// <returns>Retorna el identificador del usuario.</returns>
public long AgregarUsuario(Usuario _Usuario)
{
    return m_ManejadorUsuario.AgregarUsuario(_Usuario);
}

/// <summary>
/// Actualiza un usuario en la tabla usuario.
/// </summary>
/// <param name="_Usuario">Usuario de tipo usuario a actualizar.</param>
public void ActualizarUsuario(Usuario _Usuario)
{
    m_ManejadorUsuario.ActualizarUsuario(_Usuario);
}

/// <summary>
/// Lee un usuario de la base de datos.
/// </summary>
/// <param name="_UsuarioId">Identificador del usuario que se quiere leer.</param>
/// <returns>Usuario cargado de la base de datos.</returns>
public Usuario LeerUsuario(long _UsuarioId)
{
    return m_ManejadorUsuario.LeerUsuario(_UsuarioId);
}

/// <summary>
/// Lee un usuario de la base de datos
/// </summary>
/// <param name="_NombreIngreso">Nombre de ingreso del usuario</param>
/// <returns>Usuario cargado de la base de datos</returns>
public Usuario LeerUsuarioPorNombreIngreso(string _NombreIngreso)
{
    return m_ManejadorUsuario.LeerUsuarioPorNombreIngreso(_NombreIngreso);
}

/// <summary>
```

```

    /// Borra un usuario de la base de datos.
    /// </summary>
    /// <param name="_Usuariold">Identificador del usuario a borrar.</param>
    public void EliminarUsuario(long _Usuariold)
    {
        m_ManejadorUsuario.EliminarUsuario(_Usuariold);
    }

    /// <summary>
    /// Devuelve una colección con todos los usuarios ordenados por la propiedad nombre en forma
    ascendente.
    /// </summary>
    /// <returns>Lista de todos los usuarios ordenados por nombre.</returns>
    public IList<Usuario> LeerTodosUsuarios()
    {
        return m_ManejadorUsuario.LeerTodosUsuarios();
    }

    /// <summary>
    /// Retorna una colección con los usuarios que estén dentro de los límites establecidos.
    /// </summary>
    /// <param name="_Cuantos">Número máximo de resultados.</param>
    /// <param name="_Desde">Primera fila para recibir.</param>
    /// <returns>Retorna la lista de usuarios entre los parámetros establecidos.</returns>
    public IList<Usuario> LeerUsuarios(int _Cuantos, int _Desde)
    {
        return m_ManejadorUsuario.LeerUsuarios(_Cuantos, _Desde);
    }

    /// <summary>
    /// Devuelve una colección con todos los usuarios activos ordenados por nombre de forma
    ascendente.
    /// </summary>
    /// <returns>Lista de usuarios activos por nombre.</returns>
    public IList<Usuario> BuscarUsuariosActivos()
    {
        return m_ManejadorUsuario.BuscarUsuariosActivos();
    }

    /// <summary>
    /// Devuelve una colección con todos los Usuarios inactivos ordenados por la propiedad nombre de
    forma ascendente.
    /// </summary>
    /// <returns>Lista de Usuarios inactivos ordenada por nombre.</returns>
    public IList<Usuario> BuscarUsuariosInactivos()
    {
        return m_ManejadorUsuario.BuscarUsuariosInactivos();
    }

    /// <summary>
    /// Devuelve una colección con los Usuarios activos que están en los límites establecidos.
    /// </summary>
    /// <param name="_Desde">Primera fila para recibir.</param>

```

```

    /// <param name="_Cuantos">Número máximo de resultados.</param>
    /// <returns>Retorna la lista de Usuarios entre los parámetros recibidos.</returns>
    public IList<Usuario> BuscarUsuariosActivos(int _Desde, int _Cuantos)
    {
        return m_ManejadorUsuario.BuscarUsuariosActivos();
    }

    /// <summary>
    /// Devuelve una colección con los Usuarios inactivos que están en los límites establecidos.
    /// </summary>
    /// <param name="_Desde">Primera fila para recibir.</param>
    /// <param name="_Cuantos">Número máximo de resultados.</param>
    /// <returns>Retorna la lista de Usuarios entre los parámetros recibidos.</returns>
    public IList<Usuario> BuscarUsuariosInactivos(int _Desde, int _Cuantos)
    {
        return m_ManejadorUsuario.BuscarUsuariosInactivos();
    }

    /// <summary>
    /// Busca los Usuarios por nombre y estado (activo/inactivo).
    /// </summary>
    /// <param name="_Nombre">Nombre del Usuario. Busca fragmentos de palabra. En caso de no
    querer.
    /// usar este filtro, se debe enviar <code>null</code>.</param>
    /// <param name="_Activo">Indica si el Usuario está activo o no. En caso de no querer.
    /// usar este filtro, se debe enviar <code>null</code>.</param>
    /// <returns></returns>
    public IList<Usuario> BuscarUsuariosPorNombreYEstado(string _Nombre, bool? _Activo) //
    Nuevo +
    {
        return m_ManejadorUsuario.BuscarUsuariosPorNombreYEstado(_Nombre, _Activo);
    }

    /// <summary>
    /// Busca los Usuarios por nombre y estado (activo/inactivo).
    /// </summary>
    /// <param name="_Nombre">Nombre del Usuario. Busca fragmentos de palabra. En caso de no
    querer.
    /// usar este filtro, se debe enviar <code>null</code>.</param>
    /// <param name="_Activo">Indica si el Usuario está activo o no. En caso de no querer.
    /// <param name="_Desde">Primera fila para recibir.</param>
    /// <param name="_Cuantos">Número máximo de resultados.</param>
    /// usar este filtro, se debe enviar <code>null</code>.</param>
    /// <returns></returns>
    public IList<Usuario> BuscarUsuariosPorNombreYEstado(string _Nombre, bool? _Activo, int
    Desde, int _Cuantos) // Nuevo +
    {
        return m_ManejadorUsuario.BuscarUsuariosPorNombreYEstado(_Nombre, _Activo, _Cuantos,
    Desde);
    }

    /// <summary>
    /// Busca los usuarios que tengan cierto fragmento de palabra en el nombre de ingreso.
    /// </summary>

```



```

    /// <param name="_NombreIngreso">Nombre ingreso(Completo o fragmento) a buscar.</param>
    /// <returns>Lista de usuarios.</returns>
    public IList<Usuario> BuscarUsuariosPorNombreIngreso(string _NombreIngreso)
    {
        return m_ManejadorUsuario.BuscarUsuariosPorNombreIngreso(_NombreIngreso);
    }

    /// <summary>
    /// Devuelve la cantidad de usuarios que existen.
    /// </summary>
    /// <returns>La cantidad de usuarios.</returns>
    public long ContarTodosUsuarios()
    {
        return m_ManejadorUsuario.ContarTodosUsuarios();
    }

    /// <summary>
    /// Devuelve la cantidad de Usuarios activos.
    /// </summary>
    /// <returns>Cantidad de Usuarios activos</returns>
    public long ContarUsuariosActivos()
    {
        return m_ManejadorUsuario.ContarUsuariosActivos();
    }

    /// <summary>
    /// Devuelve la cantidad de Usuarios inactivos.
    /// </summary>
    /// <returns>Cantidad de Usuarios inactivos</returns>
    public long ContarUsuariosInactivos()
    {
        return m_ManejadorUsuario.ContarUsuariosInactivos();
    }

    /// <summary>
    /// Cuenta los usuarios por nombre y estado(activo/inactivo).
    /// </summary>
    /// <param name="_Nombre">Nombre del usuario a contar.</param>
    /// <param name="_Activo">Estado del usuario(activo/inactivo).</param>
    /// <returns>Cantidad de usuarios.</returns>
    public long ContarUsuariosPorNombreYEstado(string _Nombre, bool? _Activo) // Nuevo
    {
        return m_ManejadorUsuario.ContarUsuariosPorNombreYEstado(_Nombre, _Activo);
    }

    #endregion

    #region Rol

    /// <summary>
    /// Devuelve una colección con todos los usuarios ordenados por la propiedad nombre en forma
    /// ascendente.
    /// </summary>

```

```

    /// <returns>Lista de todos los usuarios ordenados por nombre.</returns>
    public IList<Rol> LeerTodosRoles()
    {
        return m_ManejadorRol.LeerTodosRoles();
    }

    /// <summary>
    /// Lee un usuario de la base de datos.
    /// </summary>
    /// <param name="_Usuariold">Identificador del usuario que se quiere leer.</param>
    /// <returns>Usuario cargado de la base de datos.</returns>
    public Rol LeerRol(long _RolId)
    {
        return m_ManejadorRol.LeerRol(_RolId);
    }

#endregion

#endregion Métodos
}
}

```

```

Principal.aspx
<%@ Language="C#" AutoEventWireup="true" CodeFile="Principal.aspx.cs" Inherits="Principal" %>
<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>

<!DOCTYPE HTML>

<html>
<head>
    <title>Informatio Security Status - ISS</title>
    <link rel="stylesheet" href=" ../Styles/esquema.css" type="text/css" />
    <link rel="shortcut icon" href=" ../Imgs/Modificar.ico" />
    <script src="Util.js" type="text/javascript"></script>

    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
</head>

<body>
    <form id="form1" runat="server">
        <asp:ToolkitScriptManager ID="ToolkitScriptManagerSitio"
runat="server"></asp:ToolkitScriptManager>
        <div id="contenedor1">
            <div id="encabezado">

                <a href="#">
                    <div id="BotonMenu">
                        <asp:ImageButton ID="ButtonPPal" runat="server" OnClick="ButtonPPal_Click"
ImageUrl="~/Imgs/Boceto/botonMenuIn.png" />
                    </div>
                </a>
            <div id="encabezado1">

```

```
<div id="encabezado3">
  <asp:Label ID="LabelUsuario" runat="server" Text="Bienvenido: " align="center"
  bgcolor="white"></asp:Label>
  - Proyecto SGSI - ALFCOM S.A.
```

```
</div>
<div id="encabezado4">
```

```
<div class="boton">
  <asp:ImageButton ID="LinkButtonNorma" runat="server" Text="Norma"
  AlternateText="Norma"
  CommandName="DeleteComment" OnClick="LinkButtonNorma_Click"
  ForeColor="Black" Visible=<%# EsVisible("Norma") %>'></asp:ImageButton>
</div>
```

```
<div class="boton">
  <asp:ImageButton ID="LinkButtonControles" runat="server" Text="Control"
  AlternateText="Control"
  CommandName="DeleteComment" OnClick="LinkButtonControl_Click"
  ForeColor="Black" Visible=<%# EsVisible("Control") %>'></asp:ImageButton>
</div>
```

```
<div class="boton">
  <asp:ImageButton ID="LinkButtonPlan" runat="server" Text="Plan de Acción"
  AlternateText="Plan de Acción"
  CommandName="DeleteComment" OnClick="LinkButtonPlan_Click"
  ForeColor="Black" Visible=<%# EsVisible("Plan") %>'></asp:ImageButton>
</div>
```

```
<div class="boton">
  <asp:ImageButton ID="LinkButtonComentario" runat="server" Text="Comentario"
  AlternateText="Comentario"
  CommandName="DeleteComment" OnClick="LinkButtonComentario_Click"
  ForeColor="Black" Visible=<%# EsVisible("Comentario") %>'></asp:ImageButton>
</div>
```

```
<div class="boton">
  <asp:ImageButton ID="LinkButtonUsuario" runat="server" Text="Usuario"
  AlternateText="Usuario"
  CommandName="DeleteComment" OnClick="LinkButtonUsuario_Click"
  ForeColor="Black" Visible=<%# EsVisible("Usuario") %>'></asp:ImageButton>
</div>
<div class="boton">
  <asp:ImageButton ID="ImageButtonObservacion" runat="server" Text="Observación"
  AlternateText="Observación"
  CommandName="DeleteComment" OnClick="ImageButtonObservacion_Click"
  ForeColor="Black" Visible=<%# EsVisible("Observacion") %>'></asp:ImageButton>
</div>
<div class="boton">
  <asp:ImageButton ID="ImageButtonReporte" runat="server" Text="Reporte"
  AlternateText="Reporte"
  CommandName="DeleteComment" OnClick="ImageButtonReporte_Click"
  ForeColor="Black" Visible=<%# EsVisible("Reporte") %>'></asp:ImageButton>
</div>
```

```

        </div>
    </div>
    <div id="encabezado2">
        <div id="imagenbarras" style="float: left;">
            <asp:Image ID="ImagenBarras" runat="server"
ImageUrl="~/Imgs/Boceto/cumplimiento.png" />
        </div>
        <div id="porcentaje" style="font-size: 20px; font-family: Arial, Helvetica, sans-serif; color:
#800000; float: left; text-align:center">
            <br />
            <asp:Label ID="LabelPorcentaje" runat="server"> % </asp:Label>
        </div>
    </div>
</div>
<div id="submenu1">
    <asp:Label ID=LabelUbicacion runat="server" CssClass="linkmenu" ></asp:Label>
</div>
<div id="submenu2">
    <asp:ImageButton ID="ImageButton1" runat="server" src=" ../Imgs/Boceto/salir.png"
OnClick="CerrarSesion_Click" ToolTip="Cerrar Sesión"></asp:ImageButton>
</div>
<div id="cuerpo">
    <!--contenido---->
    <asp:UpdatePanel ID="UpdatePanel1" runat="server">
        <ContentTemplate>
            <iframe id="IFrameContenido" runat="server" scrolling="yes" class="iframe"></iframe>
        </ContentTemplate>
    </asp:UpdatePanel>
    <!--fin contenido----->
</div>
</div>
</form>
</body>
</html>

```

Principal.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

using ADComun.Modelo;
using ADWeb.Constantes;
using System.Windows.Forms;
using ADNucleo.Fachada;
using ADWeb.Seguridad;

using System.Diagnostics;
using System.ComponentModel;

/// <summary>
/// Clase principal que hereda de página segura y será la clase padre de todos los CRUD.
/// </summary>
public partial class Principal : PaginaSegura
{
    /// <summary>
    /// Sobreescribe el evento page_Load de página segura.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected override void OnPageLoad(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            IFrameContenido.Attributes["onLoad"] = "calcHeight()";
            string nombreUsuarioActual = HttpContext.Current.User.Identity.Name;
            //Usuario usuario = FachadaAD.Instancia.LeerUsuarioPorNombreIngreso(nombreUsuarioActual);
            LabelUsuario.Text = LabelUsuario.Text + nombreUsuarioActual;
            IFrameContenido.Attributes["src"] = Constantes.URL_GESTION_NORMAS;
            LabelUbicacion.Text = "Normas";
        }
        FachadaAD.Instancia.CalcularPorcentajeControl();
        FachadaAD.Instancia.CalcularPorcentajeNorma();
        CalcularPorcentajeTotal();
    }

    /// <summary>
    /// Evento que se dispara al dar clic sobre el botón opción.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    /// <param name="_Url"></param>
    protected void LinkButtonOpcion_Click(object sender, EventArgs e, string _Url)
    {
        System.Web.UI.WebControls.LinkButton linkButton =
        (System.Web.UI.WebControls.LinkButton)sender;
        IFrameContenido.Attributes["src"] = _Url;
    }
}
```

```

/// <summary>
/// Evento que se dispara al dar clic sobre el botón Norma.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void LinkButtonNorma_Click(object sender, EventArgs e)
{
    IFrameContenido.Attributes["src"] = Constantes.URL_GESTION_NORMAS;
    LabelUbicacion.Text = "Normas";
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón Control.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void LinkButtonControl_Click(object sender, EventArgs e)
{
    IFrameContenido.Attributes["src"] = Constantes.URL_GESTION_CONTROLES;
    LabelUbicacion.Text = "Controles";
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón plan.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void LinkButtonPlan_Click(object sender, EventArgs e)
{
    IFrameContenido.Attributes["src"] = Constantes.URL_GESTION_PLANESDEACCION;
    LabelUbicacion.Text = "Planes de acción";
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón Comentario.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void LinkButtonComentario_Click(object sender, EventArgs e)
{
    IFrameContenido.Attributes["src"] = Constantes.URL_GESTION_COMENTARIOS;
    LabelUbicacion.Text = "Comentarios";
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón usuario.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void LinkButtonUsuario_Click(object sender, EventArgs e)
{
    IFrameContenido.Attributes["src"] = Constantes.URL_GESTION_USUARIOS;
    LabelUbicacion.Text = "Usuarios";
}

```

```

}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón Obsevación.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonObservacion_Click(object sender, EventArgs e)
{
    IFrameContenido.Attributes["src"] = Constantes.URL_GESTION_OBSERVACIONES;
    LabelUbicacion.Text = "Observaciones";
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón reporte.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonReporte_Click(object sender, System.Web.UI.ImageClickEventArgs e)
{
    IFrameContenido.Attributes["src"] = Constantes.URL_REPORTES;
    LabelUbicacion.Text = "Reportes";
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón cerrar sesión.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void CerrarSesion_Click(object sender, System.Web.UI.ImageClickEventArgs e)
{
    Response.Redirect(Constantes.URL_LOGOUT, true);
}

/// <summary>
/// Método que calcula el porcentaje total de cumplimiento del sistema.
/// </summary>
public void CalcularPorcentajeTotal()
{
    IList<Norma> normas = FachadaAD.Instancia.LeerTodasNormas();
    int porcentajeTotal = 0;
    foreach (Norma norma in normas)
    {
        porcentajeTotal += norma.PCumplimiento;
    }
    porcentajeTotal = porcentajeTotal / normas.Count();
    LabelPorcentaje.Text = porcentajeTotal.ToString() + "%";

    if (porcentajeTotal == 0)
    {
        ImagenBarras.ImageUrl = "~/Imgs/Boceto/cumplimiento.png";
    }
    else if ((porcentajeTotal > 0) && (porcentajeTotal <= 17))
    {

```

```

        ImagenBarras.ImageUrl = "~/Imgs/Boceto/cumplimiento-1.png";
    }
    else if ((porcentajeTotal > 17) && (porcentajeTotal <= 34))
    {
        ImagenBarras.ImageUrl = "~/Imgs/Boceto/cumplimiento-2.png";
    }
    else if ((porcentajeTotal > 34) && (porcentajeTotal <= 51))
    {
        ImagenBarras.ImageUrl = "~/Imgs/Boceto/cumplimiento-3.png";
    }
    else if ((porcentajeTotal > 51) && (porcentajeTotal <= 68))
    {
        ImagenBarras.ImageUrl = "~/Imgs/Boceto/cumplimiento-4.png";
    }
    else if ((porcentajeTotal > 68) && (porcentajeTotal <= 85))
    {
        ImagenBarras.ImageUrl = "~/Imgs/Boceto/cumplimiento-5.png";
    }
    else
    {
        ImagenBarras.ImageUrl = "~/Imgs/Boceto/cumplimiento-6.png";
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón principal.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ButtonPPal_Click(object sender, EventArgs e)
{
    IFrameContenido.Attributes["src"] = Constantes.URL_GESTION_NORMAS;
}
}
}

```

```

Constantes.cs
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Linq;
using System.Web;

namespace ADWeb.Constantes
{
    /// <summary>
    /// Descripción breve de Constantes
    /// </summary>
    public class Constantes
    {
        public const string PARAMETRO_COMENTARIO_ID = "comentarioid";
    }
}

```



```
public const string PARAMETRO_CONTROL_ID = "controlId";
public const string PARAMETRO_NORMA_ID = "normalId";
public const string PARAMETRO_OBSERVACION_ID = "pbservacionId";
public const string PARAMETRO_PLANDEACCION_ID = "planDeAccionId";
public const string PARAMETRO_ROL_ID = "rolId";
public const string PARAMETRO_USUARIO_ID = "usuarioid";
```

```
public const string URL_GESTION_COMENTARIOS = "~/CRUD/Comentario/Gestion.aspx";
public const string URL_VER_COMENTARIO = "~/CRUD/Comentario/Ver.aspx";
public const string URL_EDITAR_COMENTARIO = "~/CRUD/Comentario/Editar.aspx";
public const string URL_CREAR_COMENTARIO = "~/CRUD/Comentario/Crear.aspx";
```

```
public const string URL_GESTION_CONTROLES = "~/CRUD/Control/Gestion.aspx";
public const string URL_VER_CONTROL = "~/CRUD/Control/Ver.aspx";
public const string URL_EDITAR_CONTROL = "~/CRUD/Control/Editar.aspx";
public const string URL_CREAR_CONTROL = "~/CRUD/Control/Crear.aspx";
```

```
public const string URL_GESTION_NORMAS = "~/CRUD/Norma/Gestion.aspx";
public const string URL_VER_NORMA = "~/CRUD/Norma/Ver.aspx";
public const string URL_EDITAR_NORMA = "~/CRUD/Norma/Editar.aspx";
public const string URL_CREAR_NORMA = "~/CRUD/Norma/Crear.aspx";
```

```
public const string URL_GESTION_OBSERVACIONES = "~/CRUD/Observacion/Gestion.aspx";
public const string URL_VER_OBSERVACION = "~/CRUD/Observacion/Ver.aspx";
public const string URL_EDITAR_OBSERVACION = "~/CRUD/Observacion/Editar.aspx";
public const string URL_CREAR_OBSERVACION = "~/CRUD/Observacion/Crear.aspx";
```

```
public const string URL_GESTION_PLANESDEACCION = "~/CRUD/PlanDeAccion/Gestion.aspx";
public const string URL_VER_PLANDEACCION = "~/CRUD/PlanDeAccion/Ver.aspx";
public const string URL_EDITAR_PLANDEACCION = "~/CRUD/PlanDeAccion/Editar.aspx";
public const string URL_CREAR_PLANDEACCION = "~/CRUD/PlanDeAccion/Crear.aspx";
```

```
public const string URL_GESTION_ROLES = "~/CRUD/Rol/Gestion.aspx";
public const string URL_VER_ROL = "~/CRUD/Rol/Ver.aspx";
public const string URL_EDITAR_ROL = "~/CRUD/Rol/Editar.aspx";
public const string URL_CREAR_ROL = "~/CRUD/Rol/Crear.aspx";
```

```
public const string URL_GESTION_USUARIOS = "~/CRUD/Usuario/Gestion.aspx";
public const string URL_CREAR_USUARIO = "~/CRUD/Usuario/Crear.aspx";
public const string URL_EDITAR_USUARIO = "~/CRUD/Usuario/Editar.aspx";
public const string URL_VER_USUARIO = "~/CRUD/Usuario/Ver.aspx";
```

```
public const string PAGINA_LOGIN = "Login.aspx";
public const string PAGINA_LOGOUT = "Logout.aspx";
public const string PAGINA_CAMBIO_CLAVE = "ActualizarClave.aspx";
public const string PAGINA_ERROR = "Error.aspx";
public const string PAGINA_PRINCIPAL = "Principal.aspx";
public const string URL_REPORTE =
```

```
"http://localhost:8080/pentaho/api/repos/%3Ahome%3Aadmin%3AReporte_implementacion.prpt/viewer";
```

```
public const string URL_LOGIN = "~/Login.aspx";
public const string URL_LOGOUT = "~/Logout.aspx";
public const string URL_PRINCIPAL = "~/Admin/Principal.aspx";
```

```

public static string PREFIJO_URL_SERVIDOR =
ConfigurationManager.AppSettings["ServerUrlPrefix"];
}
}

```

Icono.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ADWeb.App_Code
{
    /// <summary>
    /// Clase que enumera los tipos de iconos.
    /// </summary>
    public enum Icono
    {
        Alerta,
        Error,
        Informacion,
        Ok
    }
}

```

CRUD de comentarios.

Crear.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Crear.aspx.cs" Inherits="CRUD_Comentario_Crear" %>

<%@ MasterType VirtualPath="~/Sitio.master" %>

<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
    <table style="width: 80%;" border="0" cellpadding="0" cellspacing="0" align="center">
        <tr>
            <td class="style1">
                <h1>Crear Comentario</h1>
            </td>
        </tr>
        <tr>
            <td>
                <fieldset>
                    <legend>Datos del Comentario
                    </legend>
                    <table>
                        <tr>

```

```

        <td class="formTitle">Texto:
    </td>
    <td>
        <asp:TextBox ID="TextBoxTexto" runat="server" Width="400px" MaxLength="5"
        TextMode="MultiLine"></asp:TextBox>

        <asp:RequiredFieldValidator ID="RequiredFieldValidatorTexto" runat="server"
        ValidationGroup="validar"
            ControlToValidate="TextBoxTexto" ErrorMessage="Campo Obligatorio"
            CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
        <asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderTexto" runat="server"
        PopupPosition="BottomLeft"
            TargetControlID="RequiredFieldValidatorTexto"
            WarningIconImageUrl="~/Imgs/Advertencia.png">
        </asp:ValidatorCalloutExtender>
    </td>
</tr>
<tr>
    <td class="formTitle">Norma:
    </td>
    <td>
        <asp:DropDownList ID="DropDownListNorma" runat="server" AutoPostBack="true"
            DataTextField="Nombre" DataValueField="Id" CssClass="DropDownListMedio"
            OnSelectedIndexChanged="DropDownListNorma_SelectedIndexChanged" Width="300">
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td class="formTitle">Control:
    </td>
    <td>
        <asp:DropDownList ID="DropDownListControl" runat="server" AutoPostBack="true"
            DataTextField="Nombre" DataValueField="Id" CssClass="DropDownListMedio"
            Width="300">
        </asp:DropDownList>
    </td>
</tr>
</table>

</fieldset>
</td>
</tr>

<tr>
    <td style="text-align: right">
        <asp:ImageButton ID="ImageButtonCrear" runat="server" ImageUrl="~/Imgs/Ok.png"
        OnClick="ImageButtonCrear_Click" ToolTip="Crear" ValidationGroup="validar" />
        <asp:ImageButton ID="ImageButtonCancelar" runat="server" ImageUrl="~/Imgs/Cancel.png"
        OnClick="ImageButtonCancelar_Click" ToolTip="Cancelar" CausesValidation="False" />
    </td>
</tr>

</table>
</asp:Content>

```

Crear.aspx.cs

```
using ADWeb.App_Code;
using ADWeb.Constantes;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using ADNucleo.Fachada;
using ADComun.Modelo;

/// <summary>
/// Clase para crear el comentario.
/// </summary>
public partial class CRUD_Comentario_Crear : System.Web.UI.Page
{
    /// <summary>
    /// Lógica para que cargue los datos cuando ingrese a la página.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            DropDownListNorma.DataSource = FachadaAD.Instancia.LeerTodasNormas();
            DropDownListNorma.DataBind();
            DropDownListNorma.Items.Insert(0, new ListItem("", ""));
        }
    }

    /// <summary>
    /// Evento que se dispara al dar clic sobre el botón crear.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void ImageButtonCrear_Click(object sender, ImageClickEventArgs e)
    {
        Comentario comentario = new Comentario();

        comentario.Texto = TextBoxTexto.Text;
        ADComun.Modelo.Control control =
        FachadaAD.Instancia.LeerControl(long.Parse(DropDownListControl.SelectedValue));
        comentario.Control = control;
        string nombreUsuarioActual = HttpContext.Current.User.Identity.Name;
        Usuario usuario = FachadaAD.Instancia.LeerUsuarioPorNombreIngreso(nombreUsuarioActual);
        comentario.Usuario = usuario;

        try
        {
            FachadaAD.Instancia.AgregarComentario(comentario);
            Master.MostrarMensaje("Operación exitosa", "El comentario se creó correctamente.",
            Icono.Informacion, Constantes.URL_GESTION_COMENTARIOS);
        }
    }
}
```

```

}
catch (Exception ex)
{
    Master.MostrarMensaje("Error", "Error creando comentario - " + ex.Message, Icono.Error,
    Constantes.URL_GESTION_COMENTARIOS);
}
}

```

```

/// <summary>
/// Evento que se dispara al dar clic sobre el botón cancelar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCancelar_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect(Constantes.URL_GESTION_COMENTARIOS, true);
}

```

```

/// <summary>
/// Evento que se dispara al cambiar el valor del dropDownList.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void DropDownListNorma_SelectedIndexChanged(object sender, EventArgs e)
{
    long normald;
    long.TryParse(DropDownListNorma.SelectedValue, out normald);
}

```

```

DropDownListControl.DataSource = FachadaAD.Instancia.BuscarControlesPorNorma(normald);
DropDownListControl.DataBind();
}
}

```

Gestion.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Gestion.aspx.cs" Inherits="CRUD_Comentario_Gestion" %>

```

```

<%@ MasterType VirtualPath="~/Sitio.master" %>

```

```

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
<table style="width: 80%;" border="0" cellpadding="0" cellspacing="0" align="center">
<tr>
<td class="style1">
<h1>Gestión de Comentarios</h1>
</td>
</tr>

```

```

<tr>
<td style="text-align: right">
<asp:ImageButton ID="ImageButtonCrear" OnClick="ImageButtonCrear_Click" runat="server"
ImageUrl="~/Imgs/Crear.png" ToolTip="Crear" />
</td>

```

```

</tr>
<tr>
<td style="text-align: center">
<asp:UpdatePanel ID="UpdatePanelGridView" runat="server">
<ContentTemplate>
<fieldset>
<legend>Comentarios
</legend>
<asp:GridView ID="GridViewComentarios" runat="server"
AutoGenerateColumns="False"
CellPadding="4" DataKeyNames="Id" ForeColor="#333333" GridLines="None"
HorizontalAlign="Center" OnRowCommand="GridViewComentarios_RowCommand"
EmptyDataText="No hay Comentarios registrados" Width="100%">
<AlternatingRowStyle BackColor="White" ForeColor="#666666" />
<Columns>
<asp:BoundField DataField="Texto" HeaderText="Texto">
<ItemStyle HorizontalAlign="Left" />
</asp:BoundField>
<asp:BoundField DataField="Control.Nombre" HeaderText="Control" />
<asp:BoundField DataField="Control.Norma.Nombre" HeaderText="Norma" />
<asp:BoundField DataField="Usuario.Nombre" HeaderText="Usuario" />
<asp:TemplateField HeaderText="Opciones">
<ItemStyle Width="100px" />
<ItemTemplate>
<asp:ImageButton ID="ImageButtonVer" runat="server"
ImageUrl='../Imgs/Ver.png'
ToolTip="Ver detalle" CommandName="ver"
CommandArgument="<%# Container.DataItemIndex %%" />
</ItemTemplate>
</asp:TemplateField>
</Columns>
<EditRowStyle BackColor="#999999" />
<FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
<HeaderStyle BackColor="#000000" Font-Bold="True" ForeColor="White" />
<PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />
<RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
<SelectedRowStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333"
/>
<SortedAscendingCellStyle BackColor="#E9E7E2" />
<SortedAscendingHeaderStyle BackColor="#506C8C" />
<SortedDescendingCellStyle BackColor="#FFFDF8" />
<SortedDescendingHeaderStyle BackColor="#6F8DAE" />
</asp:GridView>
</fieldset>
</ContentTemplate>
</Triggers>
<asp:PostBackTrigger ControlID="GridViewComentarios" />
</Triggers>

```

```

        </asp:UpdatePanel>
    </td>

</tr>
<p>
    <asp:CustomValidator ID="CustomValidatorEliminar" runat="server"
        ClientValidationFunction="confirmarEliminacion"
        ValidationGroup="eliminar"></asp:CustomValidator>
</p>

</table>

</asp:Content>

```

```

Gestion.aspx.cs
using ADComun.Modelo;
using ADNucleo.Fachada;
using ADComun.Modelo;
using ADWeb.App_Code;
using ADWeb.Constantes;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

/// <summary>
/// Clase para la gestión de comentarios.
/// </summary>
public partial class CRUD_Comentario_Gestion : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            CargarDatos();
        }
    }

    /// <summary>
    /// Page_load de la gestión.
    /// </summary>
    private void CargarDatos()
    {
        IList<Comentario> _Comentario = FachadaAD.Instancia.LeerTodosComentarios();

        GridViewComentarios.DataSource = _Comentario;
        GridViewComentarios.DataBind();
    }

    /// <summary>

```

```

/// Lógica para los botones de la grilla.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void GridViewComentarios_RowCommand(object sender, GridViewCommandEventArgs e)
{
    int rowIndex = Convert.ToInt32(e.CommandArgument);
    long comentarioId = (long)GridViewComentarios.DataKeys[rowIndex].Value;

    switch (e.CommandName)
    {
        case "ver":
            Server.Transfer(Constants.URL_VER_COMENTARIO + "?" +
                Constants.PARAMETRO_COMENTARIO_ID + "=" + comentarioId, false);
            break;
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón crear.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCrear_Click(object sender, ImageClickEventArgs e)
{
    Server.Transfer(Constants.URL_CREAR_COMENTARIO);
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón filtrar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonFiltrar_Click(object sender, ImageClickEventArgs e)
{
    CargarDatos();
}

```

Ver.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Ver.aspx.cs" Inherits="CRUD_Comentario_Ver" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">

    <table style="width: 80%; border="0" cellpadding="0" cellspacing="0" align="center">
        <tr>
            <td class="style1">
                <h1>Ver Comentario</h1>
            </td>
        </tr>
    </table>

```



```

</tr>
<td>
<fieldset>
<legend>Datos del Comentario
</legend>
<table>
<tr>
<td class="formTitle">Texto:
</td>
<td>
<asp:TextBox ID="TextBoxTexto" runat="server" Width="400px" MaxLength="5"
TextMode="SingleLine" ReadOnly="True"></asp:TextBox>
</td>
</tr>
<tr>
<td class="formTitle">Norma:
</td>
<td>
<asp:TextBox ID="TextBoxNorma" runat="server" Width="400px" MaxLength="5"
TextMode="SingleLine" ReadOnly="True"></asp:TextBox>
</td>
</tr>
<tr>
<td class="formTitle">Control:
</td>
<td>
<asp:TextBox ID="TextBoxControl" runat="server" Width="400px" MaxLength="5"
TextMode="SingleLine" ReadOnly="True"></asp:TextBox>
</td>
</tr>
<tr>
<td class="formTitle">Usuario:
</td>
<td>
<asp:TextBox ID="TextBoxUsuario" runat="server" Width="400px" MaxLength="5"
TextMode="SingleLine" ReadOnly="True"></asp:TextBox>
</td>
</tr>
</table>

</fieldset>
</td>
</tr>

<tr>
<td style="text-align: right">&nbsp;   
<asp:ImageButton ID="ImageButtonCancelar" runat="server" ImageUrl="~/Imgs/Cancel.png"
OnClick="ImageButtonCancelar_Click" ToolTip="Cancelar" CausesValidation="False" />
</td>
</tr>

</table>
</asp:Content>

```

Ver.aspx

```
using ADComun.Modelo;
using ADNucleo.Fachada;
using ADWeb.Constantes;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using ADCM = ADComun.Modelo;

/// <summary>
/// Clase para ver los datos del comentario.
/// </summary>
public partial class CRUD_Comentario_Ver : System.Web.UI.Page
{
    /// <summary>
    /// Lógica para que cargue los datos cuando ingrese a la página.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            long ComentarioId =
            long.Parse(Request.QueryString[Constantes.PARAMETRO_COMENTARIO_ID]);

            Comentario comentario;
            comentario = FachadaAD.Instancia.LeerComentario(ComentarioId);
            TextBoxTexto.Text = comentario.Texto;
            TextBoxNorma.Text = comentario.Control.Norma.Nombre;
            TextBoxControl.Text = comentario.Control.Nombre;
            TextBoxUsuario.Text = comentario.Usuario.NombreIngreso;
        }
    }

    /// <summary>
    /// Evento que se dispara al dar clic sobre el botón cancelar.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void ImageButtonCancelar_Click(object sender, ImageClickEventArgs e)
    {
        Response.Redirect(Constantes.URL_GESTION_COMENTARIOS, true);
    }
}
```

CRUD control

Crear.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Crear.aspx.cs" Inherits="CRUD_Control_Crear" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>
<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
  <table style="width: 80%;" border="0" cellpadding="0" cellspacing="0" align="center">
    <tr>
      <td class="style1">
        <h1>Crear Control</h1>
      </td>
    </tr>
    <tr>
      <td>
        <fieldset>
          <legend>Datos del control</legend>
          <table>
            <tr>
              <td class="formTitle">Nombre:</td>
              <td>
                <asp:TextBox ID="TextBoxNombre" runat="server" Width="400px"
                MaxLength="50"></asp:TextBox>
                <asp:RequiredFieldValidator ID="RequiredFieldValidatorNombre" runat="server"
                ValidationGroup="validar"
                ControlToValidate="TextBoxNombre" ErrorMessage="Campo Obligatorio"
                CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
                <asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderNombre" runat="server"
                PopupPosition="BottomLeft"
                TargetControlID="RequiredFieldValidatorNombre"
                WarningIconImageUrl="~/Imgs/Advertencia.png">
                </asp:ValidatorCalloutExtender>
              </td>
            </tr>
            <tr>
              <td class="formTitle">Aplicable:</td>
              <td style="margin-left: 40px">
                <asp:CheckBox ID="CheckBoxAplicable" runat="server" Width="100px"
                MaxLength="5"></asp:CheckBox>
              </td>
            </tr>
            <tr>
              <td class="formTitle">Implementada:</td>
              <td style="margin-left: 40px">
                <asp:CheckBox ID="CheckBoxImplementada" runat="server" Width="100px">
              </td>
            </tr>
          </table>
        </fieldset>
      </td>
    </tr>
  </table>
</asp:Content>
```

```

MaxLength="5"></asp:CheckBox>
</td>
</tr>
<tr>
<td class="formTitle">Nivel:
</td>
<td style="margin-left: 40px">
<asp:DropDownList ID="DropDownListNivel" runat="server" AutoPostBack="true"
OnSelectedIndexChanged="DropDownListNivel_SelectedIndexChanged">
<asp:ListItem Value="1" Text="1">1</asp:ListItem>
<asp:ListItem Value="2" Text="2">2</asp:ListItem>
<asp:ListItem Value="3" Text="3">3</asp:ListItem>
<asp:ListItem Value="4" Text="4">4</asp:ListItem>
<asp:ListItem Value="5" Text="5">5</asp:ListItem>
<asp:ListItem Value="6" Text="6">6</asp:ListItem>
<asp:ListItem Value="7" Text="7">7</asp:ListItem>
<asp:ListItem Value="8" Text="8">8</asp:ListItem>
<asp:ListItem Value="9" Text="9">9</asp:ListItem>
<asp:ListItem Value="10" Text="10">10</asp:ListItem>
</asp:DropDownList>
</td>
</tr>
<tr>
<td class="formTitle">Descripción:
</td>
<td>
<asp:TextBox ID="TextBoxDescripcion" runat="server" Width="400px"
MaxLength="5" TextMode="MultiLine"></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidatorDescripcion" runat="server"
ValidationGroup="validar"
ControlToValidate="TextBoxDescripcion" ErrorMessage="Campo Obligatorio"
CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
<asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderDescripcion"
runat="server" PopupPosition="BottomLeft"
TargetControlID="RequiredFieldValidatorDescripcion"
WarningIconImageUrl="~/Imgs/Advertencia.png">
</asp:ValidatorCalloutExtender>
</td>
</tr>
<tr>
<td class="formTitle">Norma:
</td>
<td>
<asp:DropDownList ID="DropDownListNorma" runat="server" AutoPostBack="false"
DataTextField="Nombre" DataValueField="Id" CssClass="DropDownListMedio">
</asp:DropDownList>
</td>
</tr>
<tr>
<td class="formTitle">Control al que pertenece:
</td>
<td>
<asp:DropDownList ID="DropDownListControles" runat="server"

```

```

AutoPostBack="false"
        DataTextField="Nombre" DataValueField="Id" CssClass="DropDownListMedio">
        </asp:DropDownList>
    </td>
</tr>
</table>

</fieldset>
</td>
</tr>

<tr>
    <td style="text-align: right">
        <asp:ImageButton ID="ImageButtonCrear" runat="server" ImageUrl="~/Imgs/Ok.png"
OnClick="ImageButtonCrear_Click" ToolTip="Crear" ValidationGroup="validar" />
        <asp:ImageButton ID="ImageButtonCancelar" runat="server" ImageUrl="~/Imgs/Cancel.png"
OnClick="ImageButtonCancelar_Click" ToolTip="Cancelar" CausesValidation="False" />
    </td>
</tr>

</table>
</asp:Content>

```

```

Crear.aspx.cs
using ADCM = ADComun.Modelo;
using ADWeb.App_Code;
using ADWeb.Constantes;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using ADNucleo.Fachada;
using ADComun.Modelo;

/// <summary>
/// Clase para crear el usuario.
/// </summary>
public partial class CRUD_Control_Crear : System.Web.UI.Page
{
    /// <summary>
    /// Lógica para que cargue los datos cuando ingrese a la página.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            DropDownListNorma.DataSource = FachadaAD.Instancia.LeerTodasNormas();
            DropDownListNorma.DataBind();
        }
    }
}

```

```

        DropDownListControles.Items.Insert(0, new ListItem("", ""));
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón crear.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCrear_Click(object sender, ImageClickEventArgs e)
{
    ADCM.Control control = new ADCM.Control();

    control.Nombre = TextBoxNombre.Text;
    control.Aplicable = CheckBoxAplicable.Checked;
    control.Implementada = CheckBoxImplementada.Checked;
    control.Nivel = int.Parse(DropDownListNivel.SelectedValue);
    control.Descripcion = TextBoxDescripcion.Text;

    if (DropDownListNorma.SelectedValue.Length != 0)
    {
        Norma norma =
        FachadaAD.Instancia.LeerNorma(long.Parse(DropDownListNorma.SelectedValue));
        control.Norma = norma;
    }
    if (DropDownListControles.SelectedValue.Length != 0)
    {
        ADCM.Control padre =
        FachadaAD.Instancia.LeerControl(long.Parse(DropDownListControles.SelectedValue));
        control.Padres.Add(padre);
    }

    try
    {
        FachadaAD.Instancia.AgregarControl(control);
        Master.MostrarMensaje("Operación exitosa", "El control se creó correctamente.",
        Icono.Informacion, Constantes.URL_GESTION_CONTROLES);
    }
    catch (Exception ex)
    {
        Master.MostrarMensaje("Error", "Error creando control - " + ex.Message, Icono.Error,
        Constantes.URL_GESTION_CONTROLES);
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón cancelar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCancelar_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect(Constantes.URL_GESTION_CONTROLES, true);
}

```

```

}

/// <summary>
/// Evento que se dispara al cambiar el valor del dropDownList.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void DropDownListNivel_SelectedIndexChanged(object sender, EventArgs e)
{
    DropDownListControles.DataSource =
    FachadaAD.Instancia.BuscarControlesPorNivel(int.Parse(DropDownListNivel.SelectedValue));
    DropDownListControles.DataBind();
}
}
}

```

```

Editar.aspx
<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Editar.aspx.cs" Inherits="CRUD_Control_Editar" %>

<%@ MasterType VirtualPath="~/Sitio.master" %>

<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
<table style="width: 80%; border="0" cellpadding="0" cellspacing="0" align="center">
<tr>
<td class="style1">
<h1>Editar Control</h1>
</td>
</tr>
<tr>
<td>
<fieldset>
<legend>Datos del control</legend>
<table>
<tr>
<td class="formTitle">Nombre:
</td>
<td>
<asp:TextBox ID="TextBoxNombre" runat="server" Width="400px"
MaxLength="50"></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidatorNombre" runat="server"
ValidationGroup="validar"
ControlToValidate="TextBoxNombre" ErrorMessage="Campo Obligatorio"
CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
<asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderNombre" runat="server"
PopupPosition="BottomLeft"
TargetControlID="RequiredFieldValidatorNombre"

```

```

WarningIconImageUrl="~/Imgs/Advertencia.png">
    </asp:ValidatorCalloutExtender>
    </td>
</tr>
<tr>
    <td class="formTitle">Aplicable:
    </td>
    <td style="margin-left: 40px">
        <asp:CheckBox ID="CheckBoxAplicable" runat="server" Width="100px"
MaxLength="5"></asp:CheckBox>
    </td>
</tr>
<tr>
    <td class="formTitle">Implementada:
    </td>
    <td style="margin-left: 40px">
        <asp:CheckBox ID="CheckBoxImplementada" runat="server" Width="100px"
MaxLength="5"></asp:CheckBox>
    </td>
</tr>
<tr>
    <td class="formTitle">Nivel:
    </td>
    <td style="margin-left: 40px">
        <asp:DropDownList ID="DropDownListNivel" runat="server" AutoPostBack="true"
OnSelectedIndexChanged="DropDownListNivel_SelectedIndexChanged">
            <asp:ListItem Value="1" Text="1">1</asp:ListItem>
            <asp:ListItem Value="2" Text="2">2</asp:ListItem>
            <asp:ListItem Value="3" Text="3">3</asp:ListItem>
            <asp:ListItem Value="4" Text="4">4</asp:ListItem>
            <asp:ListItem Value="5" Text="5">5</asp:ListItem>
            <asp:ListItem Value="6" Text="6">6</asp:ListItem>
            <asp:ListItem Value="7" Text="7">7</asp:ListItem>
            <asp:ListItem Value="8" Text="8">8</asp:ListItem>
            <asp:ListItem Value="9" Text="9">9</asp:ListItem>
            <asp:ListItem Value="10" Text="10">10</asp:ListItem>
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td class="formTitle">Descripción:
    </td>
    <td>
        <asp:TextBox ID="TextBoxDescripcion" runat="server" Width="400px"
MaxLength="5" TextMode="MultiLine"></asp:TextBox>
        <asp:RequiredFieldValidator ID="RequiredFieldValidatorDescripcion" runat="server"
ValidationGroup="validar"
ControlToValidate="TextBoxDescripcion" ErrorMessage="Campo Obligatorio"
CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
        <asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderDescripcion"
runat="server" PopupPosition="BottomLeft"
TargetControlID="RequiredFieldValidatorDescripcion"
WarningIconImageUrl="~/Imgs/Advertencia.png">

```



```

        </asp:ValidatorCalloutExtender>
    </td>
</tr>
<tr>
    <td class="formTitle">Porcentaje cumplimiento:
    </td>
    <td>
        <asp:TextBox ID="TextBoxPCumplimiento" runat="server" Width="100"
ReadOnly="True"></asp:TextBox>
        <asp:RegularExpressionValidator ID="RegularExpressionValidatorPCumplimiento"
runat="server" ControlToValidate="TextBoxPCumplimiento"
ErrorMessage="Campo debe ser número" CssClass="validador" Font-Bold="True"
Font-Size="0pt" ValidationGroup="validar"
ValidationExpression="\d+|\d"></asp:RegularExpressionValidator>
        <ajaxToolkit:ValidatorCalloutExtender ID="ValidatorCalloutExtenderPCumplimiento"
runat="server" PopupPosition="BottomLeft"
TargetControlID="RegularExpressionValidatorPCumplimiento"
WarningIconImageUrl="~/Imgs/Advertencia.png">
        </ajaxToolkit:ValidatorCalloutExtender>
    </td>
</tr>
<tr>
    <td class="formTitle">Norma:
    </td>
    <td>
        <asp:DropDownList ID="DropDownListNorma" runat="server" AutoPostBack="false"
DataTextField="Nombre" DataValueField="Id" CssClass="DropDownListMedio">
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td class="formTitle">Control al que pertenece:
    </td>
    <td>
        <asp:DropDownList ID="DropDownListControles" runat="server"
AutoPostBack="false"
DataTextField="Nombre" DataValueField="Id" CssClass="DropDownListMedio">
        </asp:DropDownList>
    </td>
</tr>
</table>

</fieldset>
</td>
</tr>

<tr>
    <td style="text-align: right">
        <asp:ImageButton ID="ImageButtonActualizar" runat="server"
ImageUrl="~/Imgs/Actualizar.png" OnClick="ImageButtonActualizar_Click" ToolTip="Actualizar Usuario"
ValidationGroup="validar" />
        <asp:ImageButton ID="ImageButtonCancelar" runat="server" ImageUrl="~/Imgs/Cancel.png"
OnClick="ImageButtonCancelar_Click" ToolTip="Cancelar" CausesValidation="False" />
    </td>

```

```

</tr>
</table>
</asp:Content>

```

Editar.aspx.cs

```

using ADCM = ADComun.Modelo;
using ADNucleo.Fachada;
using ADWeb.Constantes;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using ADComun.Modelo;
using ADWeb.App_Code;

/// <summary>
/// Clase para editar el control.
/// </summary>
public partial class CRUD_Control_Editar : System.Web.UI.Page
{
    /// <summary>
    /// Lógica para que cargue los datos cuando ingrese a la página.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            long ControlId = long.Parse(Request.QueryString[Constantes.PARAMETRO_CONTROL_ID]);

            ADCM.Control Control;
            Control = FachadaAD.Instancia.LeerControl(ControlId);

            DropDownListNorma.DataSource = FachadaAD.Instancia.LeerTodasNormas();
            DropDownListNorma.DataBind();

            DropDownListControles.DataSource =
            FachadaAD.Instancia.BuscarControlesPorNivel(Control.Nivel);
            DropDownListControles.DataBind();
            DropDownListControles.Items.Insert(0, new ListItem("", ""));

            TextBoxNombre.Text = Control.Nombre;
            CheckBoxAplicable.Checked = Control.Aplicable;
            CheckBoxImplementada.Checked = Control.Implementada;
            DropDownListNivel.SelectedValue = Control.Nivel.ToString();
            TextBoxDescripcion.Text = Control.Descripcion;
            TextBoxPCumplimiento.Text = Control.PCumplimiento.ToString();

```

```

        DropDownListNorma.SelectedValue = Control.Norma.Id.ToString();
        if (Control.Padres.Count != 0)
        {
            DropDownListControles.SelectedValue = Control.Padres.First().Id.ToString();
        }
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón Actualizar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonActualizar_Click(object sender, ImageClickEventArgs e)
{
    ADCM.Control control;
    long controlId = long.Parse(Request.QueryString[Constantes.PARAMETRO_CONTROL_ID]);

    control = FachadaAD.Instancia.LeerControl(controlId);

    control.Nombre = TextBoxNombre.Text;
    control.Aplicable = CheckBoxAplicable.Checked;
    control.Implementada = CheckBoxImplementada.Checked;
    control.Nivel = int.Parse(DropDownListNivel.SelectedValue);
    control.Descripcion = TextBoxDescripcion.Text;

    Norma norma =
    FachadaAD.Instancia.LeerNorma(long.Parse(DropDownListNorma.SelectedValue));
    control.Norma = norma;

    if (DropDownListControles.SelectedValue.Length != 0)
    {
        ADCM.Control padre =
        FachadaAD.Instancia.LeerControl(long.Parse(DropDownListControles.SelectedValue));
        control.Padres.Add(padre);
    }

    try
    {
        FachadaAD.Instancia.ActualizarControl(control);
        //GuardarCasos(ControlId);
        Master.MostrarMensaje("Operación exitosa", "El Control se actualizó correctamente.",
        Icono.Informacion, Constantes.URL_GESTION_CONTROLES);
    }
    //catch (ControlConNombreIngresoRepetidoExcepcion)
    //{
    //    Master.MostrarMensaje("Error", "Ya existe un Control con el nombre de ingreso " +
    Control.NombreIngreso, Icono.Error, Constantes.URL_GESTION_CCONTROLS);
    //}
    catch (Exception ex)
    {
        Master.MostrarMensaje("Error", "Error actualizando Control - " + ex.Message, Icono.Error,
        Constantes.URL_GESTION_CONTROLES);
    }
}

```

```

}
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón cancelar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCancelar_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect(Constants.URL_GESTION_CONTROLES, true);
}

/// <summary>
/// Evento que se dispara al cambiar el valor del dropdownlist.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void DropDownListNivel_SelectedIndexChanged(object sender, EventArgs e)
{
    DropDownListControles.DataSource =
    FachadaAD.Instancia.BuscarControlesPorNivel(int.Parse(DropDownListNivel.SelectedValue));
    DropDownListControles.DataBind();
}
}
}

```

```

Gestion.aspx
<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Gestion.aspx.cs" Inherits="CRUD_Control_Gestion" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
    <table style="width: 80%;" border="0" cellpadding="0" cellspacing="0" align="center">
        <tr>
            <td class="style1">
                <h1>Gestión de Controles</h1>
            </td>
        </tr>
        <tr>
            <td>
                <asp:UpdatePanel ID="UpdatePanelFiltro" runat="server">
                    <ContentTemplate>
                        <fieldset>
                            <legend>Filtro</legend>
                        </fieldset>
                    </ContentTemplate>
                </asp:UpdatePanel>
                <table cellpadding="0" cellspacing="0" border="0">
                    <tr>
                        <td class="formTitle">Nombre:
                    </td>
                </table>
            </td>
        </tr>
    </table>

```

```

        <td>&nbsp;</td>
        <asp:TextBox ID="TextBoxNombre" runat="server" CssClass="t"></asp:TextBox>
    </td>
    <td rowspan="2">&nbsp;</td>
    <asp:ImageButton ID="ImageButtonFiltrar" runat="server"
ImageUrl="~/Imgs/Filtro.png" ToolTip="Filtrar" OnClick="ImageButtonFiltrar_Click" />
    </td>
</tr>
</table>
</fieldset>
</ContentTemplate>
</asp:UpdatePanel>
</td>
</tr>

<tr>
<td style="text-align: right">
    <asp:ImageButton ID="ImageButtonCrear" runat="server" ImageUrl="~/Imgs/Crear.png"
ToolTip="Crear" OnClick="ImageButtonCrear_Click" />
</td>
</tr>

<tr>

<td style="text-align: center">
    <asp:UpdatePanel ID="UpdatePanelGridView" runat="server">
        <ContentTemplate>
            <fieldset>
                <legend>Controles
            </legend>

<asp:GridView ID="GridViewControles" runat="server" AutoGenerateColumns="False"
CellPadding="4" DataKeyNames="Id" ForeColor="#333333" GridLines="None"
HorizontalAlign="Center" OnRowCommand="GridViewControles_RowCommand"
EmptyDataText="No hay Controles registrados" Width="100%">
<AlternatingRowStyle BackColor="White" ForeColor="#666666" />
<Columns>
    <asp:BoundField DataField="Nombre" HeaderText="Nombre">
        <ItemStyle HorizontalAlign="Left" />
    </asp:BoundField>
    <asp:BoundField DataField="Norma.Nombre" HeaderText="Norma" />
    <asp:BoundField DataField="nivel" HeaderText="Nivel" />
    <!--<asp:BoundField DataField="Descripcion" HeaderText="Descripción" />-->

    <asp:TemplateField HeaderText="Aplicable">
        <ItemTemplate>
            <asp:Image ID="ImageAplicable" runat="server" ImageUrl="<%=#
ObtenerImagenAplicable((bool)Eval("Aplicable"))%>" />
        </ItemTemplate>
    </asp:TemplateField>

    <asp:BoundField DataField="PCumplimiento" HeaderText="Cumplimiento %">
        <ItemStyle HorizontalAlign="Center" />
    </asp:BoundField>

```

```

        <asp:TemplateField HeaderText="Opciones">
            <ItemStyle Width="100px" />
            <ItemTemplate>
                <asp:ImageButton ID="ImageButtonVer" runat="server"
ImageUrl='~/Imgs/Ver.png'
                ToolTip="Ver detalle" CommandName="ver"
                CommandArgument="<%# Container.DataItemIndex %>" />
                <asp:ImageButton ID="ImageButtonEditar" runat="server"
ImageUrl='~/Imgs/Editar.png' ToolTip="Editar" CommandName="editar" CommandArgument="<%#
Container.DataItemIndex %>" />
                <asp:ImageButton ID="ImageButtonEliminar" runat="server"
ImageUrl='~/Imgs/Delete.png' ToolTip="Eliminar" CommandName="eliminar" CommandArgument="<%#
Container.DataItemIndex %>" ValidationGroup="eliminar" />
            </ItemTemplate>
        </asp:TemplateField>
    </Columns>
    <EditRowStyle BackColor="#999999" />
    <FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
    <HeaderStyle BackColor="#000000" Font-Bold="True" ForeColor="White" />
    <PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />
    <RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
    <SelectedRowStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333"
/>
    <SortedAscendingCellStyle BackColor="#E9E7E2" />
    <SortedAscendingHeaderStyle BackColor="#506C8C" />
    <SortedDescendingCellStyle BackColor="#FFFDF8" />
    <SortedDescendingHeaderStyle BackColor="#6F8DAE" />
</asp:GridView>

</fieldset>
</ContentTemplate>
<Triggers>
    <asp:PostBackTrigger ControlID="GridViewControles" />
</Triggers>
</asp:UpdatePanel>
</td>

</tr>
<p>
    <asp:CustomValidator ID="CustomValidatorEliminar" runat="server"
        ClientValidationFunction="confirmarEliminacion"
        ValidationGroup="eliminar"></asp:CustomValidator>
</p>

</table>

</asp:Content>

```

Gestion.aspx.cs

```
using ADNucleo.Fachada;
using ADWeb.App_Code;
using ADWeb.Constantes;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using ADCM = ADComun.Modelo;

/// <summary>
/// Clase para la gestión de controles.
/// </summary>
public partial class CRUD_Control_Gestion : System.Web.UI.Page
{
    /// <summary>
    /// Page_load de la gestión.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            CargarDatos();
        }
    }

    /// <summary>
    /// Método para llenar los datos de la grilla.
    /// </summary>
    private void CargarDatos()
    {
        IList<ADCM.Control> _Controles = FachadaAD.Instancia.LeerTodosControles();

        GridViewControles.DataSource = _Controles;
        GridViewControles.DataBind();
    }

    /// <summary>
    /// Lógica para los botones de la grilla.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void GridViewControles_RowCommand(object sender, GridViewCommandEventArgs e)
    {
        int rowIndex = Convert.ToInt32(e.CommandArgument);
        long controlesId = (long)GridViewControles.DataKeys[rowIndex].Value;
        switch (e.CommandName)
        {
            case "ver":
                Server.Transfer(Constantes.URL_VER_CONTROL + "?" +
```

```

Constantes.PARAMETRO_CONTROL_ID + "=" + controlesId, false);
    break;
    case "editar":
        Server.Transfer(Constantes.URL_EDITAR_CONTROL + "?" +
Constantes.PARAMETRO_CONTROL_ID + "=" + controlesId, false);
        break;
    case "eliminar":
        controlesId = (long)GridViewControles.DataKeys[rowIndex].Value;
        try
        {
            FachadaAD.Instance.EliminarControl(controlesId);
        }
        catch (Exception ex)
        {
            Master.MostrarMensaje("Error", "Error eliminando el Control - " + ex.Message, Icono.Error,
null);
        }
        CargarDatos();
        break;
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón crear.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCrear_Click(object sender, ImageClickEventArgs e)
{
    Server.Transfer(Constantes.URL_CREAR_CONTROL);
}

/// <summary>
/// Obtiene la imagen y la carga en la grilla.
/// </summary>
/// <param name="_Aplicable">Valor activo.</param>
/// <returns>url de la imagen.</returns>
public static string ObtenerImagenAplicable(bool _Aplicable)
{
    if (_Aplicable)
    {
        return "~/Imgs/Activo.png";
    }
    else
    {
        return "~/Imgs/Inactivo.png";
    }
}

/// <summary>
/// Obtiene la imagen y la carga en la grilla.
/// </summary>
/// <param name="_Implementado">Valor activo.</param>
/// <returns>url de la imagen.</returns>

```



```

public static string ObtenerImagenImplementado(bool _Implementado)
{
    if (_Implementado)
    {
        return "~/Imgs/Activo.png";
    }
    else
    {
        return "~/Imgs/Inactivo.png";
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón filtrar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonFiltrar_Click(object sender, ImageClickEventArgs e)
{
    CargarDatos();
}
}

```

```

Ver.aspx
<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Ver.aspx.cs" Inherits="CRUD_Control_Ver" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">

<table style="width: 80%; border="0" cellpadding="0" cellspacing="0" align="center">
<tr>
<td class="style1">
<h1>Ver Control</h1>
</td>
</tr>
<tr>
<td>
<fieldset>
<legend>Datos del control
</legend>
<table>
<tr>
<td class="formTitle">Nombre:
</td>
<td>
<asp:TextBox ID="TextBoxNombre" runat="server" Width="400px" MaxLength="50"
ReadOnly="True"></asp:TextBox>
</td>
</tr>
</table>
</td>
</tr>
</table>

```

```

<td class="formTitle">Aplicable:
</td>
<td style="margin-left: 40px">
<asp:CheckBox ID="CheckBoxAplicable" runat="server" Width="100px"
MaxLength="5" Enabled="False"></asp:CheckBox>
</td>
</tr>
<tr>
<td class="formTitle">Implementada:
</td>
<td style="margin-left: 40px">
<asp:CheckBox ID="CheckBoxImplementada" runat="server" Width="100px"
Enabled="False"></asp:CheckBox>
</td>
</tr>
<tr>
<td class="formTitle">Nivel:
</td>
<td style="margin-left: 40px">
<asp:TextBox ID="TextBoxNivel" runat="server" Width="200px"
ReadOnly="True"></asp:TextBox>
</td>
</tr>
<tr>
<td class="formTitle">Descripción:
</td>
<td>
<asp:TextBox ID="TextBoxDescripcion" runat="server" Width="400px"
TextMode="MultiLine" ReadOnly="True"></asp:TextBox>
</td>
</tr>
<tr>
<td class="formTitle">Porcentaje cumplimiento:
</td>
<td>
<asp:TextBox ID="TextBoxPCumplimiento" runat="server" Width="100"
ReadOnly="True"></asp:TextBox>
</td>
</tr>
<tr>
<td class="formTitle">Norma:
</td>
<td>
<asp:TextBox ID="TextBoxNorma" runat="server" Width="400px"
TextMode="SingleLine" ReadOnly="True">
</asp:TextBox>
</td>
</tr>
<tr>
<td class="formTitle">Control al que pertenece:
</td>
<td>
<asp:TextBox ID="TextBoxControl" runat="server" Width="400px"
TextMode="SingleLine" ReadOnly="True">

```



```

        TextBoxDescripcion.Text = Control.Descripcion;
        if (Control.Padres.Count != 0)
        {
            TextBoxControl.Text = Control.Padres.First().Nombre;
        }
        TextBoxPCumplimiento.Text = Control.PCumplimiento.ToString();

        TextBoxNorma.Text = Control.Norma.Nombre.ToString();
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón cancelar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCancelar_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect(Constants.URL_GESTION_CONTROLES, true);
}
}
}

```

```

Crear.aspx
<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Crear.aspx.cs" Inherits="CRUD_Norma_Crear" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>

<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
    <table style="width: 80%;" border="0" cellpadding="0" cellspacing="0" align="center">
        <tr>
            <td class="style1">
                <h1>Crear Norma</h1>
            </td>
        </tr>
        <tr>
            <td>
                <fieldset>
                    <legend>Datos de la norma</legend>
                    <table>
                        <tr>
                            <td class="formTitle">Nombre:</td>
                            <td>
                                <asp:TextBox ID="TextBoxNombre" runat="server" Width="400px"
                                MaxLength="50"></asp:TextBox>
                                <asp:RequiredFieldValidator ID="RequiredFieldValidatorNombre" runat="server"

```

```

ValidationGroup="validar"
ControlToValidate="TextBoxNombre" ErrorMessage="Campo Obligatorio"
CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
<asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderNombre" runat="server"
PopupPosition="BottomLeft"
TargetControlID="RequiredFieldValidatorNombre"
WarningIconImageUrl="~/Imgs/Advertencia.png">
</asp:ValidatorCalloutExtender>
</td>
</tr>
<tr>
<td class="formTitle">Versión:
</td>
<td style="margin-left: 40px">
<asp:TextBox ID="TextBoxVersion" runat="server" Width="200px"
MaxLength="5"></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidatorVersion" runat="server"
ValidationGroup="validar"
ControlToValidate="TextBoxVersion" ErrorMessage="Campo Obligatorio"
CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
<asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderVersion" runat="server"
TargetControlID="RequiredFieldValidatorVersion"
WarningIconImageUrl="~/Imgs/Advertencia.png">
</asp:ValidatorCalloutExtender>
</td>
</tr>
<tr>
<td class="formTitle">Año:
</td>
<td>
<asp:TextBox ID="TextBoxAño" runat="server" Width="100px"
MaxLength="5"></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidatorAño" runat="server"
ValidationGroup="validar"
ControlToValidate="TextBoxAño" ErrorMessage="Campo Obligatorio"
CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
<asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderAño" runat="server"
TargetControlID="RequiredFieldValidatorAño"
WarningIconImageUrl="~/Imgs/Advertencia.png">
</asp:ValidatorCalloutExtender>
</td>
</tr>
<tr>
<td class="formTitle">Original:
</td>
<td>
<asp:TextBox ID="TextBoxOriginal" runat="server" Width="100px"
MaxLength="5"></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidatorOriginal" runat="server"
ValidationGroup="validar"
ControlToValidate="TextBoxOriginal" ErrorMessage="Campo Obligatorio"
CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
<asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderOriginal" runat="server"

```

```

        TargetControlID="RequiredFieldValidatorOriginal"
WarningIconImageUrl="~/Imgs/Advertencia.png">
    </asp:ValidatorCalloutExtender>
    </td>
</tr>
<tr>
<td class="formTitle">Descripción:
    </td>
    <td>
        <asp:TextBox ID="TextBoxDescripcion" runat="server" Width="400px"
        TextMode="MultiLine"></asp:TextBox>
        <asp:RequiredFieldValidator ID="RequiredFieldValidatorDescripcion" runat="server"
        ValidationGroup="validar"
            ControlToValidate="TextBoxDescripcion" ErrorMessage="Campo Obligatorio"
            CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
        <asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderDescripcion"
        runat="server" PopupPosition="TopLeft"
            TargetControlID="RequiredFieldValidatorDescripcion"
        WarningIconImageUrl="~/Imgs/Advertencia.png">
            </asp:ValidatorCalloutExtender>
        </td>
    </tr>
</table>

</fieldset>
</td>
</tr>

<tr>
<td style="text-align: right">
    <asp:ImageButton ID="ImageButtonCrear" runat="server" ImageUrl="~/Imgs/Ok.png"
    OnClick="ImageButtonCrear_Click" ToolTip="Crear" ValidationGroup="validar" />
    <asp:ImageButton ID="ImageButtonCancelar" runat="server" ImageUrl="~/Imgs/Cancel.png"
    OnClick="ImageButtonCancelar_Click" ToolTip="Cancelar" CausesValidation="False" />
    </td>
</tr>

</table>
</asp:Content>

```

Crear.aspx.cs

```
using ADComun.Modelo;
using ADNucleo.Fachada;
using ADComun.Modelo;
using ADWeb.App_Code;
using ADWeb.Constantes;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

/// <summary>
/// Clase para crear la norma.
/// </summary>
public partial class CRUD_Norma_Crear : System.Web.UI.Page
{
    /// <summary>
    /// Lógica para que cargue los datos cuando ingrese a la página.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    /// <summary>
    /// Evento que se dispara al dar clic sobre el botón crear.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void ImageButtonCrear_Click(object sender, ImageClickEventArgs e)
    {
        Norma norma = new Norma();

        norma.Nombre = TextBoxNombre.Text;
        norma.Version = TextBoxVersion.Text;
        norma.Ano = TextBoxAno.Text;
        norma.Original = TextBoxOriginal.Text;
        norma.Descripcion = TextBoxDescripcion.Text;

        try
        {
            FachadaAD.Instancia.AgregarNorma(norma);
            Master.MostrarMensaje("Operación exitosa", "La norma se creó correctamente.",
            Icono.Informacion, Constantes.URL_GESTION_NORMAS);
        }
        catch (Exception ex)
        {
            Master.MostrarMensaje("Error", "Error creando norma - " + ex.Message, Icono.Error,
            Constantes.URL_GESTION_NORMAS);
        }
    }
}
```

```

}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón cancelar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCancelar_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect(Constants.URL_GESTION_NORMAS, true);
}
}

```

```

Editar.aspx
<%@ Page Language="C#" MasterPageFile="~/Sitio.master" AutoEventWireup="true"
CodeFile="Editar.aspx.cs" Inherits="CRUD_Norma_Editar" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>
<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
<table style="width: 80%;" border="0" cellpadding="0" cellspacing="0" align="center">
<tr>
<td class="style1">
<h1>Editar Norma</h1>
</td>
</tr>
<tr>
<td>
<fieldset>
<legend>Datos de la norma</legend>
</fieldset>
</td>
</tr>
<tr>
<td class="formTitle">Nombre:
</td>
<td>
<asp:TextBox ID="TextBoxNombre" runat="server" Width="400px"
MaxLength="50"></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidatorNombre" runat="server"
ValidationGroup="validar"
ControlToValidate="TextBoxNombre" ErrorMessage="Campo Obligatorio"
CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
<asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderNombre" runat="server"
PopupPosition="BottomLeft"
TargetControlID="RequiredFieldValidatorNombre"
WarningIconImageUrl="~/Imgs/Advertencia.png">
</asp:ValidatorCalloutExtender>
</td>
</tr>
</tr>
</table>

```



```

<td class="formTitle">Versión:
</td>
<td style="margin-left: 40px">
<asp:TextBox ID="TextBoxVersion" runat="server" Width="200px"
MaxLength="5"></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidatorVersion" runat="server"
ValidationGroup="validar"
ControlToValidate="TextBoxVersion" ErrorMessage="Campo Obligatorio"
CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
<asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderVersion" runat="server"
TargetControlID="RequiredFieldValidatorVersion"
WarningIconImageUrl="~/Imgs/Advertencia.png">
</asp:ValidatorCalloutExtender>
</td>
</tr>
<tr>
<td class="formTitle">Año:
</td>
<td>
<asp:TextBox ID="TextBoxAño" runat="server" Width="100px"
MaxLength="5"></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidatorAño" runat="server"
ValidationGroup="validar"
ControlToValidate="TextBoxAño" ErrorMessage="Campo Obligatorio"
CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
<asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderAño" runat="server"
TargetControlID="RequiredFieldValidatorAño"
WarningIconImageUrl="~/Imgs/Advertencia.png">
</asp:ValidatorCalloutExtender>
</td>
</tr>
<tr>
<td class="formTitle">Porcentaje cumplimiento:
</td>
<td>
<asp:TextBox ID="TextBoxPCumplimiento" runat="server" Width="100"
ReadOnly="True"></asp:TextBox>
<asp:RegularExpressionValidator ID="RegularExpressionValidatorPCumplimiento"
runat="server" ControlToValidate="TextBoxPCumplimiento"
ErrorMessage="Campo debe ser número" CssClass="validador" Font-Bold="True"
Font-Size="0pt" ValidationGroup="validar"
ValidationExpression="\d+|\d"></asp:RegularExpressionValidator>
<ajaxToolkit:ValidatorCalloutExtender ID="ValidatorCalloutExtenderPCumplimiento"
runat="server" PopupPosition="BottomLeft"
TargetControlID="RegularExpressionValidatorPCumplimiento"
WarningIconImageUrl="~/Imgs/Advertencia.png">
</ajaxToolkit:ValidatorCalloutExtender>
</td>
</tr>
<tr>
<td class="formTitle">Original:
</td>
<td>

```

```

        <asp:TextBox ID="TextBoxOriginal" runat="server" Width="100px"
MaxLength="5"></asp:TextBox>
        <asp:RequiredFieldValidator ID="RequiredFieldValidatorOriginal" runat="server"
ValidationGroup="validar"
ControlToValidate="TextBoxOriginal" ErrorMessage="Campo Obligatorio"
CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
        <asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderOriginal" runat="server"
TargetControlID="RequiredFieldValidatorOriginal"
WarningIconImageUrl="~/Imgs/Advertencia.png">
</asp:ValidatorCalloutExtender>
</td>
</tr>
<tr>
<td class="formTitle">Descripción:
</td>
<td>
        <asp:TextBox ID="TextBoxDescripcion" runat="server" Width="400px"
TextMode="MultiLine"></asp:TextBox>
        <asp:RequiredFieldValidator ID="RequiredFieldValidatorDescripcion" runat="server"
ValidationGroup="validar"
ControlToValidate="TextBoxDescripcion" ErrorMessage="Campo Obligatorio"
CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
        <asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderDescripcion"
runat="server" PopupPosition="TopLeft"
TargetControlID="RequiredFieldValidatorDescripcion"
WarningIconImageUrl="~/Imgs/Advertencia.png">
</asp:ValidatorCalloutExtender>
</td>
</tr>
</table>

</fieldset>
</td>
</tr>

<tr>
<td style="text-align: right">
        <asp:ImageButton ID="ImageButtonActualizar" runat="server"
ImageUrl="~/Imgs/Actualizar.png" OnClick="ImageButtonActualizar_Click" ToolTip="Actualizar Usuario"
ValidationGroup="validar" />
        <asp:ImageButton ID="ImageButtonCancelar" runat="server" ImageUrl="~/Imgs/Cancel.png"
OnClick="ImageButtonCancelar_Click" ToolTip="Cancelar" CausesValidation="False" />
</td>
</tr>

</table>

</asp:Content>

```

Editar.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using ADNucleo.Fachada;
using ADComun.Modelo;
using ADComun.Excepciones;
using ADWeb.Constantes;
//using ADComun.Utils.JavaScript;
using System.Globalization;
using ADComun.Enums;
using ADWeb.App_Code;
using ADComun.Modelo;

/// <summary>
/// Clase para editar la norma.
/// </summary>
public partial class CRUD_Norma_Editar : System.Web.UI.Page
{
    /// <summary>
    /// Lógica para que cargue los datos cuando ingrese a la página.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            long normald = long.Parse(Request.QueryString[Constantes.PARAMETRO_NORMA_ID]);

            Norma norma;
            norma = FachadaAD.Instancia.LeerNorma(normald);
            TextBoxNombre.Text = norma.Nombre;
            TextBoxVersion.Text = norma.Version;
            TextBoxAno.Text = norma.Ano;
            TextBoxOriginal.Text = norma.Original;
            TextBoxDescripcion.Text = norma.Descripcion;
            TextBoxPCumplimiento.Text = norma.PCumplimiento.ToString();
        }
    }

    /// <summary>
    /// Evento que se dispara al dar clic sobre el botón Actualizar.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void ImageButtonActualizar_Click(object sender, ImageClickEventArgs e)
    {
        Norma norma;
        long normald = long.Parse(Request.QueryString[Constantes.PARAMETRO_NORMA_ID]);
    }
}
```

```

norma = FachadaAD.Instancia.LeerNorma(normald);

norma.Nombre = TextBoxNombre.Text;
norma.Version = TextBoxVersion.Text;
norma.Ano = TextBoxAno.Text;
norma.Original = TextBoxOriginal.Text;
norma.Descripcion = TextBoxDescripcion.Text;

try
{
    FachadaAD.Instancia.ActualizarNorma(norma);
    //GuardarCasos(normald);
    Master.MostrarMensaje("Operación exitosa", "La norma se actualizó correctamente.",
Icono.Informacion, Constantes.URL_GESTION_NORMAS);
}
//catch (NormaConNombreIngresoRepetidoExcepcion)
//{
//    Master.MostrarMensaje("Error", "Ya existe un norma con el nombre de ingreso " +
norma.NombreIngreso, Icono.Error, Constantes.URL_GESTION_NORMAS);
//}
catch (Exception ex)
{
    Master.MostrarMensaje("Error", "Error actualizando norma - " + ex.Message, Icono.Error,
Constantes.URL_GESTION_NORMAS);
}
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón cancelar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCancelar_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect(Constantes.URL_GESTION_NORMAS, true);
}
}

```

Gestion.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Gestion.aspx.cs" Inherits="CRUD_Norma_Gestion" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
    <div id="gestionNorma" style="padding-left: 5%; padding-right: 5%;">
        <div class="cabecera-1" style="width:100%">
            <div class="nombreCabecera" style="float:left; width:50%">
                <h1>Gestión de Normas</h1>
            </div>
            <div class="opcionesCabecera" style="float:left; text-align:right; width:50%">
                <br />
            </div>
        </div>
    </asp:Content>

```

```

        <asp:ImageButton ID="ImageButtonCrear" runat="server" ImageUrl="~/Imgs/Crear.png"
        ToolTip="Crear" OnClick="ImageButtonCrear_Click" />
    </div>
</div>
<div style="width:100%; float:right">
    <asp:UpdatePanel ID="UpdatePanelGridView" runat="server">
        <ContentTemplate>
            <fieldset>
                <legend>Normas
            </legend>

```

```

        <asp:GridView ID="GridViewNormas" runat="server" AutoGenerateColumns="False"
        CellPadding="4" DataKeyNames="Id" ForeColor="#333333" GridLines="None"
        HorizontalAlign="Center" OnRowCommand="GridViewNormas_RowCommand"
        EmptyDataText="No hay Normas registrados" Width="100%">
        <AlternatingRowStyle BackColor="White" ForeColor="#666666" />
        <Columns>
            <asp:TemplateField HeaderText="Norma">
                <ItemTemplate>
                    <asp:Image ID="ImageAplicable" runat="server"
                    ImageUrl='../..Imgs/Boceto/LogoNorma.png' />
                </ItemTemplate>
            </asp:TemplateField>

```

```

        <asp:BoundField DataField="Nombre" HeaderText="Nombre">
            <ItemStyle HorizontalAlign="Left" />
        </asp:BoundField>

```

```

        <asp:BoundField DataField="Version" HeaderText="Versión" />
        <asp:BoundField DataField="Año" HeaderText="Año" />
        <asp:BoundField DataField="Original" HeaderText="Original" />
        <asp:BoundField DataField="Descripcion" HeaderText="Descripción">
            <ItemStyle HorizontalAlign="Left" />
        </asp:BoundField>

```

```

        <asp:TemplateField HeaderText="Opciones">
            <ItemStyle Width="100px" />
            <ItemTemplate>
                <asp:ImageButton ID="ImageButtonVer" runat="server"
                ImageUrl='~/Imgs/Ver.png'
                ToolTip="Ver detalle" CommandName="ver"
                CommandArgument="<%# Container.DataItemIndex %>" />
                <asp:ImageButton ID="ImageButtonEditar" runat="server"
                ImageUrl='~/Imgs/Editar.png' ToolTip="Editar" CommandName="editar" CommandArgument="<%#
                Container.DataItemIndex %>" />
                <asp:ImageButton ID="ImageButtonEliminar" runat="server"
                ImageUrl='~/Imgs/Delete.png' ToolTip="Eliminar" CommandName="eliminar" CommandArgument="<%#
                Container.DataItemIndex %>" ValidationGroup="eliminar" />
            </ItemTemplate>
        </asp:TemplateField>
    </Columns>
    <EditRowStyle BackColor="#999999" />
    <FooterStyle BackColor="#F7F6F3" Font-Bold="True" ForeColor="White" />
    <HeaderStyle BackColor="#000000" Font-Bold="True" ForeColor="White" />

```

```

<PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />
<RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
<SelectedRowStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333"
/>
<SortedAscendingCellStyle BackColor="#E9E7E2" />
<SortedAscendingHeaderStyle BackColor="#506C8C" />
<SortedDescendingCellStyle BackColor="#FFFDF8" />
<SortedDescendingHeaderStyle BackColor="#6F8DAE" />
</asp:GridView>

</fieldset>
</ContentTemplate>
<Triggers>
<asp:PostBackTrigger ControlID="GridViewNormas" />
</Triggers>
</asp:UpdatePanel>
</div>
<div style="width:100%">
<asp:CustomValidator ID="CustomValidatorEliminar" runat="server"
ClientValidationFunction="confirmarEliminacion"
ValidationGroup="eliminar"></asp:CustomValidator>
</div>
</div>
</asp:Content>

```

Gestion.aspx.cs

```

using ADComun.Modelo;
using ADNucleo.Fachada;
using ADWeb.App_Code;
using ADWeb.Constantes;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

/// <summary>
/// Clase para la gestión de normas.
/// </summary>
public partial class CRUD_Norma_Gestion : System.Web.UI.Page
{
    /// <summary>
    /// Page_load de la gestión.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            CargarDatos();
        }
    }
}

```

```

}
}

/// <summary>
/// Método para llenar los datos de la grilla.
/// </summary>
private void CargarDatos()
{
    IList<Norma> _Normas = FachadaAD.Instancia.LeerTodasNormas();

    GridViewNormas.DataSource = _Normas;
    GridViewNormas.DataBind();
}

/// <summary>
/// Lógica para los botones de la grilla.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void GridViewNormas_RowCommand(object sender, GridViewCommandEventArgs e)
{
    int rowIndex = Convert.ToInt32(e.CommandArgument);
    long normasId = (long)GridViewNormas.DataKeys[rowIndex].Value;
    switch (e.CommandName)
    {
        case "ver":
            Server.Transfer(Constants.URL_VER_NORMA + "?" +
                Constants.PARAMETRO_NORMA_ID + "=" + normasId, false);
            break;
        case "editar":
            Server.Transfer(Constants.URL_EDITAR_NORMA + "?" +
                Constants.PARAMETRO_NORMA_ID + "=" + normasId, false);
            break;
        case "eliminar":
            normasId = (long)GridViewNormas.DataKeys[rowIndex].Value;
            try
            {
                FachadaAD.Instancia.EliminarNorma(normasId);
            }
            catch (Exception ex)
            {
                Master.MostrarMensaje("Error", "Error eliminando el Norma - " + ex.Message, Icono.Error,
                null);
            }
            CargarDatos();
            break;
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón crear.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>

```

```

protected void ImageButtonCrear_Click(object sender, ImageClickEventArgs e)
{
    Server.Transfer(Constants.URL_CREAR_NORMA);
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón filtrar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonFiltrar_Click(object sender, ImageClickEventArgs e)
{
    CargarDatos();
}
}
}

```

```

Ver.aspx
<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Ver.aspx.cs" Inherits="CRUD_Norma_Ver" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">

<table style="width: 90%;" border="0" cellpadding="0" cellspacing="0" align="center">
<tr>
<td class="style1">
<h1>Ver Norma</h1>
</td>
</tr>
<tr>
<td>
<fieldset>
<legend>Datos de la norma
</legend>
<div id="detallenorma">
<div id="detalle izquierda" style="width:45% ; float:left; text-align:right">
<div>
Nombre:
<asp:TextBox ID="TextBoxNombre" runat="server" MaxLength="50"
ReadOnly="True"></asp:TextBox>
</div>
<div>
Versión:
<asp:TextBox ID="TextBoxVersion" runat="server" MaxLength="5"
ReadOnly="True"></asp:TextBox>
</div>
<div>
Año:
<asp:TextBox ID="TextBoxAño" runat="server" MaxLength="5"
ReadOnly="True"></asp:TextBox>
</div>
</div>
</div>

```



```

Original:
<asp:TextBox ID="TextBoxOriginal" runat="server" MaxLength="5"
ReadOnly="True"></asp:TextBox>
</div>
<div>
Porcentaje cumplimiento:
<asp:TextBox ID="TextBoxPCumplimiento" runat="server"
ReadOnly="True"></asp:TextBox>
</div>
</div>
<div id="detalle derecha" style="float:left; width:50%; padding-left: 10px">
<div style="float:left; width:100%">
Descripción:
</div>

<div style="float:none; ">
<asp:TextBox ID="TextBoxDescripcion" runat="server" Width="100%" Height="100px"
TextMode="MultiLine" ReadOnly="True" style="margin-left: 0px"></asp:TextBox>
</div>

</div>

</div>

<div style="width: 100%; float:right">
<table>
<tr>

<td style="text-align: center; width: 200px;" rowspan="2">
<asp:UpdatePanel ID="UpdatePanelGridView" runat="server">
<ContentTemplate>
<fieldset>
<legend>Controles asociados a la norma
</legend>

<asp:GridView ID="GridViewControles" runat="server" AutoGenerateColumns="False"
CellPadding="4" DataKeyNames="Id" ForeColor="#333333" GridLines="None"
HorizontalAlign="Center"
EmptyDataText="No hay Controles registrados" Width="228%">
<AlternatingRowStyle BackColor="White" ForeColor="#666666" />
<Columns>
<asp:BoundField DataField="Nombre" HeaderText="Nombre">
<ItemStyle HorizontalAlign="Left" />
</asp:BoundField>
<asp:BoundField DataField="Norma.Nombre" HeaderText="Norma" />
<asp:BoundField DataField="nivel" HeaderText="Nivel" />
<!--<asp:BoundField DataField="Descripcion" HeaderText="Descripción" />-->

<asp:TemplateField HeaderText="Aplicable">
<ItemTemplate>
<asp:Image ID="ImageAplicable" runat="server" ImageUrl="<#%#
ObtenerImagenAplicable((bool)Eval("Aplicable")) %>' />
</ItemTemplate>
</asp:TemplateField>

```


Ver.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using ADNucleo.Fachada;
using ADComun.Modelo;
using ADWeb.Constantes;
using System.Globalization;
using TOCC = ADComun.Constantes;
using ADComun.Modelo;
using ADCM = ADComun.Modelo;

/// <summary>
/// Clase para ver los datos de la norma.
/// </summary>
public partial class CRUD_Norma_Ver : System.Web.UI.Page
{
    /// <summary>
    /// Lógica para que cargue los datos cuando ingrese a la página.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            long normald = long.Parse(Request.QueryString[Constantes.PARAMETRO_NORMA_ID]);

            Norma norma;
            norma = FachadaAD.Instancia.LeerNorma(normald);
            TextBoxNombre.Text = norma.Nombre;
            TextBoxVersion.Text = norma.Version;
            TextBoxAño.Text = norma.Año;
            TextBoxOriginal.Text = norma.Original;
            TextBoxDescripcion.Text = norma.Descripcion;
            TextBoxPCumplimiento.Text = norma.PCumplimiento.ToString();

            IList<ADCM.Control> _Controles = FachadaAD.Instancia.BuscarControlesPorNorma(norma.Id);

            GridViewControles.DataSource = _Controles;
            GridViewControles.DataBind();
        }
    }

    /// <summary>
    /// Evento que se dispara al dar clic sobre el botón cancelar.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
}
```

```

protected void ImageButtonCancelar_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect(Constants.URL_GESTION_NORMAS, true);
}

/// <summary>
/// Obtiene la imagen dinámica de aplicabilidad.
/// </summary>
/// <param name="_Aplicable">estado activo.</param>
/// <returns></returns>
public static string ObtenerImagenAplicable(bool _Aplicable)
{
    if (_Aplicable)
    {
        return "~/Imgs/Activo.png";
    }
    else
    {
        return "~/Imgs/Inactivo.png";
    }
}

/// <summary>
/// Obtiene la imagen dinámica de implementación.
/// </summary>
/// <param name="_Implementado">estado activo.</param>
/// <returns></returns>
public static string ObtenerImagenImplementado(bool _Implementado)
{
    if (_Implementado)
    {
        return "~/Imgs/Activo.png";
    }
    else
    {
        return "~/Imgs/Inactivo.png";
    }
}
}

```

CRUD observación

```

Crear.aspx
<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Crear.aspx.cs" Inherits="CRUD_Observacion_Crear" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>
<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
<table style="width: 80%; border="0" cellpadding="0" cellspacing="0" align="center">
<tr>
<td class="style1">

```

```

        <h1>Crear Obsevación</h1>
    </td>
</tr>
<tr>
<td>
    <fieldset>
        <legend>Datos de la Obsevación
        </legend>
        <table>
            <tr>
                <td class="formTitle">Texto:
            </td>
            <td>
                <asp:TextBox ID="TextBoxTexto" runat="server" Width="400px" MaxLength="50"
                TextMode="MultiLine"></asp:TextBox>
                <asp:RequiredFieldValidator ID="RequiredFieldValidatorTexto" runat="server"
                ValidationGroup="validar"
                ControlToValidate="TextBoxTexto" ErrorMessage="Campo Obligatorio"
                CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
                <asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderTexto" runat="server"
                PopupPosition="BottomLeft"
                TargetControlID="RequiredFieldValidatorTexto"
                WarningIconImageUrl="~/Imgs/Advertencia.png">
                </asp:ValidatorCalloutExtender>
            </td>
        </tr>
            <tr>
                <td class="formTitle">Plan de Acción:
            </td>
            <td>
                <asp:DropDownList ID="DropDownListPlanDeAccion" runat="server"
                AutoPostBack="False"
                DataTextField="Codigo" DataValueField="Id" CssClass="DropDownListMedio">
                </asp:DropDownList>
            </td>
        </tr>
    </table>
    </fieldset>
</td>
</tr>
<tr>
<td style="text-align: right">
    <asp:ImageButton ID="ImageButtonCrear" runat="server" ImageUrl="~/Imgs/Ok.png"
    OnClick="ImageButtonCrear_Click" ToolTip="Crear" ValidationGroup="validar" />
    <asp:ImageButton ID="ImageButtonCancelar" runat="server" ImageUrl="~/Imgs/Cancel.png"
    OnClick="ImageButtonCancelar_Click" ToolTip="Cancelar" CausesValidation="False" />
</td>
</tr>

```

```
</table>  
</asp:Content>
```

Crear.aspx.cs

```
using ADComun.Modelo;  
using ADNucleo.Fachada;  
using ADComun.Modelo;  
using ADWeb.App_Code;  
using ADWeb.Constantes;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.Web.UI;  
using System.Web.UI.WebControls;  
  
/// <summary>  
/// Clase para crear el observación.  
/// </summary>  
public partial class CRUD_Observacion_Crear : System.Web.UI.Page  
{  
    /// <summary>  
    /// Lógica para que cargue los datos cuando ingrese a la página.  
    /// </summary>  
    /// <param name="sender"></param>  
    /// <param name="e"></param>  
    protected void Page_Load(object sender, EventArgs e)  
    {  
        if (!IsPostBack)  
        {  
            DropDownListPlanDeAccion.DataSource = FachadaAD.Instancia.LeerTodosPlanesDeAccion();  
            DropDownListPlanDeAccion.DataBind();  
        }  
    }  
  
    /// <summary>  
    /// Evento que se dispara al dar clic sobre el botón crear.  
    /// </summary>  
    /// <param name="sender"></param>  
    /// <param name="e"></param>  
    protected void ImageButtonCrear_Click(object sender, ImageClickEventArgs e)  
    {  
        Observacion observacion = new Observacion();  
  
        observacion.Texto = TextBoxTexto.Text;  
        PlanDeAccion planDeAccion =  
        FachadaAD.Instancia.LeerPlanDeAccion(long.Parse(DropDownListPlanDeAccion.SelectedValue));  
        observacion.PlanDeAccion = planDeAccion;  
  
        try  
        {
```

```

        FachadaAD.Instancia.AgregarObservacion(observacion);
        Master.MostrarMensaje("Operación exitosa", "La observación se creó correctamente.",
        Icono.Informacion, Constantes.URL_GESTION_OBSERVACIONES);
    }
    catch (Exception ex)
    {
        Master.MostrarMensaje("Error", "Error creando observación - " + ex.Message, Icono.Error,
        Constantes.URL_GESTION_OBSERVACIONES);
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón cancelar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCancelar_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect(Constantes.URL_GESTION_OBSERVACIONES, true);
}
}

```

Editar.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Editar.aspx.cs" Inherits="CRUD_Observacion_Editar" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>
<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
    <table style="width: 80%; border="0" cellpadding="0" cellspacing="0" align="center">
        <tr>
            <td class="style1">
                <h1>Ver Obsevación</h1>
            </td>
        </tr>
        <tr>
            <td>
                <fieldset>
                    <legend>Datos de la Obsevación</legend>
                    <table>
                        <tr>
                            <td class="formTitle">Texto:</td>
                            <td>
                                <asp:TextBox ID="TextBoxTexto" runat="server" Width="400px" MaxLength="50"
                                TextMode="MultiLine"></asp:TextBox>
                                <asp:RequiredFieldValidator ID="RequiredFieldValidatorTexto" runat="server"
                                ValidationGroup="validar"
                                ControlToValidate="TextBoxTexto" ErrorMessage="Campo Obligatorio"
                                CssClass="validador" Font-Bold="True" Font-Size="0pt"></asp:RequiredFieldValidator>
                                <asp:ValidatorCalloutExtender ID="ValidatorCalloutExtenderTexto" runat="server"
                                PopupPosition="BottomLeft"

```

```

        TargetControlID="RequiredFieldValidatorTexto"
WarningIconImageUrl="~/Imgs/Advertencia.png">
    </asp:ValidatorCalloutExtender>
    </td>
</tr>
<tr>
    <td class="formTitle">Plan de Acción:
    </td>
    <td>
        <asp:DropDownList ID="DropDownListPlanDeAccion" runat="server"
AutoPostBack="false"
        DataTextField="Codigo" DataValueField="Id" CssClass="DropDownListMedio">
    </asp:DropDownList>
    </td>
</tr>
</table>
</fieldset>
</td>
</tr>
<tr>
    <td style="text-align: right">
        <asp:ImageButton ID="ImageButtonActualizar" runat="server"
ImageUrl="~/Imgs/Actualizar.png" OnClick="ImageButtonActualizar_Click" ToolTip="Actualizar Usuario"
ValidationGroup="validar" />
        <asp:ImageButton ID="ImageButtonCancelar" runat="server" ImageUrl="~/Imgs/Cancel.png"
OnClick="ImageButtonCancelar_Click" ToolTip="Cancelar" CausesValidation="False" />
    </td>
</tr>
</table>
</asp:Content>

```

```

Editar.aspx.cs
using ADCM = ADComun.Modelo;
using ADNucleo.Fachada;
using ADWeb.Constantes;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using ADComun.Modelo;
using ADWeb.App_Code;

/// <summary>
/// Clase para editar la observación.
/// </summary>
public partial class CRUD_Observacion_Editar : System.Web.UI.Page
{
    /// <summary>
    /// Lógica para que cargue los datos cuando ingrese a la página.
    /// </summary>

```



```

/// <param name="sender"></param>
/// <param name="e"></param>
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        DropDownListPlanDeAccion.DataSource = FachadaAD.Instancia.LeerTodosPlanesDeAccion();
        DropDownListPlanDeAccion.DataBind();

        long ObservacionId =
        long.Parse(Request.QueryString[Constantes.PARAMETRO_OBSERVACION_ID]);

        Observacion Observacion;
        Observacion = FachadaAD.Instancia.LeerObservacion(ObservacionId);
        TextBoxTexto.Text = Observacion.Texto;

        DropDownListPlanDeAccion.SelectedValue = Observacion.PlanDeAccion.Id.ToString();
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón Actualizar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonActualizar_Click(object sender, ImageClickEventArgs e)
{
    Observacion observacion = new Observacion();

    observacion.Texto = TextBoxTexto.Text;
    PlanDeAccion planDeAccion =
    FachadaAD.Instancia.LeerPlanDeAccion(long.Parse(DropDownListPlanDeAccion.SelectedValue));
    observacion.PlanDeAccion = planDeAccion;

    try
    {
        FachadaAD.Instancia.ActualizarObservacion(observacion);
        //GuardarCasos(ObservacionId);
        Master.MostrarMensaje("Operación exitosa", "La observación se actualizó correctamente.",
        Icono.Informacion, Constantes.URL_GESTION_OBSERVACIONES);
    }
    catch (Exception ex)
    {
        Master.MostrarMensaje("Error", "Error actualizando observación - " + ex.Message, Icono.Error,
        Constantes.URL_GESTION_OBSERVACIONES);
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón cancelar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCancelar_Click(object sender, ImageClickEventArgs e)

```

```
{
    Response.Redirect(Constants.URL_GESTION_OBSERVACIONES, true);
}
```

```
Gestion.aspx
<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Gestion.aspx.cs" Inherits="CRUD_Observacion_Gestion" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
    <table style="width: 80%;" border="0" cellpadding="0" cellspacing="0" align="center">
        <tr>
            <td class="style1">
                <h1>Gestión de Observaciones</h1>
            </td>
        </tr>
        <tr>
            <td style="text-align: right">
                <asp:ImageButton ID="ImageButtonCrear" OnClick="ImageButtonCrear_Click" runat="server"
ImageUrl="~/Imgs/Crear.png" ToolTip="Crear" />
            </td>
        </tr>
        <tr>
            <td style="text-align: center">
                <asp:UpdatePanel ID="UpdatePanelGridView" runat="server">
                    <ContentTemplate>
                        <fieldset>
                            <legend>Observaciones</legend>
                            <asp:GridView ID="GridViewObservaciones" runat="server"
AutoGenerateColumns="False"
CellPadding="4" DataKeyNames="Id" ForeColor="#333333" GridLines="None"
HorizontalAlign="Center" OnRowCommand="GridViewObservacion_RowCommand"
EmptyDataText="No hay Observaciones registrados" Width="100%">
                            <AlternatingRowStyle BackColor="White" ForeColor="#666666" />
                            <Columns>
                                <asp:BoundField DataField="Texto" HeaderText="Texto">
                                    <ItemStyle HorizontalAlign="Left" />
                                </asp:BoundField>
                                <asp:BoundField DataField="PlanDeAccion.Codigo" HeaderText="Plan de
Acción" />
                                <asp:TemplateField HeaderText="Opciones">
                                    <ItemStyle Width="100px" />
                                    <ItemTemplate>
                                        <asp:ImageButton ID="ImageButtonVer" runat="server"
ImageUrl='~/Imgs/Ver.png'
                                        ToolTip="Ver detalle" CommandName="ver"
                                        CommandArgument="<%# Container.DataItemIndex %>" />
                                    </ItemTemplate>
                                </asp:TemplateField>
                            </Columns>
                            <EditRowStyle BackColor="#999999" />
                        </fieldset>
                    </ContentTemplate>
                </asp:UpdatePanel>
            </td>
        </tr>
    </table>
</asp:Content>
```

```

<FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
<HeaderStyle BackColor="#000000" Font-Bold="True" ForeColor="White" />
<PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />
<RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
<SelectedRowStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333"
/>
<SortedAscendingCellStyle BackColor="#E9E7E2" />
<SortedAscendingHeaderStyle BackColor="#506C8C" />
<SortedDescendingCellStyle BackColor="#FFFDF8" />
<SortedDescendingHeaderStyle BackColor="#6F8DAE" />
</asp:GridView>
</fieldset>
</ContentTemplate>
<Triggers>
<asp:PostBackTrigger ControlID="GridViewObservaciones" />
</Triggers>
</asp:UpdatePanel>
</td>
</tr>
<p>
<asp:CustomValidator ID="CustomValidatorEliminar" runat="server"
ClientValidationFunction="confirmarEliminacion"
ValidationGroup="eliminar"></asp:CustomValidator>
</p>
</table>
</asp:Content>

```

```

Gestion.aspx.cs
using ADComun.Modelo;
using ADNucleo.Fachada;
using ADComun.Modelo;
using ADWeb.App_Code;
using ADWeb.Constantes;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

/// <summary>
/// Clase para la gestión de observaciones.
/// </summary>
public partial class CRUD_Observacion_Gestion : System.Web.UI.Page
{
    /// <summary>
    /// Page_load de la gestión.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {

```

```

    if (!IsPostBack)
    {
        CargarDatos();
    }
}

/// <summary>
/// Método para llenar los datos de la grilla.
/// </summary>
private void CargarDatos()
{
    IList<Observacion> _Observacion = FachadaAD.Instancia.LeerTodasObservaciones();

    GridViewObservaciones.DataSource = _Observacion;
    GridViewObservaciones.DataBind();
}

/// <summary>
/// Lógica para los botones de la grilla.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void GridViewObservacion_RowCommand(object sender, GridViewCommandEventArgs e)
{
    int rowIndex = Convert.ToInt32(e.CommandArgument);
    long observacionId = (long)GridViewObservaciones.DataKeys[rowIndex].Value;

    switch (e.CommandName)
    {
        case "ver":
            Server.Transfer(Constants.URL_VER_OBSERVACION + "?" +
                Constants.PARAMETRO_OBSERVACION_ID + "=" + observacionId, false);
            break;
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón crear.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCrear_Click(object sender, ImageClickEventArgs e)
{
    Server.Transfer(Constants.URL_CREAR_OBSERVACION);
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón filtrar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonFiltrar_Click(object sender, ImageClickEventArgs e)
{
    CargarDatos();
}
}

```

Ver.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Ver.aspx.cs" Inherits="CRUD_Observacion_Ver" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">

<table style="width: 80%;" border="0" cellpadding="0" cellspacing="0" align="center">
<tr>
<td class="style1">
<h1>Ver Obsevación</h1>
</td>
</tr>
<tr>
<td>
<fieldset>
<legend>Datos de la Obsevación</legend>
<table>
<tr>
<td class="formTitle">Texto:
</td>
<td>
<asp:TextBox ID="TextBoxTexto" runat="server" Width="400px" MaxLength="50"
TextMode="SingleLine" ReadOnly="True"></asp:TextBox>
</td>
</tr>
<tr>
<td class="formTitle">Plan de Acción:
</td>
<td>
<asp:TextBox ID="TextBoxPlanDeAccion" runat="server" Width="400px"
MaxLength="50" TextMode="SingleLine" ReadOnly="True"></asp:TextBox>
</td>
</tr>
</table>
</fieldset>
</td>
</tr>
<tr>
<td style="text-align: right">&nbsp;&nbsp;&nbsp;&
<asp:ImageButton ID="ImageButtonCancelar" runat="server" ImageUrl="~/Imgs/Cancel.png"
OnClick="ImageButtonCancelar_Click" ToolTip="Cancelar" CausesValidation="False" />
</td>
</tr>
</table>
</asp:Content>
```

Ver.aspx.cs

```
using ADNucleo.Fachada;
using ADWeb.Constantes;
using System;
using System.Collections.Generic;
```

```

using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using ADComun.Modelo;

/// <summary>
/// Clase para ver los datos de la observación.
/// </summary>
public partial class CRUD_Observacion_Ver : System.Web.UI.Page
{
    /// <summary>
    /// Lógica para que cargue los datos cuando ingrese a la página.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            long ObservacionId =
long.Parse(Request.QueryString[Constantes.PARAMETRO_OBSERVACION_ID]);

            Observacion Observacion;
            Observacion = FachadaAD.Instancia.LeerObservacion(ObservacionId);
            TextBoxTexto.Text = Observacion.Texto;

            TextBoxPlanDeAccion.Text = Observacion.PlanDeAccion.Codigo;
        }
    }

    /// <summary>
    /// Evento que se dispara al dar clic sobre el botón cancelar.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void ImageButtonCancelar_Click(object sender, ImageClickEventArgs e)
    {
        Response.Redirect(Constantes.URL_GESTION_OBSERVACIONES, true);
    }
}

```

CRUD plan de acción

```
Crear.aspx
<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Crear.aspx.cs" Inherits="CRUD_PlanDeAccion_Crear" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>
<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
  <table style="width: 80%; border="0" cellpadding="0" cellspacing="0" align="center">
    <tr>
      <td class="style1">
        <h1>Crear plan de acción</h1>
      </td>
    </tr>
    <tr>
      <td>
        <fieldset>
          <legend>Datos del plan de acción</legend>
          <table>
            <tr>
              <td class="formTitle">Código:</td>
              <td>
                <asp:TextBox ID="TextBoxCodigo" runat="server" CssClass="textBoxLargo"
Width="400" TextMode="SingleLine"></asp:TextBox>
                <asp:RequiredFieldValidator ID="RequiredFieldValidatorCodigo" runat="server"
ControlToValidate="TextBoxCodigo"
ErrorMessage="Campo Obligatorio" CssClass="validador" Font-Bold="True" Font-
Size="0pt" ValidationGroup="validar">
                </asp:RequiredFieldValidator>
                <ajaxToolkit:ValidatorCalloutExtender ID="ValidatorCalloutExtenderCodigo"
runat="server" PopupPosition="BottomLeft"
TargetControlID="RequiredFieldValidatorCodigo"
WarningIconImageUrl="~/Imgs/Advertencia.png">
                </ajaxToolkit:ValidatorCalloutExtender>
              </td>
            </tr>
            <tr>
              <td class="formTitle">Actividad:</td>
              <td>
                <asp:TextBox ID="TextBoxActividad" runat="server" CssClass="textBoxLargo"
Width="400" ToolTip="Actividad a desarrollar." TextMode="MultiLine"></asp:TextBox>
                <asp:RequiredFieldValidator ID="RequiredFieldValidatorActividad" runat="server"
ControlToValidate="TextBoxActividad"
ErrorMessage="Campo Obligatorio" CssClass="validador" Font-Bold="True" Font-
Size="0pt" ValidationGroup="validar">
                </asp:RequiredFieldValidator>
                <ajaxToolkit:ValidatorCalloutExtender ID="ValidatorCalloutExtenderActividad"
runat="server" PopupPosition="BottomLeft"
TargetControlID="RequiredFieldValidatorActividad"
WarningIconImageUrl="~/Imgs/Advertencia.png">
              </td>
            </tr>
          </table>
        </fieldset>
      </td>
    </tr>
  </table>
</asp:Content>
```

```

</ajaxToolkit:ValidatorCalloutExtender>
</td>
</tr>
<tr>
<td class="formTitle">Fecha de cumplimiento:
</td>
<td>
<asp:Calendar ID="Calendar1" runat="server" CaptionAlign="Left"
BackColor="White" BorderColor="Black" DayNameFormat="Shortest" Font-Names="Times New Roman"
Font-Size="10pt" ForeColor="Black" Height="136px" NextPrevFormat="FullMonth" TitleFormat="Month"
Width="267px" CellPadding="1" SelectedDate="<%# DateTime.Today %>">
<DayHeaderStyle BackColor="#CCCCCC" Font-Bold="True" Font-Size="7pt"
ForeColor="#333333" Height="10pt" />
<DayStyle Width="14%" />
<NextPrevStyle Font-Size="8pt" ForeColor="White" />
<OtherMonthDayStyle ForeColor="#999999" />
<SelectedDayStyle BackColor="#f36f21" ForeColor="White" />
<SelectorStyle BackColor="#CCCCCC" Font-Bold="True" Font-Names="Verdana"
Font-Size="8pt" ForeColor="#333333" Width="1%" />
<TitleStyle BackColor="Black" Font-Bold="True" Font-Size="13pt"
ForeColor="White" Height="14pt" />
<TodayDayStyle BackColor="#CCCC99" ForeColor="Black" />
</asp:Calendar>
</td>
</tr>
<tr>
<td class="formTitle">Porcentaje cumplimiento:
</td>
<td>
<asp:TextBox ID="TextBoxPCumplimiento" runat="server"
Width="100"></asp:TextBox>
<asp:RegularExpressionValidator ID="RegularExpressionValidatorPCumplimiento"
runat="server" ControlToValidate="TextBoxPCumplimiento"
ErrorMessage="Campo debe ser número" CssClass="validador" Font-Bold="True"
Font-Size="0pt" ValidationGroup="validar"
ValidationExpression="\d+|\d"></asp:RegularExpressionValidator>
<ajaxToolkit:ValidatorCalloutExtender ID="ValidatorCalloutExtenderPCumplimiento"
runat="server" PopupPosition="BottomLeft"
TargetControlID="RegularExpressionValidatorPCumplimiento"
WarningIconImageUrl="~/Imgs/Advertencia.png">
</ajaxToolkit:ValidatorCalloutExtender>
</td>
</tr>
<tr>
<td class="formTitle">Norma:
</td>
<td>
<asp:DropDownList ID="DropDownListNorma" runat="server" AutoPostBack="true"
DataTextField="Nombre" DataValueField="Id" CssClass="DropDownListMedio"
OnSelectedIndexChanged="DropDownListNorma_SelectedIndexChanged" Width="300">
</asp:DropDownList>
</td>
</tr>
</tr>
</tr>

```



```

        <td class="formTitle">Control:
        </td>
        <td>
            <asp:DropDownList ID="DropDownListControl" runat="server" AutoPostBack="true"
                DataTextField="Nombre" DataValueField="Id" CssClass="DropDownListMedio"
Width="300">
            </asp:DropDownList>
        </td>
    </tr>
</table>
</fieldset>
</td>
</tr>
<tr>
<td style="text-align: right">
    <asp:ImageButton ID="ImageButtonCrear" runat="server" ImageUrl="~/Imgs/Ok.png"
OnClick="ImageButtonCrear_Click" ToolTip="Crear" ValidationGroup="validar" />
    <asp:ImageButton ID="ImageButtonCancelar" runat="server" ImageUrl="~/Imgs/Cancel.png"
OnClick="ImageButtonCancelar_Click" ToolTip="Cancelar" CausesValidation="False" />
</td>
</tr>
</table>
</asp:Content>

```

Crear.aspx.cs

```

using ADComun.Modelo;
using ADNucleo.Fachada;
using ADWeb.App_Code;
using ADWeb.Constantes;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

/// <summary>
/// Clase para crear el plan de acción.
/// </summary>
public partial class CRUD_PlanDeAccion_Crear : System.Web.UI.Page
{
    /// <summary>
    /// Lógica para que cargue los datos cuando ingrese a la página.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            DropDownListNorma.DataSource = FachadaAD.Instancia.LeerTodasNormas();
            DropDownListNorma.DataBind();
        }
    }
}

```

```

        DropDownListNorma.Items.Insert(0, new ListItem("", ""));
        FachadaAD.Instancia.CalcularPorcentajeControl();
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón crear.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCrear_Click(object sender, ImageClickEventArgs e)
{
    PlanDeAccion planDeAccion = new PlanDeAccion();

    long planDeAccionId = 0;

    planDeAccion.Actividad = TextBoxActividad.Text;
    planDeAccion.FechaCreacion = DateTime.Today;

    if (Calendar1.SelectedDate.ToString() != "01/01/0001 0:00:00")
    {
        planDeAccion.FechaCumplimiento = Calendar1.SelectedDate;
    }
    else
    {
        planDeAccion.FechaCumplimiento = DateTime.Today;
    }

    planDeAccion.Codigo = TextBoxCodigo.Text;
    planDeAccion.PCumplimiento = TextBoxPCumplimiento.Text;

    if (DropDownListControl.SelectedValue != "")
    {
        ADComun.Modelo.Control control =
        FachadaAD.Instancia.LeerControl(long.Parse(DropDownListControl.SelectedValue));

        try
        {
            control.PlanDeAccion.Add(planDeAccion);
            control.Implementada = true;
            planDeAccionId = FachadaAD.Instancia.AgregarPlanDeAccion(planDeAccion);
            Master.MostrarMensaje("Operación exitosa", "El PlanDeAccion se creó correctamente.",
            Icono.Informacion, Constantes.URL_GESTION_PLANESDEACCION);
        }
        catch (Exception ex)
        {
            Master.MostrarMensaje("Error", "Error creando el plan de acción - " + ex.Message,
            Icono.Error, Constantes.URL_GESTION_PLANESDEACCION);
        }
    }
    else
    {
        Master.MostrarMensaje("Error", "Debe seleccionar un control.", Icono.Error, null);
    }
}

```

```

}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón cancelar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCancelar_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect(Constants.URL_GESTION_PLANESDEACCION, true);
}

/// <summary>
/// Evento que se dispara al cambiar el valor del dropdownlist.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void DropDownListNorma_SelectedIndexChanged(object sender, EventArgs e)
{
    long normald;
    long.TryParse(DropDownListNorma.SelectedValue, out normald);

    //DropDownListControl.DataSource = FachadaAD.Instancia.BuscarControlesPorNorma(normald);
    IList<ADComun.Modelo.Control> _Controles =
    FachadaAD.Instancia.BuscarControlesPorNormaImplementado(normald, false);
    IList<ADComun.Modelo.Control> _ControlesSinHijos = new List<ADComun.Modelo.Control>();
    foreach(ADComun.Modelo.Control control in _Controles)
    {
        if(control.Controles.Count() > 0)
        {
            _ControlesSinHijos.Add(control);
        }
    }

    foreach (ADComun.Modelo.Control controlSinHijo in _ControlesSinHijos)
    {
        _Controles.Remove(controlSinHijo);
    }

    DropDownListControl.DataSource = _Controles;
    DropDownListControl.DataBind();
}
}

```

```

Editar.aspx
<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Editar.aspx.cs" Inherits="CRUD_PlanDeAccion_Editar" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>
<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
    <table style="width: 80%;" border="0" cellpadding="0" cellspacing="0" align="center">
    <tr>

```

```

<td class="style1">
  <h1>Editar plan de acción</h1>
</td>
</tr>
<tr>
<td>
  <fieldset>
    <legend>Datos del plan de acción</legend>
    <table>
      <tr>
        <td class="formTitle">Código:</td>
        <td>
          <asp:TextBox ID="TextBoxCodigo" runat="server" CssClass="textBoxLargo"
Width="400" TextMode="SingleLine"></asp:TextBox>
          <asp:RequiredFieldValidator ID="RequiredFieldValidatorCodigo" runat="server"
ControlToValidate="TextBoxCodigo"
ErrorMessage="Campo Obligatorio" CssClass="validador" Font-Bold="True" Font-
Size="0pt" ValidationGroup="validar">
            </asp:RequiredFieldValidator>
            <ajaxToolkit:ValidatorCalloutExtender ID="ValidatorCalloutExtenderCodigo"
runat="server" PopupPosition="BottomLeft"
TargetControlID="RequiredFieldValidatorCodigo"
WarningIconImageUrl="~/Imgs/Advertencia.png">
              </ajaxToolkit:ValidatorCalloutExtender>
            </td>
          </tr>
          <tr>
            <td class="formTitle">Actividad:</td>
            <td>
              <asp:TextBox ID="TextBoxActividad" runat="server" CssClass="textBoxLargo"
Width="400" ToolTip="Actividad a desarrollar." TextMode="MultiLine"></asp:TextBox>
              <asp:RequiredFieldValidator ID="RequiredFieldValidatorActividad" runat="server"
ControlToValidate="TextBoxActividad"
ErrorMessage="Campo Obligatorio" CssClass="validador" Font-Bold="True" Font-
Size="0pt" ValidationGroup="validar">
                </asp:RequiredFieldValidator>
                <ajaxToolkit:ValidatorCalloutExtender ID="ValidatorCalloutExtenderActividad"
runat="server" PopupPosition="BottomLeft"
TargetControlID="RequiredFieldValidatorActividad"
WarningIconImageUrl="~/Imgs/Advertencia.png">
                  </ajaxToolkit:ValidatorCalloutExtender>
                </td>
              </tr>
              <tr>
                <td class="formTitle">Fecha de cumplimiento:</td>
                <td>
                  <asp:TextBox ID="TextBoxFechaCumplimiento" runat="server"></asp:TextBox>
                </td>
              </tr>
            </tr>
          </table>
        </td>
      </tr>
    </table>
  </fieldset>
</td>
</tr>

```

```

        <td class="formTitle">Control:
        </td>
        <td>
            <asp:TextBox ID="TextBoxControl" runat="server"
ReadOnly="True"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td class="formTitle">Porcentaje cumplimiento:
        </td>
        <td>
            <asp:TextBox ID="TextBoxPCumplimiento" runat="server"
Width="100"></asp:TextBox>
            <asp:RegularExpressionValidator ID="RegularExpressionValidatorPCumplimiento"
runat="server" ControlToValidate="TextBoxPCumplimiento"
ErrorMessage="Campo Obligatorio" CssClass="validador" Font-Bold="True" Font-
Size="0pt" ValidationGroup="validar" ValidationExpression="\d+"></asp:RegularExpressionValidator>
            <ajaxToolkit:ValidatorCalloutExtender ID="ValidatorCalloutExtenderPCumplimiento"
runat="server" PopupPosition="BottomLeft"
TargetControlID="RegularExpressionValidatorPCumplimiento"
WarningIconImageUrl="~/Imgs/Advertencia.png">
            </ajaxToolkit:ValidatorCalloutExtender>
        </td>
    </tr>
</table>
</fieldset>
</td>
</tr>
<tr>
    <td style="text-align: right">
        <asp:ImageButton ID="ImageButtonActualizar" runat="server"
ImageUrl="~/Imgs/Actualizar.png" OnClick="ImageButtonActualizar_Click" ToolTip="Actualizar Usuario"
ValidationGroup="validar" />
        <asp:ImageButton ID="ImageButtonCancelar" runat="server" ImageUrl="~/Imgs/Cancel.png"
OnClick="ImageButtonCancelar_Click" ToolTip="Cancelar" CausesValidation="False" />
    </td>
</tr>
</table>
</asp:Content>

```

```

Editar.aspx.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using ADNucleo.Fachada;
using ADComun.Modelo;
using ADComun.Excepciones;
using ADWeb.Constantes;

```

```

using System.Globalization;
using ADComun.Enums;
using ADWeb.App_Code;
using ADComun.Modelo;

/// <summary>
/// Clase para editar el plan de acción.
/// </summary>
public partial class CRUD_PlanDeAccion_Editar : System.Web.UI.Page
{
    /// <summary>
    /// Lógica para que cargue los datos cuando ingrese a la página.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            long planDeAccionId =
long.Parse(Request.QueryString[Constantes.PARAMETRO_PLANDEACCION_ID]);
            PlanDeAccion planDeAccion;
            planDeAccion = FachadaAD.Instancia.LeerPlanDeAccion(planDeAccionId);
            TextBoxActividad.Text = planDeAccion.Actividad;
            TextBoxFechaCumplimiento.Text = planDeAccion.FechaCumplimiento.ToShortDateString();
            TextBoxCodigo.Text = planDeAccion.Codigo;
            TextBoxPCumplimiento.Text = planDeAccion.PCumplimiento;
            if (planDeAccion.Control.Count != 0)
            {
                TextBoxControl.Text = planDeAccion.Control.First().Nombre;
            }
        }
    }
    /// <summary>
    /// Evento que se dispara al dar clic sobre el botón Actualizar.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void ImageButtonActualizar_Click(object sender, ImageClickEventArgs e)
    {
        PlanDeAccion planDeAccion;
        long planDeAccionId =
long.Parse(Request.QueryString[Constantes.PARAMETRO_PLANDEACCION_ID]);
        planDeAccion = FachadaAD.Instancia.LeerPlanDeAccion(planDeAccionId);
        planDeAccion.Actividad = TextBoxActividad.Text;
        planDeAccion.FechaCreacion = DateTime.Today;
        planDeAccion.FechaCumplimiento = Convert.ToDateTime(TextBoxFechaCumplimiento.Text);
        planDeAccion.Codigo = TextBoxCodigo.Text;
        planDeAccion.PCumplimiento = TextBoxPCumplimiento.Text;
        if (TextBoxControl.Text != null)
        {
            planDeAccion.Control.First().PCumplimiento = int.Parse(TextBoxPCumplimiento.Text);
        }
    }
    try

```

```

        FachadaAD.Instancia.ActualizarPlanDeAccion(planDeAccion);
        Master.MostrarMensaje("Operación exitosa", "El plan de acción se actualizó correctamente.",
        Icono.Informacion, Constantes.URL_GESTION_PLANESDEACCION);
    }
    catch (Exception ex)
    {
        Master.MostrarMensaje("Error", "Error actualizando planDeAccion - " + ex.Message, Icono.Error,
        Constantes.URL_GESTION_PLANESDEACCION);
    }
}
/// <summary>
/// Evento que se dispara al dar clic sobre el botón cancelar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCancelar_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect(Constantes.URL_GESTION_PLANESDEACCION, true);
}
}
}

```

Gestion.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Gestion.aspx.cs" Inherits="CRUD_PlanDeAccion_Gestion" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
    <table style="width: 80%;" border="0" cellpadding="0" cellspacing="0" align="center">
        <tr>
            <td class="style1">
                <h1>Gestión de planes de acción</h1>
            </td>
            <td style="text-align: right">
                <asp:ImageButton ID="ImageButtonCrear" runat="server" ImageUrl="~/Imgs/Crear.png"
                ToolTip="Crear" OnClick="ImageButtonCrear_Click" />
            </td>
        </tr>
        <tr>
            <td style="text-align: center">
                <asp:UpdatePanel ID="UpdatePanelGridView" runat="server">
                    <ContentTemplate>
                        <fieldset>
                            <legend>Planes de acción</legend>
                            <asp:GridView ID="GridViewPlanesDeAccion" runat="server"
                            AutoGenerateColumns="False"
                            CellPadding="4" DataKeyNames="Id" ForeColor="#333333" GridLines="None"
                            HorizontalAlign="Center"
                            OnRowCommand="GridViewPlanesDeAccion_RowCommand"

```

```

EmptyDataText="No hay PlanesDeAccion registrados" Width="100%" />
<AlternatingRowStyle BackColor="White" ForeColor="#666666" />
<Columns>
  <asp:BoundField DataField="Codigo" HeaderText="Código">
    <ItemStyle HorizontalAlign="Left" />
  </asp:BoundField>
  <asp:BoundField DataField="Actividad" HeaderText="Actividad" />
  <asp:BoundField DataField="FechaCreacion" HeaderText="Fecha de creacion" />
  <asp:BoundField DataField="FechaCumplimiento" HeaderText="Fecha de
cumplimiento" />
  <asp:BoundField DataField="PCumplimiento" HeaderText="Cumplimiento %" />
  <asp:TemplateField HeaderText="Opciones">
    <ItemStyle Width="100px" />
    <ItemTemplate>
      <asp:ImageButton ID="ImageButtonVer" runat="server"
ImageUrl='~/Imgs/Ver.png'
      Tooltip="Ver detalle" CommandName="ver"
      CommandArgument="<%# Container.DataItemIndex %>" />
      <asp:ImageButton ID="ImageButtonEditar" runat="server"
ImageUrl='~/Imgs/Editar.png' Tooltip="Editar" CommandName="editar" CommandArgument="<%#
Container.DataItemIndex %>" />
      <asp:ImageButton ID="ImageButtonEliminar" runat="server"
ImageUrl='~/Imgs/Delete.png' Tooltip="Eliminar" CommandName="eliminar" CommandArgument="<%#
Container.DataItemIndex %>" ValidationGroup="eliminar" />
    </ItemTemplate>
  </asp:TemplateField>
</Columns>
<EditRowStyle BackColor="#999999" />
<FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
<HeaderStyle BackColor="#000000" Font-Bold="True" ForeColor="White" />
<PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />
<RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
<SelectedRowStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333"
/>
  <SortedAscendingCellStyle BackColor="#E9E7E2" />
  <SortedAscendingHeaderStyle BackColor="#506C8C" />
  <SortedDescendingCellStyle BackColor="#FFFDF8" />
  <SortedDescendingHeaderStyle BackColor="#6F8DAE" />
</asp:GridView>
</fieldset>
</ContentTemplate>
<Triggers>
  <asp:PostBackTrigger ControlID="GridViewPlanesDeAccion" />
</Triggers>
</asp:UpdatePanel>
</td>
</tr>
<p>
  <asp:CustomValidator ID="CustomValidatorEliminar" runat="server"
  ClientValidationFunction="confirmarEliminacion"
  ValidationGroup="eliminar"></asp:CustomValidator>
</p>
</table>
</asp:Content>

```


Gestion.aspx.cs

```
using ADComun.Modelo;
using ADNucleo.Fachada;
using ADWeb.App_Code;
using ADWeb.Constantes;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

/// <summary>
/// Clase para la gestión de planes de acción.
/// </summary>
public partial class CRUD_PlanDeAccion_Gestion : System.Web.UI.Page
{
    /// <summary>
    /// Page_load de la gestión.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            CargarDatos();
        }
    }

    /// <summary>
    /// Método para llenar los datos de la grilla.
    /// </summary>
    private void CargarDatos()
    {
        IList<PlanDeAccion> _PlanDeAccions = FachadaAD.Instancia.LeerTodosPlanesDeAccion();

        GridViewPlanesDeAccion.DataSource = _PlanDeAccions;
        GridViewPlanesDeAccion.DataBind();
    }

    /// <summary>
    /// Lógica para los botones de la grilla.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void GridViewPlanesDeAccion_RowCommand(object sender, GridViewCommandEventArgs e)
    {
        int rowIndex = Convert.ToInt32(e.CommandArgument);
        long planesDeAccionId = (long)GridViewPlanesDeAccion.DataKeys[rowIndex].Value;
        switch (e.CommandName)
        {

```

```

        case "ver":
            Server.Transfer(Constants.URL_VER_PLANDEACCION + "?" +
Constants.PARAMETRO_PLANDEACCION_ID + "=" + planesDeAccionId, false);
            break;
        case "editar":
            Server.Transfer(Constants.URL_EDITAR_PLANDEACCION + "?" +
Constants.PARAMETRO_PLANDEACCION_ID + "=" + planesDeAccionId, false);
            break;
        case "eliminar":
            planesDeAccionId = (long)GridViewPlanesDeAccion.DataKeys[rowIndex].Value;
            try
            {
                FachadaAD.Instancia.EliminarPlanDeAccion(planesDeAccionId);
            }
            catch (Exception ex)
            {
                Master.MostrarMensaje("Error", "Error eliminando el PlanDeAccion - " + ex.Message,
Icono.Error, null);
            }
            CargarDatos();
            break;
    }
}

```

```

/// <summary>
/// Evento que se dispara al dar clic sobre el botón crear.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCrear_Click(object sender, ImageClickEventArgs e)
{
    Server.Transfer(Constants.URL_CREAR_PLANDEACCION);
}

```

```

/// <summary>
/// Evento que se dispara al dar clic sobre el botón filtrar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonFiltrar_Click(object sender, ImageClickEventArgs e)
{
    CargarDatos();
}
}

```

Ver.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Sitio.Master" AutoEventWireup="true"
CodeFile="Ver.aspx.cs" Inherits="CRUD_PlanDeAccion_Ver"%>
<%@ MasterType VirtualPath="~/Sitio.master" %>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
    <table style="width: 60%; border="0" cellpadding="0" cellspacing="0" align="center">
        <tr>

```


Ver.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using ADNucleo.Fachada;
using ADComun.Modelo;
using ADComun.Excepciones;
using ADWeb.Constantes;
using System.Globalization;
using ADComun.Enums;
using ADWeb.App_Code;

/// <summary>
/// Clase para ver los datos de los planes de acción.
/// </summary>
public partial class CRUD_PlanDeAccion_Ver : System.Web.UI.Page
{
    /// <summary>
    /// Lógica para que cargue los datos cuando ingrese a la página.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            long planDeAccionId =
            long.Parse(Request.QueryString[Constantes.PARAMETRO_PLANDEACCION_ID]);

            PlanDeAccion planDeAccion;
            planDeAccion = FachadaAD.Instancia.LeerPlanDeAccion(planDeAccionId);
            TextBoxActividad.Text = planDeAccion.Actividad;
            TextBoxFechaCumplimiento.Text = planDeAccion.FechaCumplimiento.ToShortDateString();
            TextBoxCodigo.Text = planDeAccion.Codigo;
            TextBoxPCumplimiento.Text = planDeAccion.PCumplimiento;
        }
    }

    /// <summary>
    /// Evento que se dispara al dar clic sobre el botón cancelar.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void ImageButtonCancelar_Click(object sender, ImageClickEventArgs e)
    {
        Response.Redirect(Constantes.URL_GESTION_PLANESDEACCION, true);
    }
}
```

CRUD usuario

```
Crear.aspx
<%@ Page Language="C#" MasterPageFile="~/Sitio.master" AutoEventWireup="true"
CodeFile="Crear.aspx.cs" Inherits="CRUD_Usuario_Crear" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
  <table style="width: 80%; border="0" cellpadding="0" cellspacing="0" align="center">
    <tr>
      <td class="style1">
        <h1>Crear Usuario</h1>
      </td>
    </tr>
    <tr>
      <td>
        <fieldset>
          <legend>Datos del usuario</legend>
          <table>
            <tr>
              <td class="formTitle">Nombre de ingreso:</td>
              <td>
                <asp:TextBox ID="TextBoxNombreIngreso" runat="server" CssClass="textBoxLargo"
MaxLength="100" ToolTip="Nombre de ingreso del usuario, mínimo 5 caracteres."></asp:TextBox>
                <asp:RequiredFieldValidator ID="RequiredFieldValidatorNombreIngreso"
runat="server" ControlToValidate="TextBoxNombreIngreso"
ErrorMessage="Campo Obligatorio" CssClass="validador" Font-Bold="True" Font-
Size="0pt" ValidationGroup="validar">
                </asp:RequiredFieldValidator>
                <ajaxToolkit:ValidatorCalloutExtender ID="ValidatorCalloutExtenderNombreIngreso"
runat="server"
TargetControlID="RequiredFieldValidatorNombreIngreso"
WarningIconImageUrl="~/Imgs/Advertencia.png">
                </ajaxToolkit:ValidatorCalloutExtender>
              </td>
            </tr>
            <tr>
              <td class="formTitle">Clave:</td>
              <td style="margin-left: 40px">
                <asp:TextBox ID="TextBoxClave" runat="server" CssClass="textBoxLargo"
MaxLength="255" TextMode="Password" ToolTip="Clave o password, mínimo 5
caracteres."></asp:TextBox>
                <asp:RequiredFieldValidator ID="RequiredFieldValidatorClave" runat="server"
ControlToValidate="TextBoxClave"
ErrorMessage="Campo Obligatorio" CssClass="validador" Font-Bold="True" Font-
Size="0pt" ValidationGroup="validar"></asp:RequiredFieldValidator>
                <ajaxToolkit:ValidatorCalloutExtender ID="ValidatorCalloutExtenderClave"
runat="server"
TargetControlID="RequiredFieldValidatorClave"
WarningIconImageUrl="~/Imgs/Advertencia.png">
                </ajaxToolkit:ValidatorCalloutExtender>
              </td>
            </tr>
          </table>
        </fieldset>
      </td>
    </tr>
  </table>
</asp:Content>
```

```

</td>
</tr>
<tr>
<td class="formTitle">Confirmar clave:
</td>
<td>
<asp:TextBox ID="TextBoxConfirmarClave" runat="server"
CssClass="textBoxLargo" MaxLength="255" TextMode="Password" ToolTip="Confirmar la clave o
password, mínimo 5 caracteres."></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidatorConfirmarClave"
runat="server" ControlToValidate="TextBoxConfirmarClave"
ErrorMessage="Campo Obligatorio" CssClass="validador" Font-Bold="True" Font-
Size="0pt" ValidationGroup="validar">
</asp:RequiredFieldValidator>
<ajaxToolkit:ValidatorCalloutExtender ID="ValidatorCalloutExtenderConfirmarClave"
runat="server"
TargetControlID="RequiredFieldValidatorConfirmarClave"
WarningIconImageUrl="~/Imgs/Advertencia.png">
</ajaxToolkit:ValidatorCalloutExtender>
<asp:CompareValidator ID="CompararClaves" runat="server"
ControlToCompare="TextBoxClave" ControlToValidate="TextBoxConfirmarClave"
ErrorMessage="Claves diferentes." ValidationGroup="validar" Font-Size="0pt">
</asp:CompareValidator>
<ajaxToolkit:ValidatorCalloutExtender
ID="ValidatorCalloutExtenderCompararClaves" runat="server"
TargetControlID="CompararClaves"
WarningIconImageUrl="~/Imgs/Advertencia.png">
</ajaxToolkit:ValidatorCalloutExtender>
</td>
</tr>
<tr>
<td class="formTitle">Nombre:
</td>
<td>
<asp:TextBox ID="TextBoxNombre" runat="server" CssClass="textBoxLargo"
MaxLength="100" ToolTip="Nombre completo del usuario."></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidatorNombre" runat="server"
ControlToValidate="TextBoxNombre"
ErrorMessage="Campo Obligatorio" CssClass="validador" Font-Bold="True" Font-
Size="0pt" ValidationGroup="validar">
</asp:RequiredFieldValidator>
<ajaxToolkit:ValidatorCalloutExtender ID="ValidatorCalloutExtenderNombre"
runat="server"
TargetControlID="RequiredFieldValidatorNombre"
WarningIconImageUrl="~/Imgs/Advertencia.png">
</ajaxToolkit:ValidatorCalloutExtender>
</td>
</tr>
<tr>
<td class="formTitle">Rol:
</td>
<td>
<asp:DropDownList ID="DropDownListRoles" runat="server" AutoPostBack="false"
DataTextField="Nombre" DataValueField="Id" CssClass="DropDownListMedio">

```

```

        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td class="formTitle">Activo:
    </td>
    <td>
        <asp:CheckBox ID="CheckBoxActivo" runat="server" />
    </td>
</tr>
</table>
</fieldset>
</td>
</tr>
<tr>
    <td style="text-align: right">
        <asp:ImageButton ID="ImageButtonCrear" runat="server" ImageUrl="~/Imgs/Ok.png"
        OnClick="ImageButtonCrear_Click" ToolTip="Crear" ValidationGroup="validar" />
        <asp:ImageButton ID="ImageButtonCancelar" runat="server" ImageUrl="~/Imgs/Cancel.png"
        OnClick="ImageButtonCancelar_Click" ToolTip="Cancelar" CausesValidation="False" />
    </td>
</tr>
</table>
</asp:Content>

```

```

Crear.aspx.cs
using ADComun.Modelo;
using ADNucleo.Fachada;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using ADNucleo.Fachada;
using ADComun.Modelo;
using ADComun.Excepciones;
using ADWeb.Constantes;
using System.Globalization;
using ADComun.Enums;
using ADWeb.App_Code;

/// <summary>
/// Clase para crear el usuario.
/// </summary>
public partial class CRUD_Usuario_Crear : System.Web.UI.Page
{
    /// <summary>
    /// Lógica para que cargue los datos cuando ingrese a la página.
    /// </summary>
    /// <param name="sender"></param>

```

```

/// <param name="e"></param>
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        DropDownListRoles.DataSource = FachadaAD.Instancia.LeerTodosRoles();
        DropDownListRoles.DataBind();

        CheckBoxActivo.Checked = true;
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón crear.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonCrear_Click(object sender, ImageClickEventArgs e)
{
    Usuario usuario = new Usuario();

    long usuariold = 0;

    usuario.Nombre = TextBoxNombre.Text;
    usuario.NombreIngreso = TextBoxNombreIngreso.Text;
    usuario.Clave = TextBoxClave.Text;
    usuario.Activo = CheckBoxActivo.Checked;
    Rol rol = FachadaAD.Instancia.LeerRol(long.Parse(DropDownListRoles.SelectedValue));
    usuario.Roles.Add(rol);

    try
    {
        usuariold = FachadaAD.Instancia.AgregarUsuario(usuario);
        //GuardarCasos(usuariold);
        Master.MostrarMensaje("Operación exitosa", "El Usuario se creó correctamente.",
        Icono.Informacion, Constantes.URL_GESTION_USUARIOS);
    }
    catch (UsuarioConNombreIngresoRepetidoExcepcion)
    {
        Master.MostrarMensaje("Error", "Ya existe una Usuario con el nombre de ingreso " +
        usuario.NombreIngreso, Icono.Error, Constantes.URL_GESTION_USUARIOS);
    }
    catch (Exception ex)
    {
        Master.MostrarMensaje("Error", "Error creando Usuario - " + ex.Message, Icono.Error,
        Constantes.URL_GESTION_USUARIOS);
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón cancelar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>

```



```
protected void ImageButtonCancelar_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect(Constants.URL_GESTION_USUARIOS, true);
}
```

```
Editar.aspx
<%@ Page Language="C#" MasterPageFile="~/Sitio.master" AutoEventWireup="true"
CodeFile="Editar.aspx.cs" Inherits="CRUD_Usuario_Editar" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>
<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
  <table style="width: 60%; border="0" cellpadding="0" cellspacing="0" align="center">
    <tr>
      <td class="style1">
        <h1>Editar Usuario</h1>
      </td>
    </tr>
    <tr>
      <td>
        <fieldset>
          <legend>Datos del usuario
          </legend>
          <table>
            <tr>
              <td class="formTitle">Nombre de ingreso:
            </td>
            <td>
              <asp:TextBox ID="TextBoxNombreIngreso" runat="server" CssClass="textBoxLargo"
              MaxLength="100" ToolTip="Nombre de ingreso del usuario, mínimo 5 caracteres."></asp:TextBox>
              <asp:RequiredFieldValidator ID="RequiredFieldValidatorNombreIngreso"
              runat="server" ControlToValidate="TextBoxNombreIngreso"
              ErrorMessage="Campo Obligatorio" CssClass="validador" Font-Bold="True" Font-
              Size="0pt" ValidationGroup="validar">
              <asp:RequiredFieldValidator>
              <ajaxToolkit:ValidatorCalloutExtender ID="ValidatorCalloutExtenderNombreIngreso"
              runat="server"
              TargetControlID="RequiredFieldValidatorNombreIngreso"
              WarningImageUrl="~/Imgs/Advertencia.png">
              <ajaxToolkit:ValidatorCalloutExtender>
            </td>
          </tr>
        </table>
      </td>
    </tr>
    <tr>
      <td class="formTitle">Clave:
      </td>
      <td style="margin-left: 40px">
        <asp:TextBox ID="TextBoxClave" runat="server" CssClass="textBoxLargo"
        MaxLength="255" TextMode="Password" ToolTip="Clave o password, mínimo 5
        caracteres."></asp:TextBox>
      </td>
    </tr>
  </table>
</asp:Content>
```

```

<tr>
  <td class="formTitle">Confirmar clave:
</td>
<td>
  <asp:TextBox ID="TextBoxConfirmarClave" runat="server"
  CssClass="textBoxLargo" MaxLength="255" TextMode="Password" ToolTip="Confirmar la clave o
  password, mínimo 5 caracteres."></asp:TextBox>
  <asp:CompareValidator ID="CompararClaves" runat="server"
  ControlToCompare="TextBoxClave" ControlToValidate="TextBoxConfirmarClave"
  ErrorMessage="Claves diferentes." ValidationGroup="validar" Font-Size="0pt">
  </asp:CompareValidator>
  <ajaxToolkit:ValidatorCalloutExtender
  ID="ValidatorCalloutExtenderCompararClaves" runat="server"
  TargetControlID="CompararClaves"
  WarningIconImageUrl="~/Imgs/Advertencia.png">
  </ajaxToolkit:ValidatorCalloutExtender>
</td>
</tr>
<tr>
  <td class="formTitle">Nombre:
</td>
<td>
  <asp:TextBox ID="TextBoxNombre" runat="server" CssClass="textBoxLargo"
  MaxLength="100" ToolTip="Nombre completo del usuario."></asp:TextBox>
  <asp:RequiredFieldValidator ID="RequiredFieldValidatorNombre" runat="server"
  ControlToValidate="TextBoxNombre"
  ErrorMessage="Campo Obligatorio" CssClass="validador" Font-Bold="True" Font-
  Size="0pt" ValidationGroup="validar">
  </asp:RequiredFieldValidator>
  <ajaxToolkit:ValidatorCalloutExtender ID="ValidatorCalloutExtenderNombre"
  runat="server"
  TargetControlID="RequiredFieldValidatorNombre"
  WarningIconImageUrl="~/Imgs/Advertencia.png">
  </ajaxToolkit:ValidatorCalloutExtender>
</td>
</tr>
<tr>
  <td class="formTitle">Rol:
</td>
<td>
  <asp:DropDownList ID="DropDownListRoles" runat="server" AutoPostBack="false"
  DataTextField="Nombre" DataValueField="Id" CssClass="DropDownListMedio">
  </asp:DropDownList>
</td>
</tr>
<tr>
  <td class="formTitle">Activo:
</td>
<td>
  <asp:CheckBox ID="CheckBoxActivo" runat="server" />
</td>
</tr>
</table>

```

```

        </fieldset>
    </td>
</tr>
<tr>
    <td style="text-align: right">
        <asp:ImageButton ID="ImageButtonActualizar" runat="server"
        ImageUrl="~/Imgs/Actualizar.png" OnClick="ImageButtonActualizar_Click" ToolTip="Actualizar Usuario"
        ValidationGroup="validar" />
        <asp:ImageButton ID="ImageButtonCancelar" runat="server" ImageUrl="~/Imgs/Cancel.png"
        OnClick="ImageButtonCancelar_Click" ToolTip="Cancelar" CausesValidation="False" />
    </td>
</tr>
</table>
</asp:Content>

```

Editar.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using ADNucleo.Fachada;
using ADComun.Modelo;
using ADComun.Excepciones;
using ADWeb.Constantes;
using System.Globalization;
using ADComun.Enums;
using ADWeb.App_Code;

/// <summary>
/// Clase para editar el usuario.
/// </summary>
public partial class CRUD_Usuario_Editar : System.Web.UI.Page
{
    /// <summary>
    /// Lógica para que cargue los datos cuando ingrese a la página.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            long usuariold = long.Parse(Request.QueryString[Constantes.PARAMETRO_USUARIO_ID]);

            DropDownListRoles.DataSource = FachadaAD.Instance.LeerTodosRoles();
            DropDownListRoles.DataBind();

            Usuario usuario;

```

```

        usuario = FachadaAD.Instancia.LeerUsuario(usuarioId);
        TextBoxNombre.Text = usuario.Nombre;
        TextBoxNombreIngreso.Text = usuario.NombreIngreso;
        TextBoxConfirmarClave.Text = usuario.Clave;
        CheckBoxActivo.Checked = usuario.Activo;

        DropDownListRoles.SelectedValue = usuario.Roles.Last().Id.ToString();
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón Actualizar.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ImageButtonActualizar_Click(object sender, ImageClickEventArgs e)
{
    Usuario usuario;
    long usuarioId = long.Parse(Request.QueryString[Constantes.PARAMETRO_USUARIO_ID]);

    usuario = FachadaAD.Instancia.LeerUsuario(usuarioId);
    usuario.Nombre = TextBoxNombre.Text;
    usuario.NombreIngreso = TextBoxNombreIngreso.Text;
    if (TextBoxClave.Text != "" && TextBoxClave.Text.Trim().Length > 0)
    {
        usuario.Clave = TextBoxClave.Text;
    }
    usuario.Activo = CheckBoxActivo.Checked;
    Rol rol = FachadaAD.Instancia.LeerRol(long.Parse(DropDownListRoles.SelectedValue));
    usuario.Roles.Clear();
    usuario.Roles.Add(rol);

    try
    {
        FachadaAD.Instancia.ActualizarUsuario(usuario);
        Master.MostrarMensaje("Operación exitosa", "El usuario se actualizó correctamente.",
        Icono.Informacion, Constantes.URL_GESTION_USUARIOS);
    }
    catch (UsuarioConNombreIngresoRepetidoExcepcion)
    {
        Master.MostrarMensaje("Error", "Ya existe un usuario con el nombre de ingreso " +
        usuario.NombreIngreso, Icono.Error, Constantes.URL_GESTION_USUARIOS);
    }
    catch (Exception ex)
    {
        Master.MostrarMensaje("Error", "Error actualizando usuario - " + ex.Message, Icono.Error,
        Constantes.URL_GESTION_USUARIOS);
    }
}

/// <summary>
/// Evento que se dispara al dar clic sobre el botón cancelar.
/// </summary>
/// <param name="sender"></param>

```



```

<asp:GridView ID="GridViewUsuarios" runat="server" AutoGenerateColumns="False"
  CellPadding="4" DataKeyNames="Id" ForeColor="#333333" GridLines="None"
  HorizontalAlign="Center" OnRowCommand="GridViewUsuarios_RowCommand"
  EmptyDataText="No hay Usuarios registrados" Width="100%">
  <AlternatingRowStyle BackColor="White" ForeColor="#284775" />
  <Columns>
    <asp:BoundField DataField="NombreIngreso" HeaderText="Nombre de ingreso"
  />
    <asp:BoundField DataField="Nombre" HeaderText="Nombre">
      <ItemStyle HorizontalAlign="Left" />
    </asp:BoundField>
    <asp:TemplateField HeaderText="activo">
      <ItemTemplate>
        <asp:Image ID="ImageActivo" runat="server" ImageUrl="
ObtenerImagen((bool)Eval("Activo")) %>" />
      </ItemTemplate>
    </asp:TemplateField>
    <asp:TemplateField HeaderText="Opciones">
      <ItemStyle Width="100px" />
      <ItemTemplate>
        <asp:ImageButton ID="ImageButtonVer" runat="server"
ImageUrl='~/Imgs/Ver.png'
          Tooltip="Ver detalle" CommandName="ver"
          CommandArgument="<%# Container.DataItemIndex %>" />
        <asp:ImageButton ID="ImageButtonEditar" runat="server"
ImageUrl='~/Imgs/Editar.png' Tooltip="Editar" CommandName="editar" CommandArgument="<%#
Container.DataItemIndex %>" />
        <asp:ImageButton ID="ImageButtonEliminar" runat="server"
ImageUrl='~/Imgs/Delete.png' Tooltip="Eliminar" CommandName="eliminar" CommandArgument="<%#
Container.DataItemIndex %>" ValidationGroup="eliminar" />
      </ItemTemplate>
    </asp:TemplateField>
  </Columns>
  <EditRowStyle BackColor="#999999" />
  <FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
  <HeaderStyle BackColor="#000000" Font-Bold="True" ForeColor="White" />
  <PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />
  <RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
  <SelectedRowStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333"
  />
  </asp:GridView>
  </fieldset>
</ContentTemplate>
<Triggers>
  <asp:PostBackTrigger ControlID="GridViewUsuarios" />
</Triggers>
</asp:UpdatePanel>
</td>

```

```

</tr>
<p>
  <asp:CustomValidator ID="CustomValidatorEliminar" runat="server"
    ClientValidationFunction="confirmarEliminacion"
    ValidationGroup="eliminar"></asp:CustomValidator>
</p>
</table>
</asp:Content>

```

Gestion.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using ADComun.Modelo;
using ADNucleo.Fachada;
using ADWeb.Constantes;
using System.IO;
using System.Web.UI.HtmlControls;
using System.Text;
using ADComun.Excepciones;
using ADWeb.App_Code;

/// <summary>
/// Clase para la gestión de usuarios.
/// </summary>
public partial class CRUD_Usuario_Gestion : System.Web.UI.Page
{
    #region ViewState

    /// <summary>
    /// ViewState para el valor del nombre filtrado.
    /// </summary>
    string FiltroNombre
    {
        get
        {
            return (string)ViewState["filtroNombre"];
        }
        set
        {
            ViewState["filtroNombre"] = value;
        }
    }

    #endregion ViewState

    /// <summary>

```

```

/// Page_load de la gestión.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        CargarDatos();
    }
}

/// <summary>
/// Método para llenar los datos de la grilla.
/// </summary>
private void CargarDatos()
{
    string script = "parent.calcHeight()";

    long totalUsuarios = 0;
    IList<Usuario> _Usuarios = null;

    Boolean? activo;

    totalUsuarios = FachadaAD.Instancia.ContarTodosUsuarios();
    _Usuarios = FachadaAD.Instancia.BuscarUsuariosPorNombreIngreso(FiltroNombre);

    GridViewUsuarios.DataSource = _Usuarios;
    GridViewUsuarios.DataBind();
}

/// <summary>
/// Obtiene la imagen y la carga en la grilla.
/// </summary>
/// <param name="_Activa">Valor activo.</param>
/// <returns>url de la imagen.</returns>
public static string ObtenerImagen(bool _Activa)
{
    if (_Activa)
    {
        return "~/Imgs/Activo.png";
    }
    else
    {
        return "~/Imgs/Inactivo.png";
    }
}

/// <summary>
/// Lógica para los botones de la grilla.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void GridViewUsuarios_RowCommand(object sender, GridViewCommandEventArgs e)

```



```

    {
        int rowIndex = Convert.ToInt32(e.CommandArgument);
        long usuariold = (long)GridViewUsuarios.DataKeys[rowIndex].Value;
        switch (e.CommandName)
        {
            case "ver":
                Server.Transfer(Constants.URL_VER_USUARIO + "?" +
                Constants.PARAMETRO_USUARIO_ID + "=" + usuariold, false);
                break;
            case "editar":
                Server.Transfer(Constants.URL_EDITAR_USUARIO + "?" +
                Constants.PARAMETRO_USUARIO_ID + "=" + usuariold, false);
                break;
            case "eliminar":
                usuariold = (long)GridViewUsuarios.DataKeys[rowIndex].Value;
                try
                {
                    FachadaAD.Instancia.EliminarUsuario(usuariold);
                }
                catch (Exception ex)
                {
                    Master.MostrarMensaje("Error", "Error eliminando la Usuario - " + ex.Message, Icono.Error,
                    null);
                }
                CargarDatos();
                break;
        }
    }

    /// <summary>
    /// Evento que se dispara al dar clic sobre el botón crear.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void ImageButtonCrear_Click(object sender, ImageClickEventArgs e)
    {
        Server.Transfer(Constants.URL_CREAR_USUARIO);
    }

    /// <summary>
    /// Evento que se dispara al dar clic sobre el botón filtrar.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void ImageButtonFiltrar_Click(object sender, ImageClickEventArgs e)
    {
        FiltroNombre = TextBoxNombre.Text.Trim();
        CargarDatos();
    }
}

```

Ver.aspx

```
<%@ Page Language="C#" MasterPageFile="~/Sitio.master" AutoEventWireup="true"
CodeFile="Ver.aspx.cs" Inherits="CRUD_Usuario_Ver" %>
<%@ MasterType VirtualPath="~/Sitio.master" %>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="Server">
  <table style="width: 60%; border="0" cellpadding="0" cellspacing="0" align="center">
    <tr>
      <td class="style1">
        <h1>Ver usuario</h1>
      </td>
    </tr>
    <tr>
      <td>
        <fieldset>
          <legend>Datos del usuario
          </legend>
          <table>
            <tr>
              <td class="formTitle">Nombre de ingreso:
            </td>
            <td>
              <asp:TextBox ID="TextBoxNombreIngreso" runat="server" CssClass="textBoxLargo"
MaxLength="100" ToolTip="Nombre de ingreso del usuario, mínimo 5 caracteres."></asp:TextBox>
            </td>
            </tr>
            <tr>
              <td class="formTitle">Clave:
            </td>
            <td style="margin-left: 40px">
              <asp:TextBox ID="TextBoxClave" runat="server" CssClass="textBoxLargo"
MaxLength="255" TextMode="Password" ToolTip="Clave o password, mínimo 5
caracteres."></asp:TextBox>
            </td>
            </tr>
            <tr>
              <td class="formTitle">Confirmar clave:
            </td>
            <td>
              <asp:TextBox ID="TextBoxConfirmarClave" runat="server"
CssClass="textBoxLargo" MaxLength="255" TextMode="Password" ToolTip="Confirmar la clave o
password, mínimo 5 caracteres."></asp:TextBox>
            </td>
            </tr>
            <tr>
              <td class="formTitle">Nombre:
            </td>
            <td>
              <asp:TextBox ID="TextBoxNombre" runat="server" CssClass="textBoxLargo"
MaxLength="100" ToolTip="Nombre completo del usuario."></asp:TextBox>
            </td>
            </tr>
            <tr>
              <td class="formTitle">Rol:
            </td>
```

```

        <td>
            <asp:TextBox ID="TextBoxRol" runat="server" CssClass="textBoxLargo"
MaxLength="100" ></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td class="formTitle">Activo:
        </td>
        <td>
            <asp:CheckBox ID="CheckBoxActivo" runat="server" />
        </td>
    </tr>
</table>
<fieldset class="fieldset">
    <legend>Asignar usuarios al caso
    </legend>
    <table width="100%">
        <tr>
            <td class="ArbolRad">
                <asp:TreeView ID="TreeViewUsuarios" runat="server" CheckBoxes="true"
CheckChildNodes="true"
                OnClientNodeCollapsed="clientNodeExpanded"
                OnClientNodeExpanded="clientNodeExpanded"
                ExpandAnimation-Type="None" CollapseAnimation-Type="None"
                TriStateCheckBoxes="True" ShowCheckBoxes="Leaf" ForeColor="Black" Enabled="False">
                </asp:TreeView>
            </td>
        </tr>
    </table>
</fieldset>
</fieldset>
</td>
</tr>
<tr>
    <td style="text-align: right">&nbsp;   
        <asp:ImageButton ID="ImageButtonCancelar" runat="server" ImageUrl="~/Imgs/Cancel.png"
OnClick="ImageButtonCancelar_Click" ToolTip="Cancelar" CausesValidation="False" />
    </td>
</tr>
</table>
</asp:Content>

```

Ver.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using ADNucleo.Fachada;
using ADComun.Modelo;

```

```

using ADWeb.Constantes;
using System.Globalization;
using TOCC = ADComun.Constantes; //TOCC - ADComun Constantes

/// <summary>
/// Clase para ver los datos del usuario.
/// </summary>
public partial class CRUD_Usuario_Ver : System.Web.UI.Page
{
    /// <summary>
    /// Lógica para que cargue los datos cuando ingrese a la página.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            long usuariold = long.Parse(Request.QueryString[Constantes.PARAMETRO_USUARIO_ID]);

            Usuario usuario;
            usuario = FachadaAD.Instancia.LeerUsuario(usuariold);
            TextBoxNombre.Text = usuario.Nombre;
            TextBoxNombreIngreso.Text = usuario.NombreIngreso;
            TextBoxClave.Text = usuario.Clave;
            TextBoxConfirmarClave.Text = usuario.Clave;
            TextBoxRol.Text = usuario.Roles.Last().Nombre;
            CheckBoxActivo.Checked = usuario.Activo;
        }
    }

    /// <summary>
    /// Evento que se dispara al dar clic sobre el botón cancelar.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void ImageButtonCancelar_Click(object sender, ImageClickEventArgs e)
    {
        Response.Redirect(Constantes.URL_GESTION_USUARIOS, true);
    }
}

```

Sitio.master

```

<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Sitio.master.cs" Inherits="Sitio" %>

<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<script type="text/javascript">
    function confirmarEliminacion(oSrc, args) {

```

```

var result = confirm("Se eliminará el registro seleccionado. ¿Desea continuar?");
args.IsValid = result;
}
</script>

<script language="JavaScript">
function recargarPagina() {
window.location.reload();
}
</script>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title></title>
<link href="-/Styles/Principal.css" rel="stylesheet" type="text/css" />
</head>
<body>
<form id="form1" runat="server">
<asp:ToolkitScriptManager ID="ToolkitScriptManagerSitiomaster"
runat="server"></asp:ToolkitScriptManager>

<asp:Panel ID="PanelMesssageBox" runat="server">
<div class="messagebox" style="border:none;">
<fieldset class="fieldset" style="background:#F8F8F8">
<asp:UpdatePanel ID="UpdatePanelMessageBox" runat="server">
<ContentTemplate>
<div id="DivMessageBoxTitulo" runat="server">
<asp:Label ID="LabelMessageBoxTitulo" runat="server" ></asp:Label>
</div>
<div class="imagen">
&nbsp;
<asp:Image ID="ImageMessageBoxIcono" runat="server"
ImageUrl="-/Imgs/alerta.ico"/>
</div>
<div class="mensaje">
<asp:Label ID="LabelMessageBoxMensaje" runat="server" ></asp:Label>
</div>
<div class="botones">
<asp:Button ID="ButtonMessageBoxAceptar" runat="server" Text="Aceptar"
OnClick="ButtonMessageBoxAceptar_Click" CausesValidation="False" Width="100"/>
</div>
</ContentTemplate>
</asp:UpdatePanel>
</fieldset>
</div>
</asp:Panel>
<div style="display:none">
<asp:LinkButton ID="LinkButtonMessageBox" runat="server">No se debe ver</asp:LinkButton>
</div>
<asp:ModalPopupExtender ID="ModalPopupExtenderMessageBox" runat="server"
TargetControlID="LinkButtonMessageBox"
PopupControlID="PanelMesssageBox" BackgroundCssClass="modalBackground">

```

```

    </asp:ModalPopupExtender>
    <asp:ContentPlaceHolder ID="MainContent" runat="server">
    </asp:ContentPlaceHolder>
  </form>
</body>
</html>

```

Sitio.master.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using System.Data;
using AjaxControlToolkit;
using ADComun.Modelo;
using System.Web.UI.HtmlControls;
using ADWeb.App_Code;

/// <summary>
/// Clase de la lógica para la clase maestra.
/// </summary>
public partial class Sitio : System.Web.UI.MasterPage
{
    /// <summary>
    /// Controla la redirección de urls.
    /// </summary>
    private string RedirectUrl
    {
        get
        {
            return (string)ViewState["RedirectUrl"];
        }
        set
        {
            ViewState["RedirectUrl"] = value;
        }
    }

    /// <summary>
    /// Carga los datos de la clase al iniciar.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
        }
    }
}

```

```

/// <summary>
/// Muestra los mensajes para ser mostrados al usuario.
/// </summary>
/// <param name="_Titulo">Título del mensaje.</param>
/// <param name="_Texto">texto del mensaje.</param>
/// <param name="_Icono">Icono a mostrar.</param>
/// <param name="_RedirectUrl">URL a redirigir.</param>
public void MostrarMensaje(string _Titulo, string _Texto, Icono _Icono, string _RedirectUrl)
{
    Label labelMessageBoxTitulo = (Label)FindControl("LabelMessageBoxTitulo");
    Label labelMessageBoxMensaje = (Label)FindControl("LabelMessageBoxMensaje");
    ModalPopupExtender modalPopupExtenderMessageBox =
(ModalPopupExtender)FindControl("ModalPopupExtenderMessageBox");
    Image imageMessageBoxIcono = (Image)FindControl("ImageMessageBoxIcono");
    HtmlControl ivMessageBoxTitulo = (HtmlControl)FindControl("DivMessageBoxTitulo");
    Button buttonMessageBoxAceptar = (Button)FindControl("ButtonMessageBoxAceptar");
    ToolkitScriptManager toolkitScriptManagerSitoMaster =
(ToolkitScriptManager)FindControl("ToolkitScriptManagerSitoMaster");
    labelMessageBoxTitulo.Text = _Titulo;
    labelMessageBoxMensaje.Text = _Texto;
    modalPopupExtenderMessageBox.Show();
    string url = null;
    string estiloTitulo = null;
    switch (_Icono)
    {
        case Icono.Alerta:
            url = "~/Imgs/alerta.ico";
            estiloTitulo = "tituloAlerta";
            break;
        case Icono.Error:
            url = "~/Imgs/error.ico";
            estiloTitulo = "tituloError";
            break;
        case Icono.Informacion:
            url = "~/Imgs/informacion.ico";
            estiloTitulo = "tituloInformacion";
            break;
        case Icono.Ok:
            url = "~/Imgs/ok.ico";
            estiloTitulo = "tituloOk";
            break;
    }
    imageMessageBoxIcono.ImageUrl = url;
    ivMessageBoxTitulo.Attributes["class"] = estiloTitulo;
    if (_RedirectUrl != null)
    {
        RedirectUrl = _RedirectUrl;
    }
}

/// <summary>
/// Evento que se dispara cuando el usuario da clic sobre el botón aceptar.
/// </summary>

```

```
/// <param name="sender"></param>
/// <param name="e"></param>
protected void ButtonMessageBoxAceptar_Click(object sender, EventArgs e)
{
    ModalPopupExtender modalPopupExtenderMessageBox =
(ModalPopupExtender)FindControl("ModalPopupExtenderMessageBox");
    modalPopupExtenderMessageBox.Hide();
    if (RedirectUrl != null)
    {
        Response.Redirect(RedirectUrl, false);
    }
}
```


Anexo 4. Manuales técnicos de operación, seguridad y administración del sistema.

Manual Técnico de configuración de Pentaho Reporting Server.

INFORMATION SECURITY STATUS

Aplicación de implementación de Sistemas de Gestión de Seguridad de la Información SGSI

Manual técnico de configuración de Pentaho
Versión 1.0

Configuración del servidor de reportes Pentaho.



1. Introducción

El servidor de reportes Pentaho permite generar los reportes asociados a las normas implementadas en el sistema de información ISS (Information Security Status) de forma autónoma, tomando la información directamente de la base de datos de la aplicación.

2. Objetivo

Configurar el servidor de reportes Pentaho para tomar la información del sistema de información ISS y generar los reportes configurados.

3. Términos y definiciones

ISS: Information Security Status, aplicación desarrollada para la implementación de normas de gestión de la seguridad de la información.

Pentaho: Es una suite open-source que permite crear reportes relacionales y analíticos de un amplio rango de fuentes de datos y salidas de datos como: PDF, Excel, HTML, Texto, RTF (Rich-Text-File), XML y CSV.

4. Desarrollo del documento

4.1. Condiciones generales

- Agregar variable del sistema JAVA_HOME: es necesario agregar esta variable para que el servidor Pentaho Business Analytics funcione correctamente.

<http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=287#configuraciontomcat>

- Abrir Pentaho Business Analytics: Para plataformas Microsoft Windows se realiza doble clic sobre el archivo “start-pentaho.bat”:

<http://wiki.pentaho.com/display/ServerDoc2x/Basic+BI+Platform+Documentation>

- Crear reporte en Pentaho Report Designer: El reporte es creado por los desarrolladores del proyecto, la creación de reportes puede ser consultada a través del siguiente sitio:

<http://wiki.pentaho.com/display/Reporting/Pentaho+Reporting+-+User+Guide+for+Report+Designer>>

- Cargar reporte en Pentaho BI: Descrito en el presente manual.

4.2. Pasos para la configuración de conexión ODBC

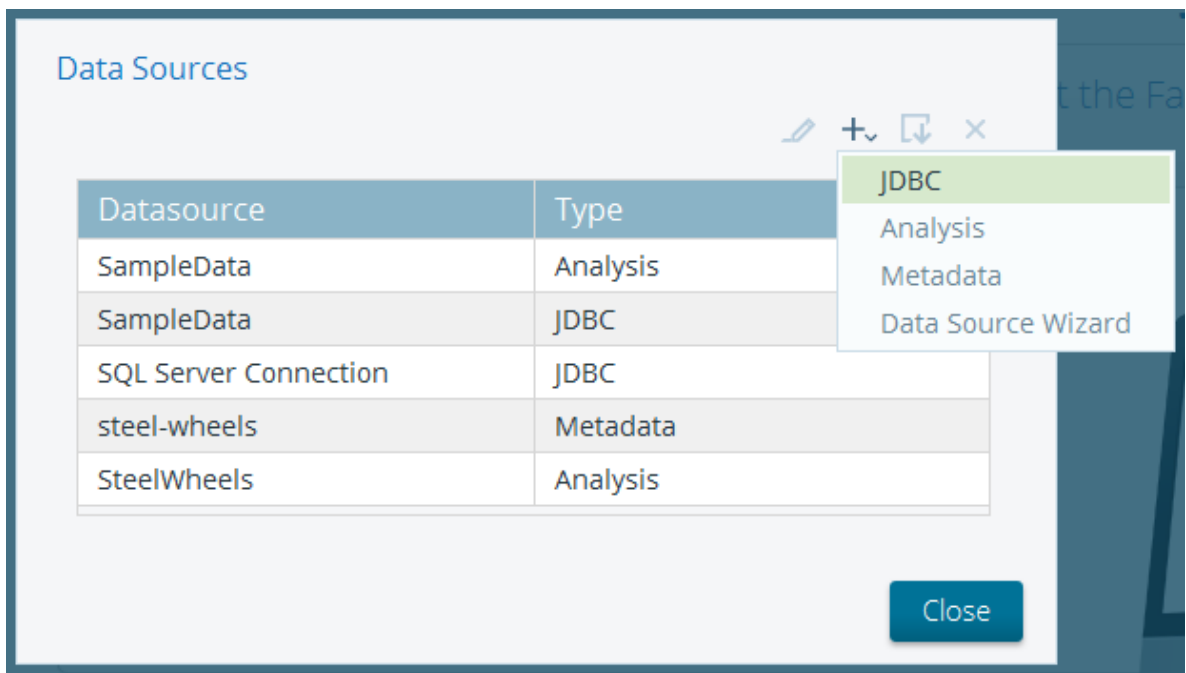
4.2.1. Iniciar sesión en el servidor Pentaho BI a través de la dirección http://ip_servidor:8080, e ingresar con las siguientes credenciales: Usuario: admin, contraseña: password.



4.2.2. Seleccionar el botón “Manage Data Sources”



4.2.3. Seleccionar “Add” Datasource y seleccionar la opción “JDBC”.



4.2.4. Seleccionar el tipo de base de datos “MS SQL Server”, seleccionar el Acceso como “Native (JDBC)”, asignar un nombre a la conexión y configurar los parámetros de conexión. El usuario por defecto de conexión a la base de datos Gestión es “ISS” y su contraseña “123ISS456!”. Seleccionar el botón OK para guardar los cambios.

Database Connection

General
Advanced
Options
Pooling

Connection Name:
SQL Server

Database Type:
MySQL
Hadoop Hive
Hadoop Hive 2
Hypersonic
Impala
MS SQL Server
PostgreSQL
H2

Access:
Native (JDBC)
ODBC
JNDI

Settings

Host Name:
localhost

Database Name:
Gestion

Port Number:
1433

User Name:
ISS

Password:
●●●●●●●●

[Adding Databases](#)

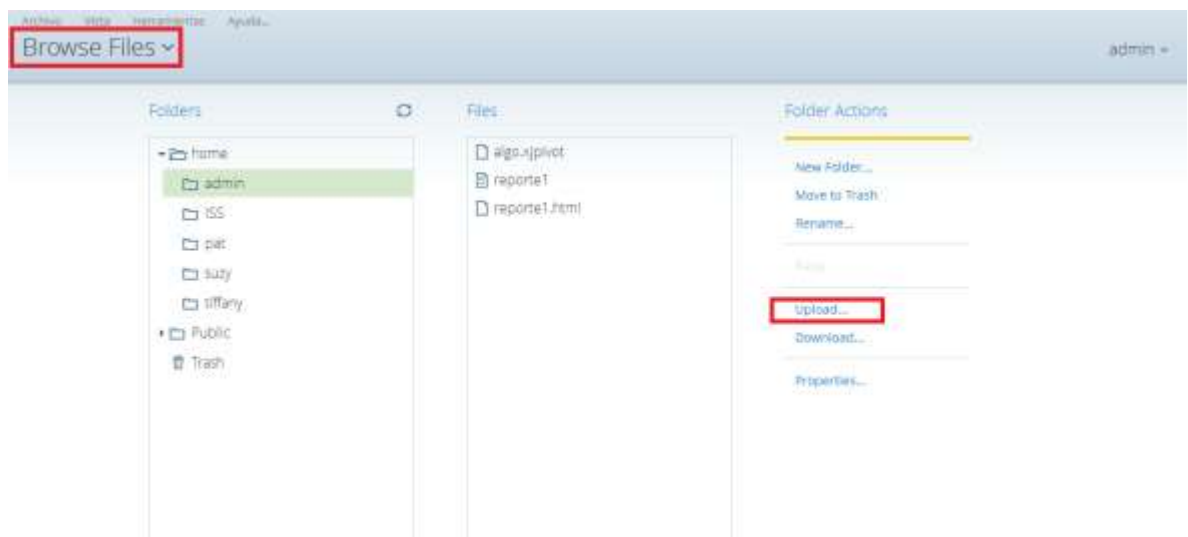
Test

4.3. Cargar reporte

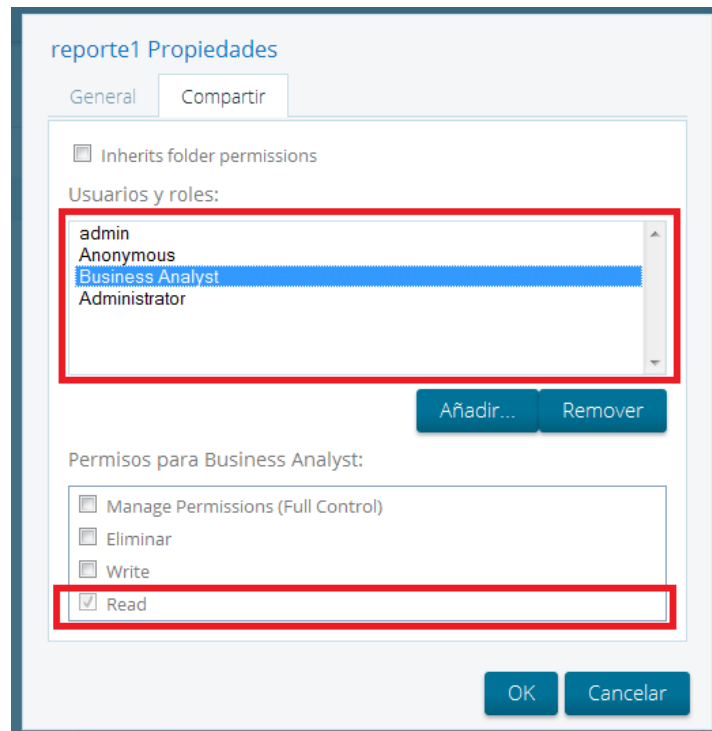
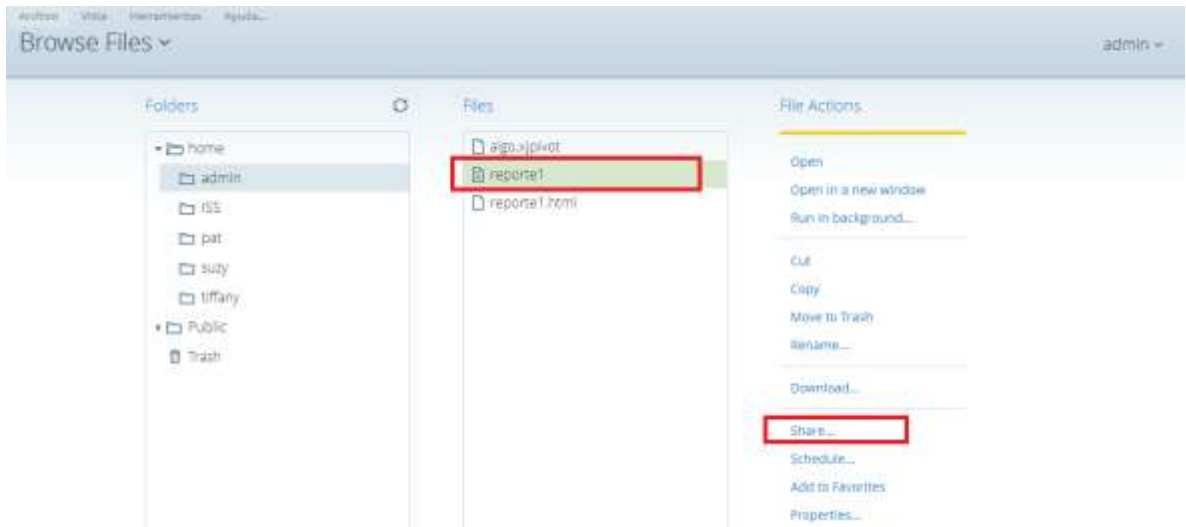
4.3.1. Seleccionar el botón “Browse Files”



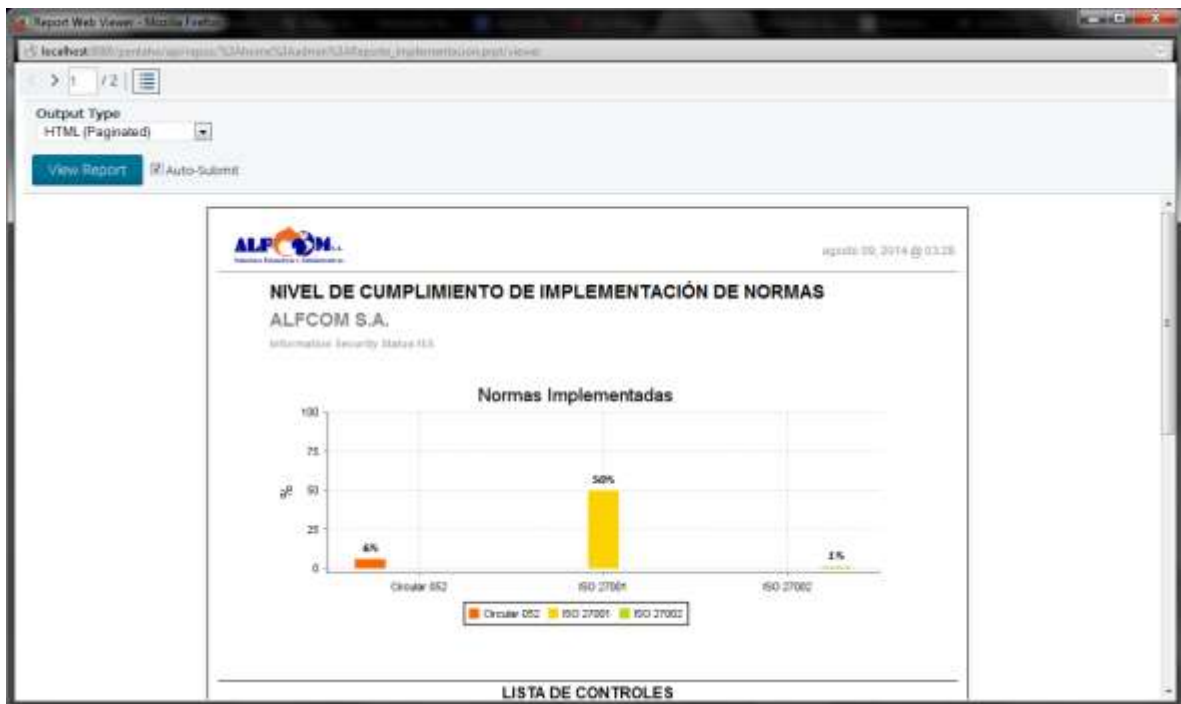
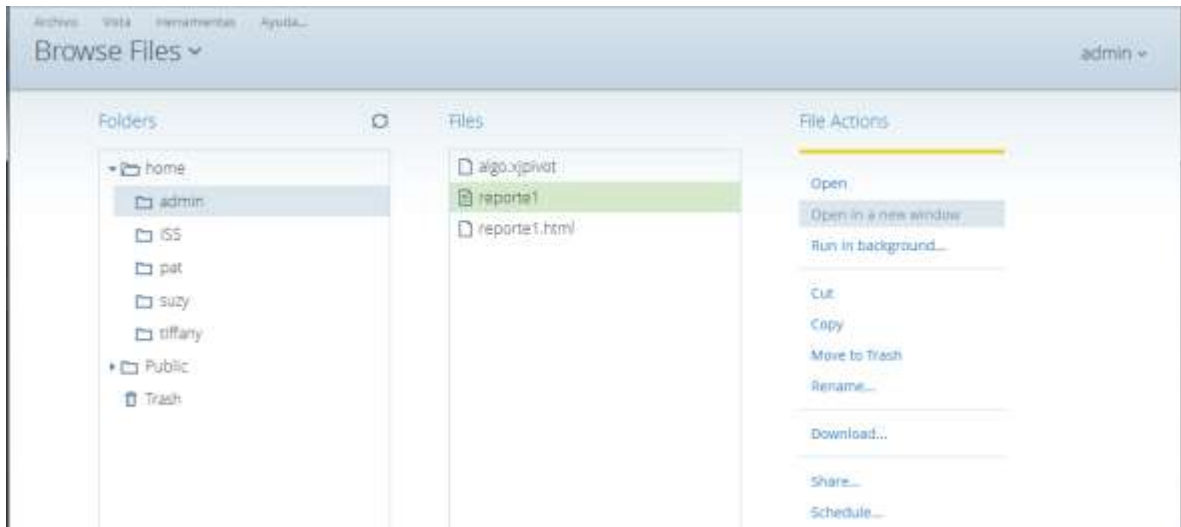
4.3.2. Seleccionar el botón “Upload” y subir el archivo del reporte de la norma.



4.3.3. Compartir el archivo del reporte y su archivo html para los usuarios que vayan a tener autorización para acceder al reporte.



4.3.4. Abrir el reporte en una nueva ventana, el enlace generado será el utilizado por la aplicación ISS para mostrar el reporte de la norma a consultar.



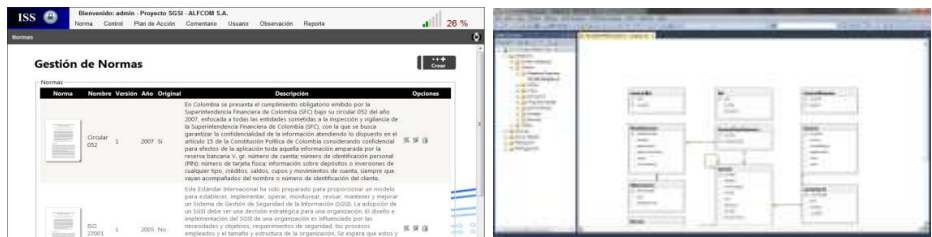
Manual de seguridad de la aplicación.

INFORMATION SECURITY STATUS

Aplicación de implementación de Sistemas de Gestión de Seguridad de la Información SGSI

Manual de configuración de seguridad
Versión 1.0

Configuración y prácticas de seguridad de la aplicación ISS.



1. Introducción

A través de este manual se describen las configuraciones y prácticas a realizar sobre la aplicación Information Security Status ISS, con el fin de proteger los puntos considerados como vulnerables dada la configuración por defecto del sistema.

2. Objetivo

Asegurar las configuraciones por defecto del sistema y proponer las mejores prácticas de uso de la aplicación con el fin de garantizar la confidencialidad, integridad y disponibilidad de la información gestionada.

3. Términos y definiciones

ISS: Information Security Status, aplicación desarrollada para la implementación de normas de gestión de la seguridad de la información.

Microsoft SQL Server: Microsoft® SQL Server™ es un sistema de administración y análisis de bases de datos relacionales de Microsoft para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos.

4. Desarrollo del documento

4.4. Condiciones generales

- Se describen las mejores prácticas para el aseguramiento de la aplicación Information Security Status ISS y la información por esta gestionada.

4.5. Cambiar contraseña de acceso a los reportes

Se recomienda cambiar la contraseña de acceso por defecto de los usuarios el reportador Pentaho, con el fin de garantizar que solo los usuarios autorizados pueden acceder a los reportes de la aplicación.

Acceder al servidor Pentaho Business Intelligence a través del enlace [http://ip_del_servidor:8080 /pentaho/Login](http://ip_del_servidor:8080/pentaho/Login). El usuario de acceso por defecto es “admin” y su contraseña “password”



Acceder al menú Home, Administration:



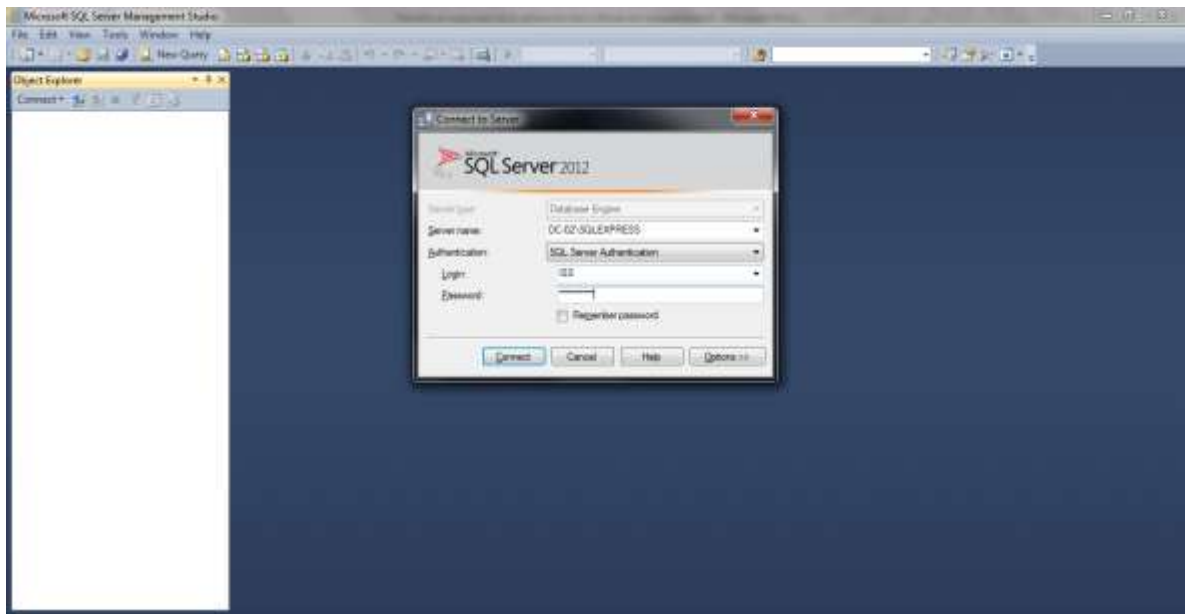
Seleccionar cada uno de los usuarios y luego presionar el botón editar, para cambiar la contraseña.



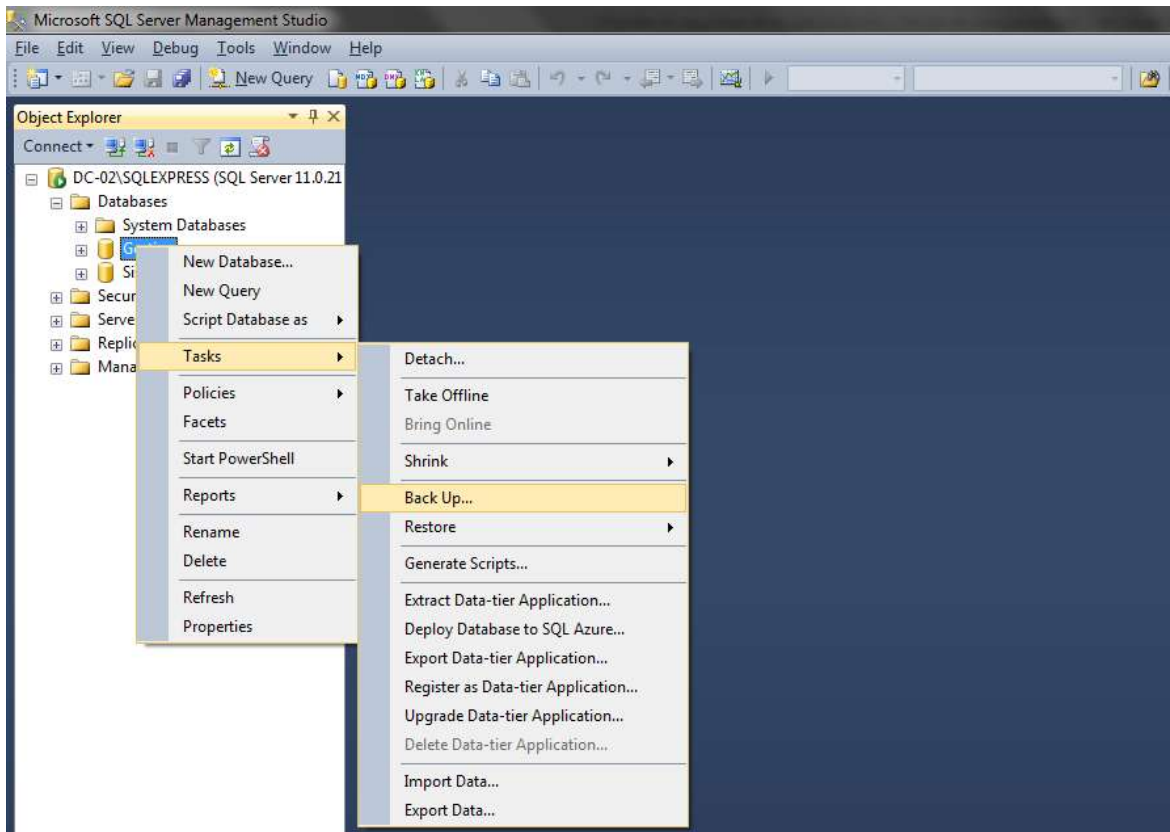
Guardar los cambios.

4.6. Copia de seguridad de la base de datos de la aplicación.

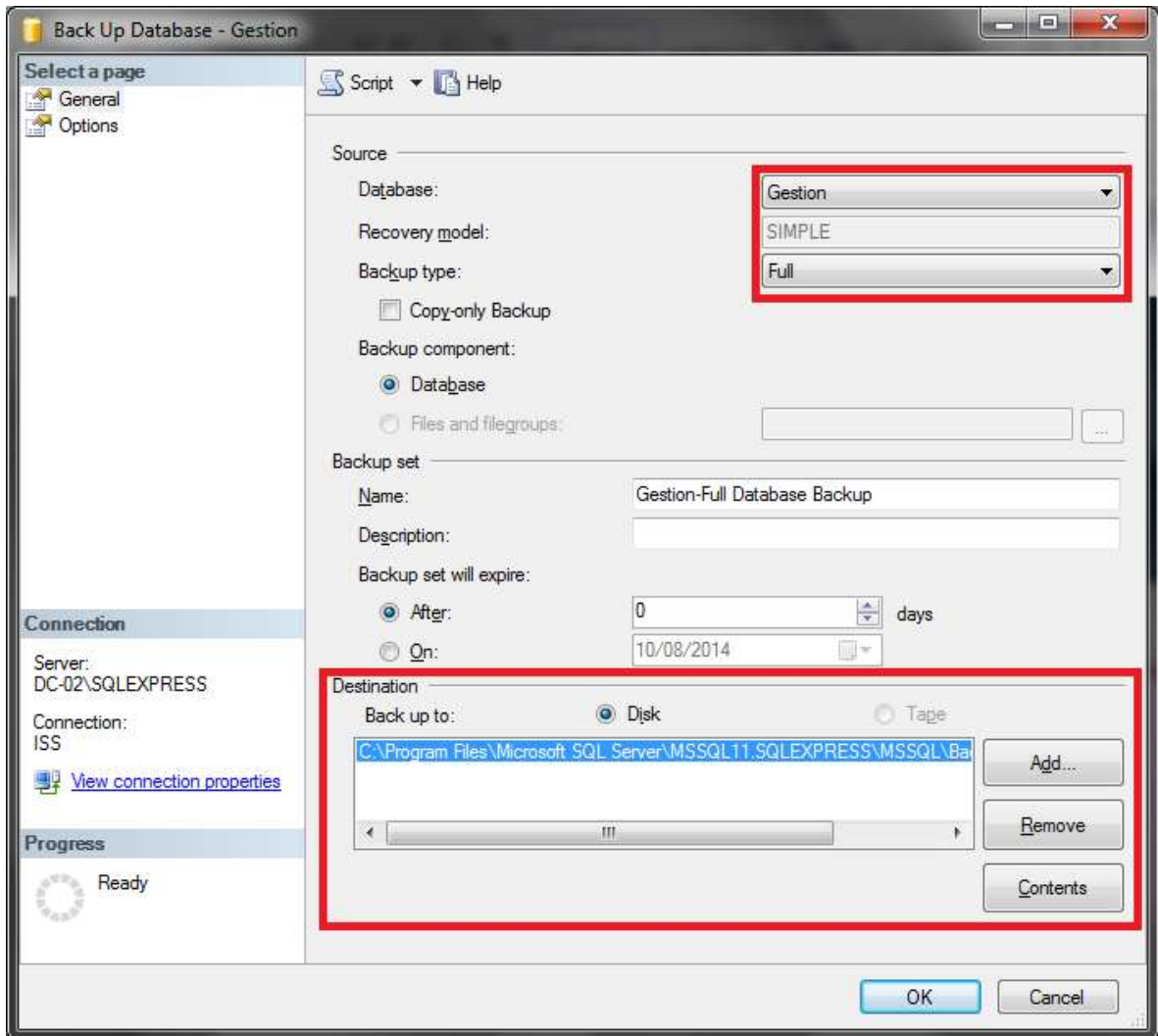
Acceder al Sistema Gestor de Bases de Datos Microsoft SQL Server 2012 con las credenciales usuario "ISS" y password "123ISS456!".



Dar clic derecho sobre la base de datos Gestion, y luego seleccionar los menús Task, Backup.



Agregar una ruta destino para guardar la copia de la base de datos bajo la sección “Destination” y luego dar clic en el botón OK para realizar la copia de respaldo.

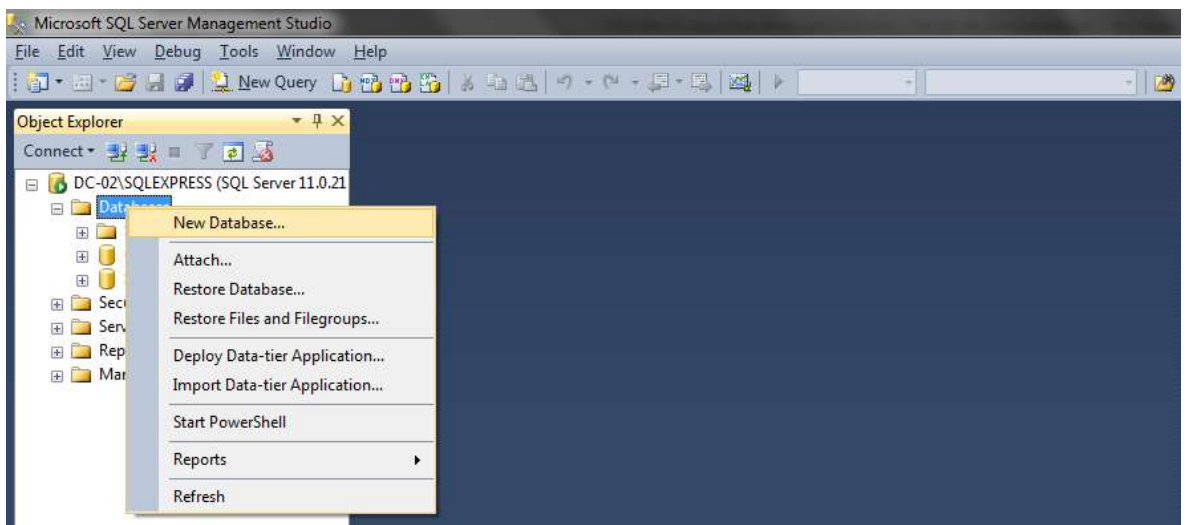


4.7. Restaurar la base de datos de la aplicación.

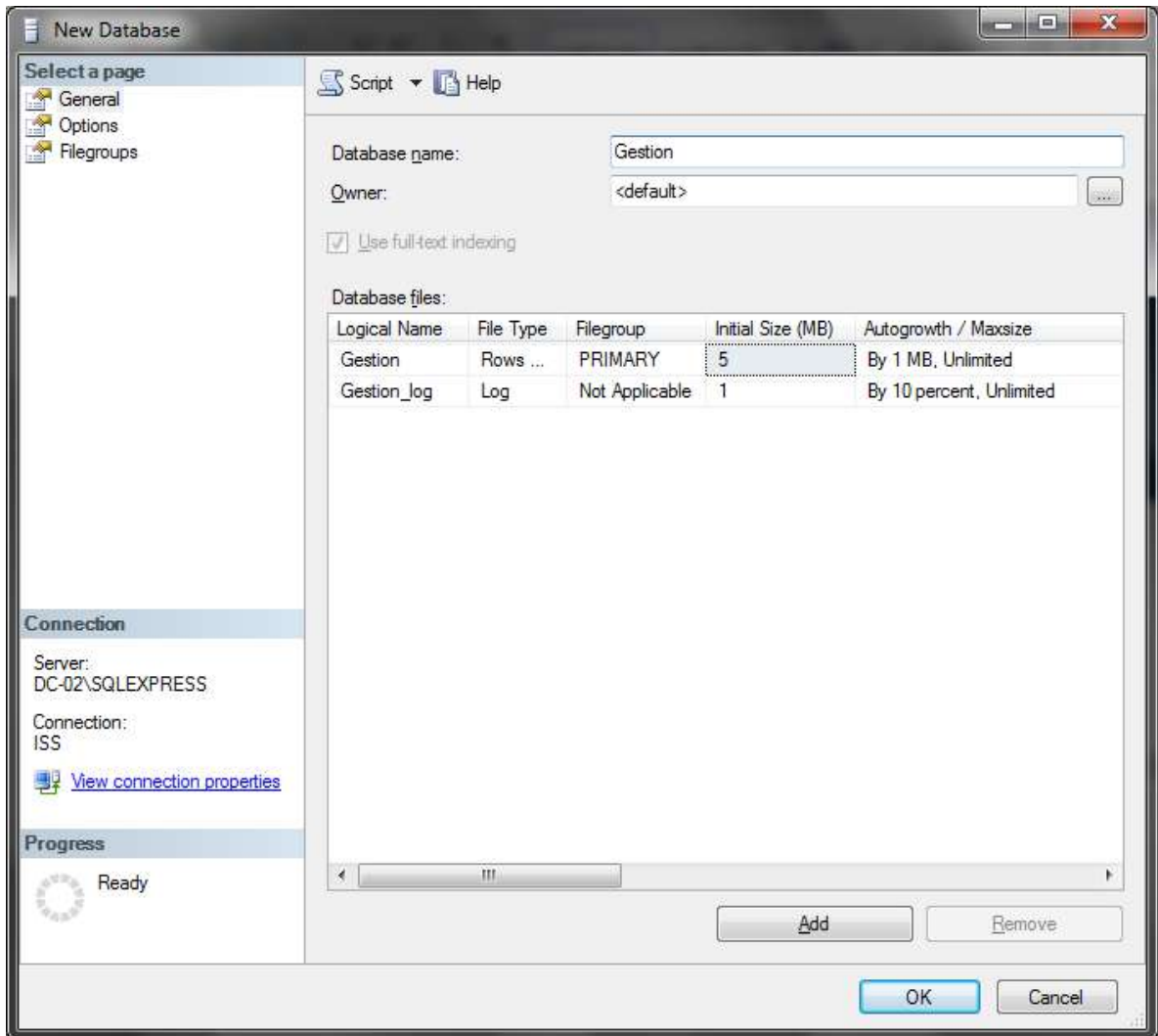
Acceder al Sistema Gestor de Bases de Datos Microsoft SQL Server 2012 con las credenciales usuario "ISS" y password "123ISS456!".



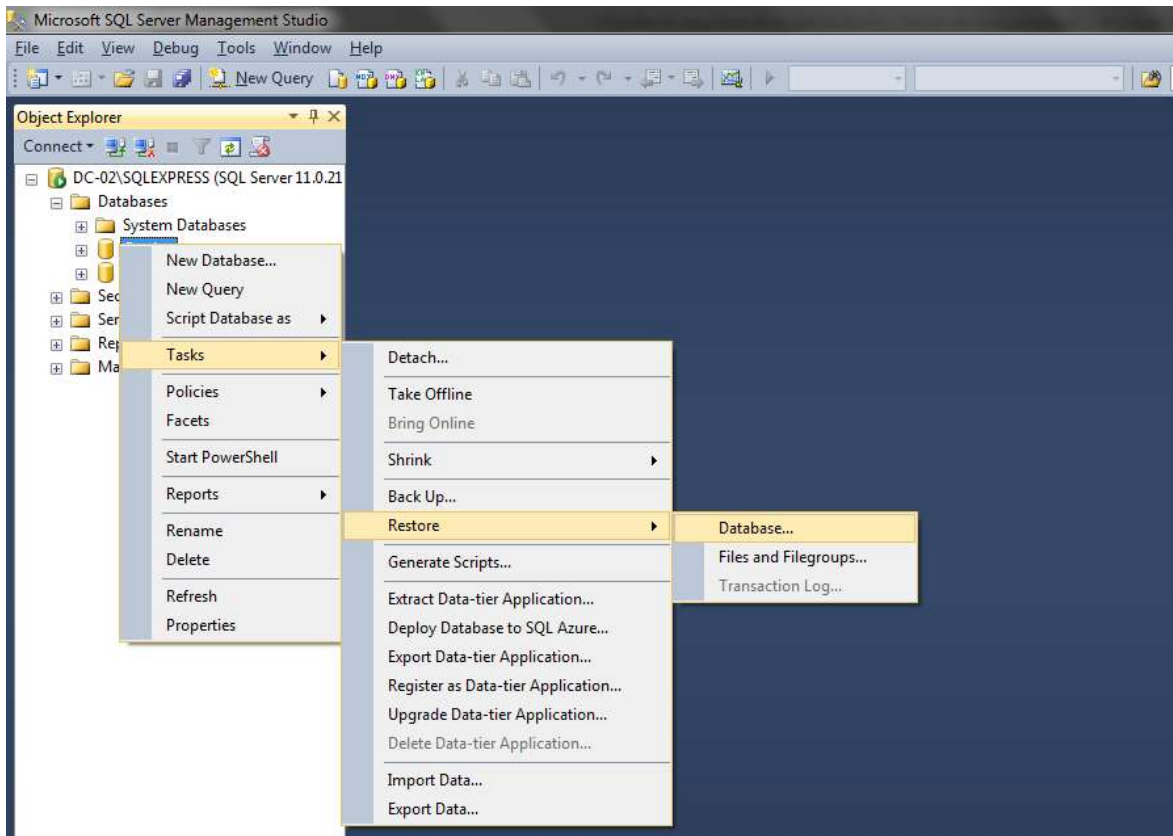
Dar clic derecho sobre la carpeta "Databases" del "Object Explorer" y luego dar clic en el botón "New Database".



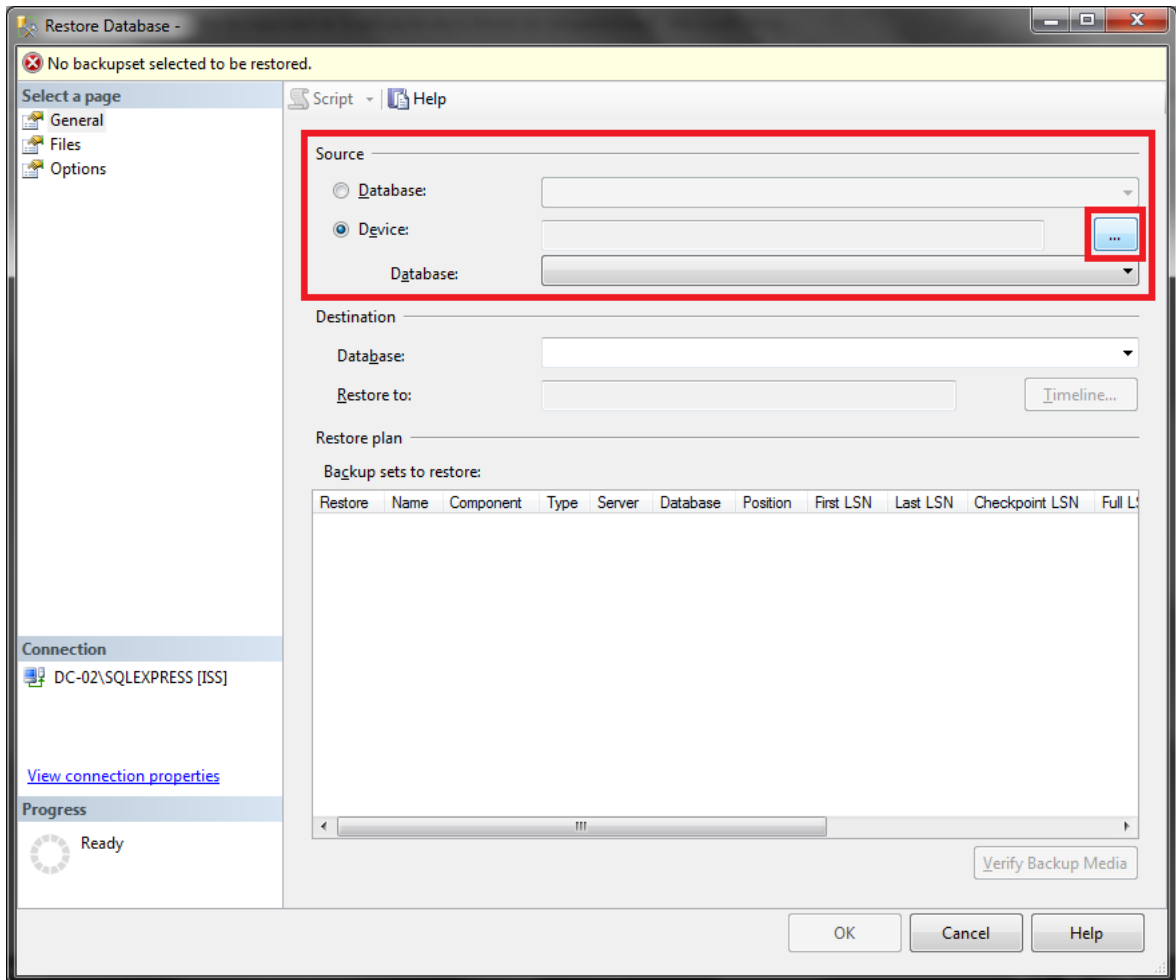
Nombrar la base de datos como “Gestion” y dar clic en el botón “OK” para guardar los cambios.



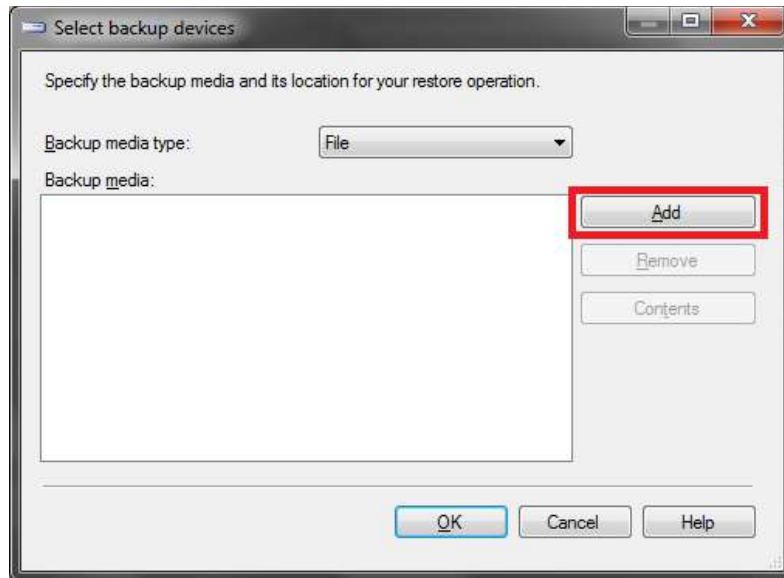
Realizar clic derecho sobre la base de datos “Gestion” y luego seleccionar los menús Tasks, Restore y Database.



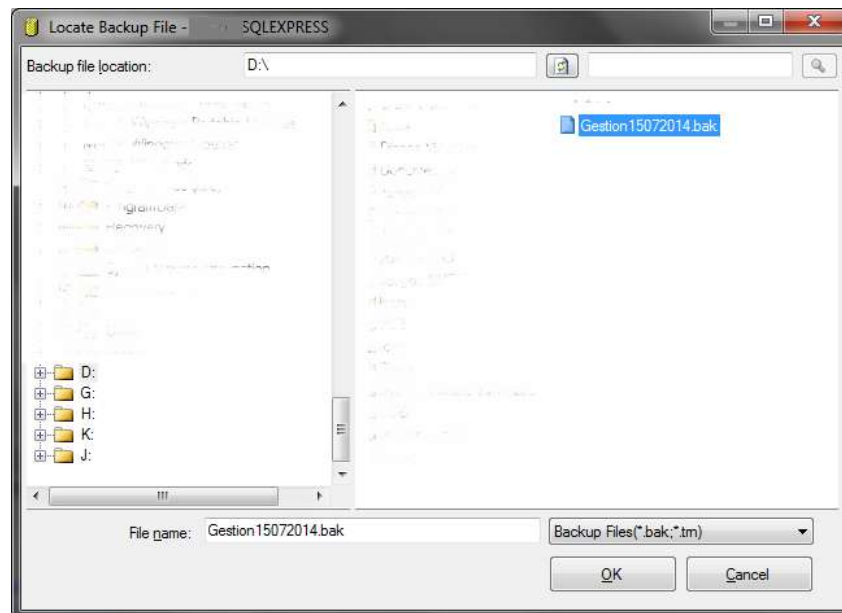
En la ventana “Restore Database” seleccionar la fuente “Device” en la sección “Source” para buscar la base de datos a restaurar.



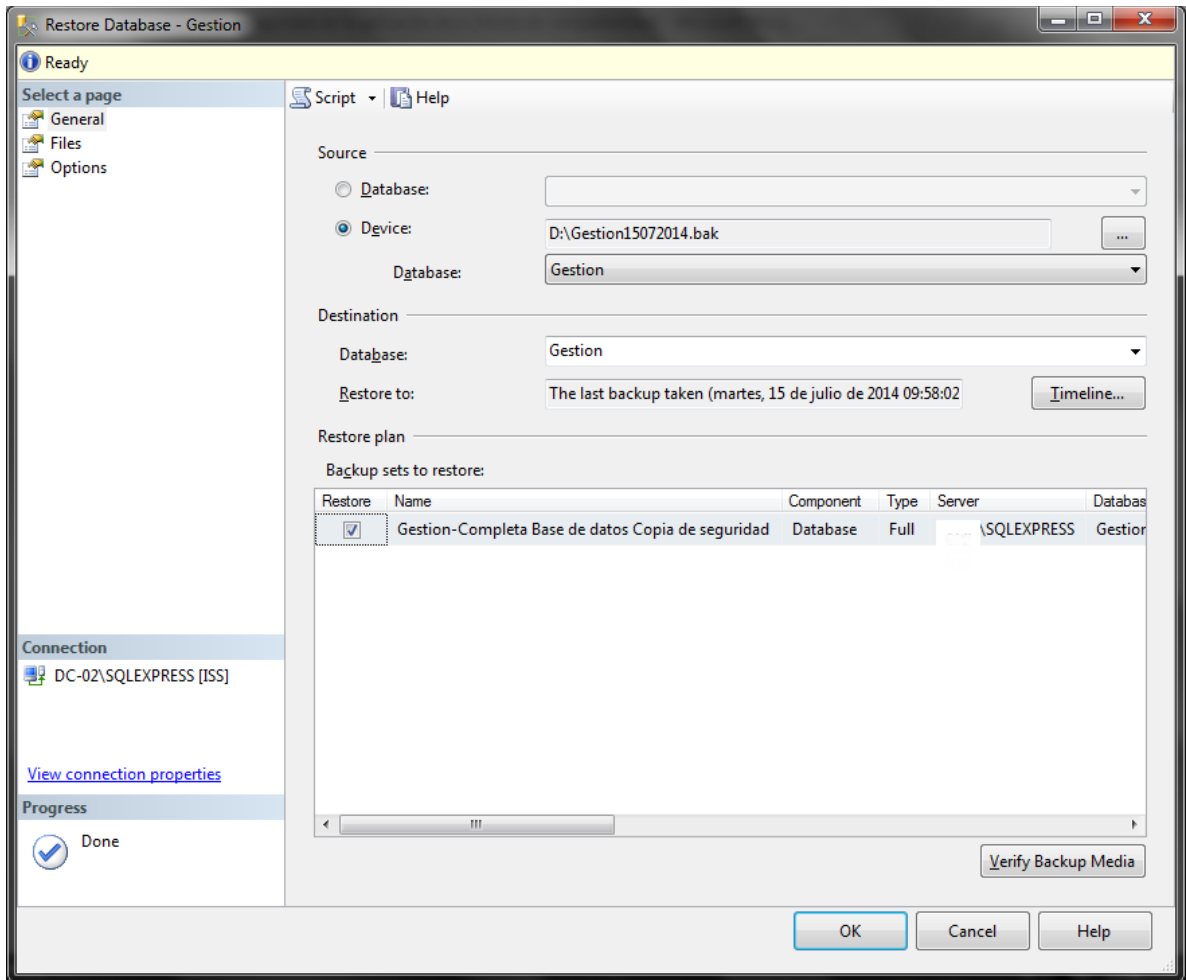
En la ventana “Select backup devices” presionar el botón “Add”.



Buscar el archivo de copia de seguridad de la base de datos en la estructura de directorios y presionar el botón OK y luego nuevamente OK.



Finalmente presionar el botón OK para realizar la restauración de los datos.



Manual de instalación de la aplicación.

INFORMATION SECURITY STATUS

Aplicación de implementación de Sistemas de Gestión de Seguridad de la Información SGSI

Manual de instalación
Versión 1.0

Configuración de la aplicación ISS.



1. Introducción

Para la correcta instalación de la aplicación en el servidor se debe tener en cuenta que es necesaria la instalación previa de Microsoft SQL server y el servidor IIS, en el presente manual se presenta la configuración y puesta en marcha del aplicativo para la gestión y consulta de normas ISS.

2. Objetivo

Configurar el servidor de la aplicación ISS con los parámetros iniciales para el correcto funcionamiento en los navegadores web de los clientes o usuarios.

3. Términos y definiciones

ISS: Information Security Status, aplicación desarrollada para la implementación de normas de gestión de la seguridad de la información.

Microsoft SQL Server: Es un servidor que permite almacenar los datos ingresados a la aplicación de forma ordenada para trabajar con ellos en la aplicación ISS.

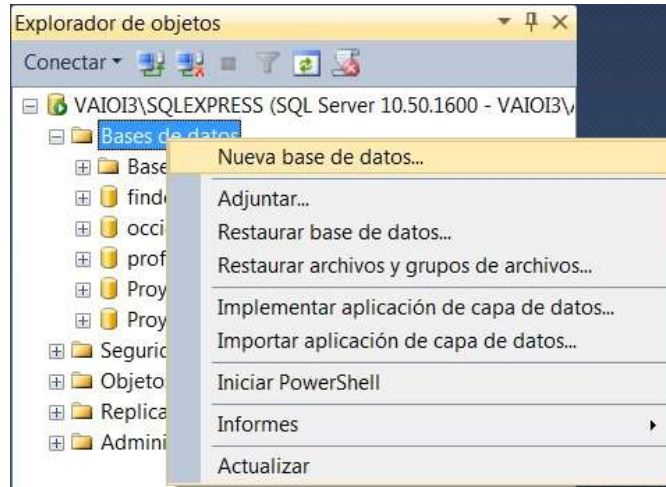
4. Desarrollo del documento

a. Condiciones generales

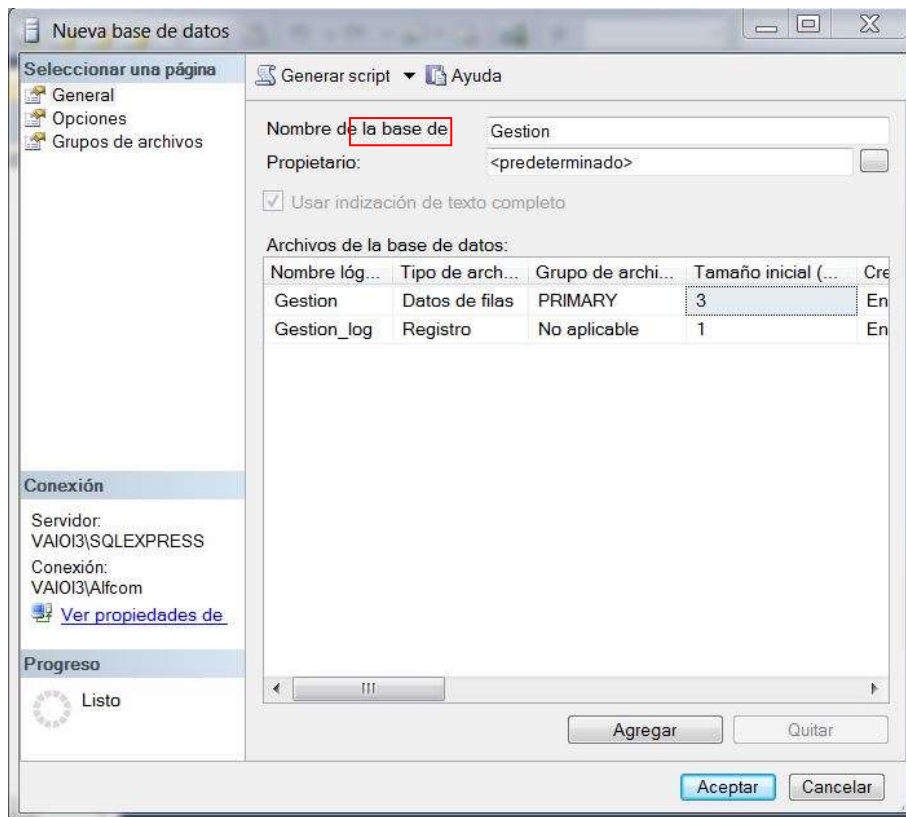
- El servidor debe tener instalado el motor de base de datos así como su managment para la configuración de la base de datos en versión 2008 o superior.
- La herramienta o rol de administrador IIS en su versión 7 o superior debe estar habilitada e instalada en el servidor.

4.1 Creación de la base de datos.

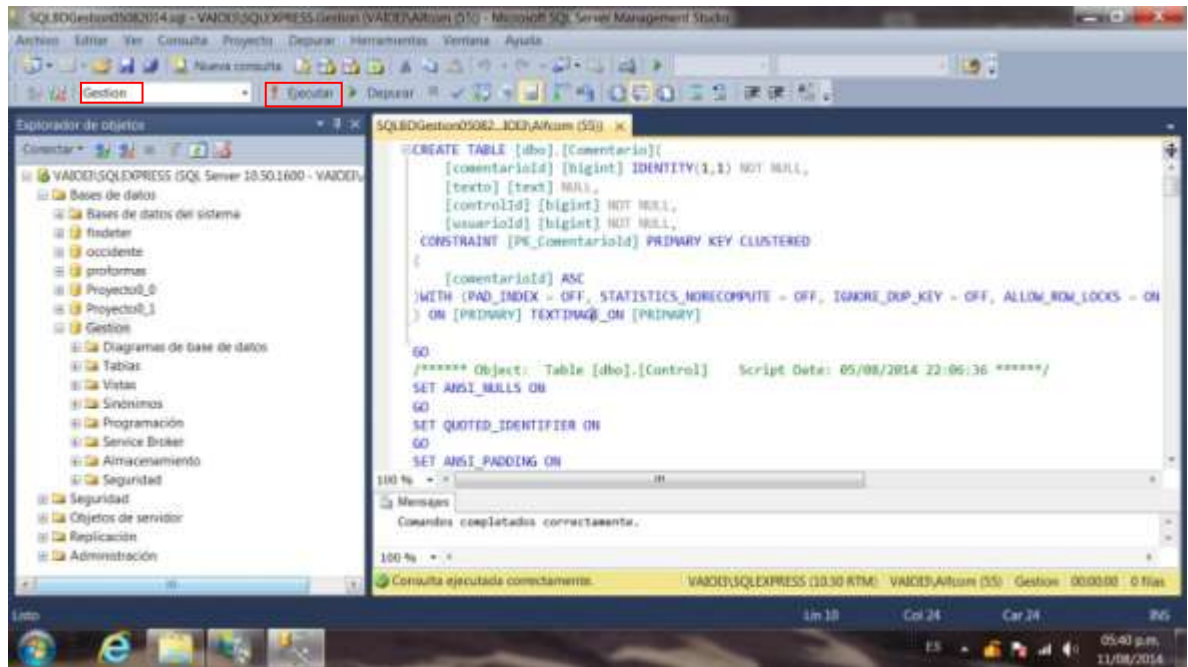
Ejecutar el programa SQL Server Management Studio, expandir el panel *el explorador de objetos*, clic derecho sobre base de datos y *Nueva base de datos*.



Crear una base de datos llamada *Gestion*, Clic en aceptar.

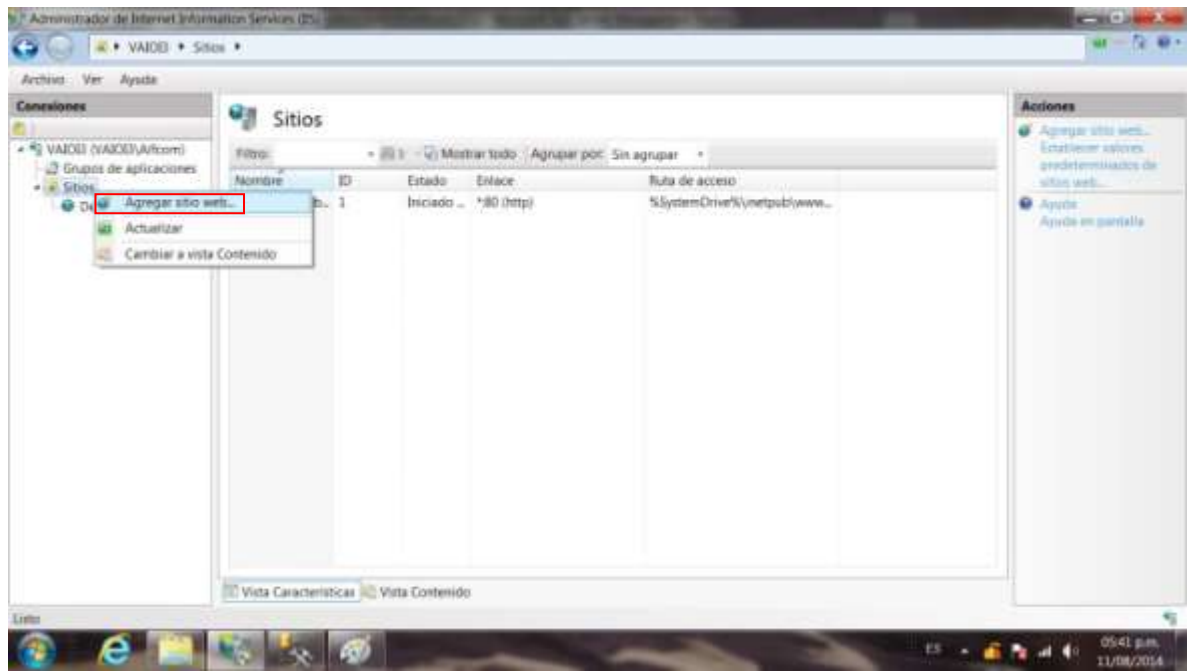


Clic sobre el botón *Nueva consulta*, Seleccionar la base de datos que se acaba de crear llamada *Gestion*, copiar y pegar el script inicial y luego dar clic en *Ejecutar*.



4.2 Creación del Sitio web.

Abrir el *Administrador de Internet Information Services (IIS)*, en el panel *Conexiones*, expandir el servidor, dar clic derecho sobre *Sitios* y luego *Agregar Sitio web...*



Al abrir la nueva pantalla *Agregar Sitio Web ISS* escribir el nombre del sitio, seleccionar *DefaultAppPool* como grupo de aplicaciones, buscar la ruta de acceso física en donde está almacenada la carpeta con el código fuente, seleccionar el nombre del Host y el puerto *8081*.

Agregar sitio web

Nombre del sitio: ISS Grupo de aplicaciones: DefaultAppPool Seleccionar...

Directorio de contenido

Ruta de acceso física: C:\Users\Alfcom\Documents\ISS ...

Autenticación de paso a través

Conectar como... Probar configuración...

Enlace

Tipo: http Dirección IP: Todas las no asignadas Puerto: 8081

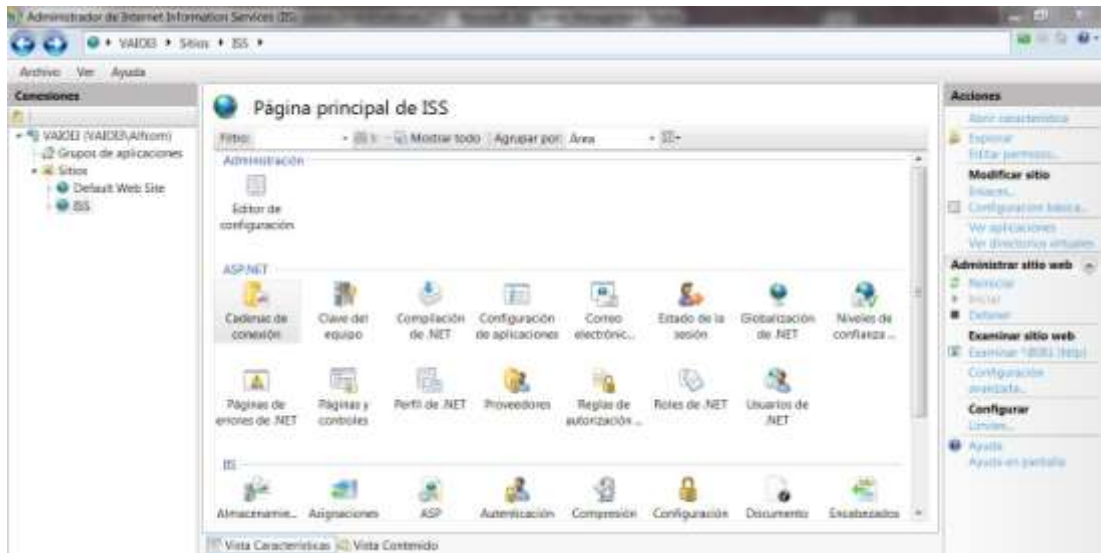
Nombre de host:

Ejemplo: www.contoso.com o marketing.contoso.com

Iniciar sitio web inmediatamente

Aceptar Cancelar

En la página principal del nuevo sitio, podrá modificar las opciones generales del sitio, como cadenas de conexión, páginas por defecto, entre otras.



Para iniciar la aplicación abra un navegador web, de preferencia Google Chrome o mozilla Firefox en sus últimas versiones o internet Explorer en su versión 10 o posterior, digite el nombre del servidor configurado en como nombre del HOST, puerto 8081 y la página principal, por ejemplo; ISS:8081/Login.aspx, la cual abrirá una pantalla como la siguiente.



Anexo 5. Manual de usuario de la aplicación Information Security Status

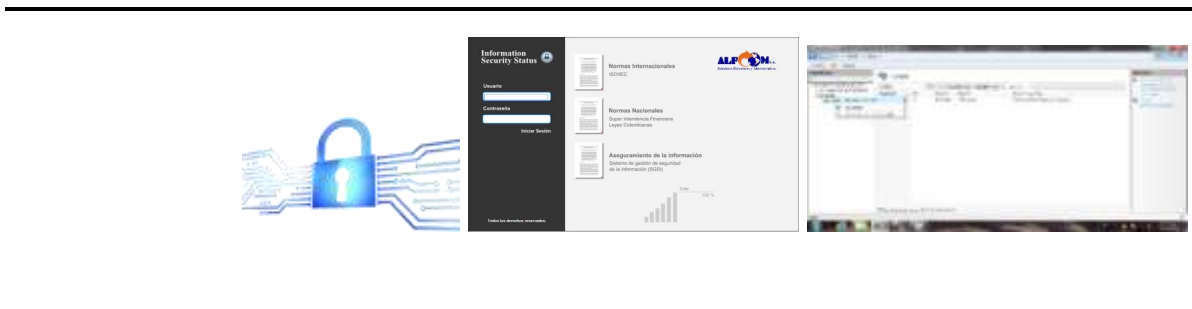
Manual de usuario.

INFORMATION SECURITY STATUS

Aplicación de implementación de Sistemas de Gestión de Seguridad de la Información SGSI

Manual de usuario
Versión 1.0

Manual de usuario de la aplicación ISS.



1. Introducción

El manual se usuario presenta la forma correcta de ejecutar y trabajar en la aplicación ISS, está dirigido a cualquier tipo de público con fines de aprendizaje basado en imágenes reales de la aplicación.

2. Objetivo

Dar una aproximación al usuario a la aplicación indicando la forma correcta y las diversas funciones de la aplicación ISS.

3. Términos y definiciones

ISS: Information Security Status, aplicación desarrollada para la implementación de normas de gestión de la seguridad de la información.

Navegador WEB: Es un programa de computador que permite el acceso a internet o aplicaciones internas desarrolladas de forma web.

4. Desarrollo del documento

b. Condiciones generales

- El administrador de la entidad deberá facilitar la dirección URL para el acceso a la aplicación.
- El usuario administrador tiene todos los privilegios sobre la aplicación.

4.3 Acceso a la aplicación ISS.

Ejecutar el navegador de internet y digitar el URL asignada por la entidad, una vez que el usuario ingresa al Sitio Web, lo primero que observa es la página principal del Sitio.



Luego de digitar y validar las credenciales de ingreso, se cargará la ventana siguiente:



Anexo 6. Código Patron Fachada

Operaciones de las clases

```
FachadaAD.cs
#region Métodos

#region Métodos Comentario

/// <summary>
/// Agrega un Comentario en la tabla correspondiente.
/// </summary>
/// <param name="_Comentario">Comentario a insertar</param>
/// <returns>Retorna el identificador del Comentario.</returns>
public long AgregarComentario(Comentario _Comentario)
{
    return m_ManejadorComentario.AgregarComentario(_Comentario);
}

/// <summary>
/// Actualiza un Comentario en la tabla correspondiente.
/// </summary>
/// <param name="_Comentario">Comentario a insertar</param>
public void ActualizarComentario(Comentario _Comentario)
{
    m_ManejadorComentario.ActualizarComentario(_Comentario);
}

/// <summary>
/// Lee un Comentario de la tabla correspondiente.
/// </summary>
/// <param name="_Comentariold">Identificador del Comentario que se quiere leer.</param>
/// <returns>Comentario leida.</returns>
public Comentario LeerComentario(long _Comentariold)
{
    return m_ManejadorComentario.LeerComentario(_Comentariold);
}

/// <summary>
/// Borra el Comentario de la base de datos.
/// </summary>
/// <param name="_Comentariold">Identificador del Comentario a borrar.</param>
public void EliminarComentario(long _Comentariold)
{
    m_ManejadorComentario.EliminarComentario(_Comentariold);
}

/// <summary>
/// Lee todos los comentarios de la base de datos.
/// </summary>
/// <returns>Lista de comentarios.</returns>
```

FachadaAD.cs

```
public IList<Comentario> LeerTodosComentarios()  
{  
    return m_ManejadorComentario.LeerTodosComentarios();  
}
```

```
#endregion Métodos Comentario
```

```
#region Métodos Control
```

```
/// <summary>  
/// Agrega un Control en la tabla correspondiente.  
/// </summary>  
/// <param name="_Control">Control a insertar</param>  
/// <returns>Retorna el identificador del Control.</returns>  
public long AgregarControl(Control _Control)  
{  
    return m_ManejadorControl.AgregarControl(_Control);  
}
```

```
/// <summary>  
/// Actualiza un Control en la tabla correspondiente.  
/// </summary>  
/// <param name="_Control">Control a insertar</param>  
public void ActualizarControl(Control _Control)  
{  
    m_ManejadorControl.ActualizarControl(_Control);  
}
```

```
/// <summary>  
/// Lee un Control de la base de datos.  
/// </summary>  
/// <param name="_ControlId">Identificador del Control que se quiere leer.</param>  
/// <returns>Control leído.</returns>  
public Control LeerControl(long _ControlId)  
{  
    return m_ManejadorControl.LeerControl(_ControlId);  
}
```

```
/// <summary>  
/// Borra el Control de la base de datos.  
/// </summary>  
/// <param name="_ControlId">Identificador del Control a borrar.</param>  
public void EliminarControl(long _ControlId)  
{  
    m_ManejadorControl.EliminarControl(_ControlId);  
}
```

```
/// <summary>  
/// Lee todos los controles de la base de datos.  
/// </summary>  
/// <returns>Lista de los controles leídos.</returns>  
public IList<Control> LeerTodosControles()  
{
```


FachadaAD.cs

```
        return m_ManejadorControl.LeerTodosControles();
    }

    /// <summary>
    /// Busca los controles por norma.
    /// </summary>
    /// <param name="_Normald">id de la norma a buscar.</param>
    /// <returns>Lista de controles.</returns>
    public IList<Control> BuscarControlesPorNorma(long _Normald)
    {
        return m_ManejadorControl.BuscarControlesPorNorma(_Normald);
    }

    /// <summary>
    /// Busca los controles por norma y estado de implementación.
    /// </summary>
    /// <param name="_Normald">Id de la norma.</param>
    /// <param name="_Implementado">Estado de implementación.</param>
    /// <returns>Lista de controles.</returns>
    public IList<Control> BuscarControlesPorNormaElImplementado(long _Normald, bool
    Implementado) // Nuevo +
    {
        return m_ManejadorControl.BuscarControlesPorNormaElImplementado(_Normald,
    Implementado);
    }

    /// <summary>
    /// Busca los controles por nivel.
    /// </summary>
    /// <param name="_Nivel">Nivel superior a buscar.</param>
    /// <returns>Lista de controles</returns>
    public IList<Control> BuscarControlesPorNivel(int _Nivel)
    {
        return m_ManejadorControl.BuscarControlesPorNivel(_Nivel);
    }

    /// <summary>
    /// Calcula el porcentaje de cumplimiento del control.
    /// </summary>
    public void CalcularPorcentajeControl()
    {
        m_ManejadorControl.CalcularPorcentajeControl();
    }

    #endregion

    #region Métodos Norma

    /// <summary>
    /// Agrega una norma en la tabla correspondiente.
    /// </summary>
    /// <param name="_Norma">Norma a insertar</param>
    /// <returns>Retorna el identificador de la norma.</returns>
```

FachadaAD.cs

```
public long AgregarNorma(Norma _Norma)
{
    return m_ManejadorNorma.AgregarNorma(_Norma);
}

/// <summary>
/// Actualiza una norma en la tabla correspondiente.
/// </summary>
/// <param name="_Norma">norma a insertar</param>
public void ActualizarNorma(Norma _Norma)
{
    m_ManejadorNorma.ActualizarNorma(_Norma);
}

/// <summary>
/// Lee una norma de la base de datos.
/// </summary>
/// <param name="_Normald">Identificador de la norma que se quiere leer.</param>
/// <returns>norma leida.</returns>
public Norma LeerNorma(long _Normald)
{
    return m_ManejadorNorma.LeerNorma(_Normald);
}

/// <summary>
/// Borra la norma de la base de datos.
/// </summary>
/// <param name="_Normald">Identificador de la norma a borrar.</param>
public void EliminarNorma(long _Normald)
{
    m_ManejadorNorma.EliminarNorma(_Normald);
}

/// <summary>
/// Retorna una colección con todas las normas por la propiedad id de forma ascendente.
/// </summary>
/// <returns>Lista de todas las normas ordenados por id.</returns>
public IList<Norma> LeerTodasNormas()
{
    return m_ManejadorNorma.LeerTodasNormas();
}

/// <summary>
/// Calcula el porcentaje de la norma.
/// </summary>
public void CalcularPorcentajeNorma()
{
    m_ManejadorNorma.CalcularPorcentajeNorma();
}

#endregion Métodos Norma

#region Métodos Observacion
```

FachadaAD.cs

```
/// <summary>
/// Agrega una Observacion en la tabla correspondiente.
/// </summary>
/// <param name="_ Observacion">Observacion a insertar</param>
/// <returns>Retorna el identificador de la Observacion.</returns>
public long AgregarObservacion(Observacion _Observacion)
{
    return m_ManejadorObservacion.AgregarObservacion(_Observacion);
}

/// <summary>
/// Actualiza una Observacion en la tabla correspondiente.
/// </summary>
/// <param name="_ Observacion">Observacion a insertar</param>
public void ActualizarObservacion(Observacion _Observacion)
{
    m_ManejadorObservacion.ActualizarObservacion(_Observacion);
}

/// <summary>
/// Lee una Observacion de la base de datos.
/// </summary>
/// <param name="_ observacionId">Identificador de la Observacion que se quiere leer.</param>
/// <returns>Observacion leida.</returns>
public Observacion LeerObservacion(long _observacionId)
{
    return m_ManejadorObservacion.LeerObservacion(_observacionId);
}

/// <summary>
/// Borra la Observacion de la base de datos.
/// </summary>
/// <param name="_ observacionId">Identificador de la Observacion a borrar.</param>
public void EliminarObservacion(long _observacionId)
{
    m_ManejadorObservacion.EliminarObservacion(_observacionId);
}

/// <summary>
/// Retorna una colección con todos las Observaciones por la propiedad id de forma ascendente.
/// </summary>
/// <returns>Lista de todos las Observaciones ordenadas por id.</returns>
public IList<Observacion> LeerTodasObservaciones()
{
    return m_ManejadorObservacion.LeerTodasObservaciones();
}

#endregion Métodos Observacion

#region Métodos Plan de Acción

/// <summary>
```

FachadaAD.cs

```
/// Agrega un PlanDeAccion en la tabla correspondiente.
/// </summary>
/// <param name="_PlanDeAccion">PlanDeAccion a insertar</param>
/// <returns>Retorna el identificador del PlanDeAccion.</returns>
public long AgregarPlanDeAccion(PlanDeAccion _PlanDeAccion)
{
    return m_ManejadorPlanDeAccion.AgregarPlanDeAccion(_PlanDeAccion);
}

/// <summary>
/// Agrega un PlanDeAccion en la tabla correspondiente.
/// </summary>
/// <param name="_PlanDeAccion">PlanDeAccion a insertar</param>
public void ActualizarPlanDeAccion(PlanDeAccion _PlanDeAccion)
{
    m_ManejadorPlanDeAccion.ActualizarPlanDeAccion(_PlanDeAccion);
}

/// <summary>
/// Lee un PlanDeAccion de la base de datos.
/// </summary>
/// <param name="_PlanDeAccionId">Identificador del PlanDeAccion que se quiere leer.</param>
/// <returns>PlanDeAccion leida.</returns>
public PlanDeAccion LeerPlanDeAccion(long _PlanDeAccionId)
{
    return m_ManejadorPlanDeAccion.LeerPlanDeAccion(_PlanDeAccionId);
}

/// <summary>
/// Borra el PlanDeAccion de la base de datos.
/// </summary>
/// <param name="_PlanDeAccionId">Identificador del PlanDeAccion a borrar.</param>
public void EliminarPlanDeAccion(long _PlanDeAccionId)
{
    m_ManejadorPlanDeAccion.EliminarPlanDeAccion(_PlanDeAccionId);
}

/// <summary>
/// Retorna una colección con todos los PlanDeAccion por la propiedad id de forma ascendente.
/// </summary>
/// <returns>Lista de todos los PlanDeAccion ordenados por id.</returns>
public IList<PlanDeAccion> LeerTodosPlanesDeAccion()
{
    return m_ManejadorPlanDeAccion.LeerTodosPlanesDeAccion();
}

#endregion Métodos Plan de Acción

#region Usuario

/// <summary>
/// Inserta un usuario a la tabla usuario.
/// </summary>
```

FachadaAD.cs

```
/// <param name="_Usuario">Usuario a insertar.</param>
/// <returns>Retorna el identificador del usuario.</returns>
public long AgregarUsuario(Usuario _Usuario)
{
    return m_ManejadorUsuario.AgregarUsuario(_Usuario);
}

/// <summary>
/// Actualiza un usuario en la tabla usuario.
/// </summary>
/// <param name="_Usuario">Usuario de tipo usuario a actualizar.</param>
public void ActualizarUsuario(Usuario _Usuario)
{
    m_ManejadorUsuario.ActualizarUsuario(_Usuario);
}

/// <summary>
/// Lee un usuario de la base de datos.
/// </summary>
/// <param name="_UsuarioId">Identificador del usuario que se quiere leer.</param>
/// <returns>Usuario cargado de la base de datos.</returns>
public Usuario LeerUsuario(long _UsuarioId)
{
    return m_ManejadorUsuario.LeerUsuario(_UsuarioId);
}

/// <summary>
/// Lee un usuario de la base de datos
/// </summary>
/// <param name="_NombreIngreso">Nombre de ingreso del usuario</param>
/// <returns>Usuario cargado de la base de datos</returns>
public Usuario LeerUsuarioPorNombreIngreso(string _NombreIngreso)
{
    return m_ManejadorUsuario.LeerUsuarioPorNombreIngreso(_NombreIngreso);
}

/// <summary>
/// Borra un usuario de la base de datos.
/// </summary>
/// <param name="_UsuarioId">Identificador del usuario a borrar.</param>
public void EliminarUsuario(long _UsuarioId)
{
    m_ManejadorUsuario.EliminarUsuario(_UsuarioId);
}

}

/// <summary>
/// Devuelve una colección con todos los usuarios ordenados por la propiedad nombre en forma
ascendente.
/// </summary>
/// <returns>Lista de todos los usuarios ordenados por nombre.</returns>
public IList<Usuario> LeerTodosUsuarios()
{
}
```

FachadaAD.cs

```
        return m_ManejadorUsuario.LeerTodosUsuarios();
    }

    /// <summary>
    /// Retorna una colección con los usuarios que estén dentro de los límites establecidos.
    /// </summary>
    /// <param name="_Cuantos">Número máximo de resultados.</param>
    /// <param name="_Desde">Primera fila para recibir.</param>
    /// <returns>Retorna la lista de usuarios entre los parámetros establecidos.</returns>
    public IList<Usuario> LeerUsuarios(int _Cuantos, int _Desde)
    {
        return m_ManejadorUsuario.LeerUsuarios(_Cuantos, _Desde);
    }

    /// <summary>
    /// Devuelve una colección con todos los usuarios activos ordenados por nombre de forma
    ascendente.
    /// </summary>
    /// <returns>Lista de usuarios activos por nombre.</returns>
    public IList<Usuario> BuscarUsuariosActivos()
    {
        return m_ManejadorUsuario.BuscarUsuariosActivos();
    }

    /// <summary>
    /// Devuelve una colección con todos los Usuarios inactivos ordenados por la propiedad nombre
    de forma ascendente.
    /// </summary>
    /// <returns>Lista de Usuarios inactivos ordenada por nombre.</returns>
    public IList<Usuario> BuscarUsuariosInactivos()
    {
        return m_ManejadorUsuario.BuscarUsuariosInactivos();
    }

    /// <summary>
    /// Devuelve una colección con los Usuarios activos que están en los límites establecidos.
    /// </summary>
    /// <param name="_Desde">Primera fila para recibir.</param>
    /// <param name="_Cuantos">Número máximo de resultados.</param>
    /// <returns>Retorna la lista de Usuarios entre los parámetros recibidos.</returns>
    public IList<Usuario> BuscarUsuariosActivos(int _Desde, int _Cuantos)
    {
        return m_ManejadorUsuario.BuscarUsuariosActivos();
    }

    /// <summary>
    /// Devuelve una colección con los Usuarios inactivos que están en los límites establecidos.
    /// </summary>
    /// <param name="_Desde">Primera fila para recibir.</param>
    /// <param name="_Cuantos">Número máximo de resultados.</param>
    /// <returns>Retorna la lista de Usuarios entre los parámetros recibidos.</returns>
    public IList<Usuario> BuscarUsuariosInactivos(int _Desde, int _Cuantos)
    {
    }
}
```

FachadaAD.cs

```
        return m_ManejadorUsuario.BuscarUsuariosInactivos();
    }

    /// <summary>
    /// Busca los Usuarios por nombre y estado (activo/inactivo).
    /// </summary>
    /// <param name="_Nombre">Nombre del Usuario. Busca fragmentos de palabra. En caso de no
    querer.
    /// usar este filtro, se debe enviar <code>null</code>.</param>
    /// <param name="_Activo">Indica si el Usuario está activo o no. En caso de no querer.
    /// usar este filtro, se debe enviar <code>null</code>.</param>
    /// <returns></returns>
    public IList<Usuario> BuscarUsuariosPorNombreYEstado(string _Nombre, bool? _Activo) //
    Nuevo +
    {
        return m_ManejadorUsuario.BuscarUsuariosPorNombreYEstado(_Nombre, _Activo);
    }

    /// <summary>
    /// Busca los Usuarios por nombre y estado (activo/inactivo).
    /// </summary>
    /// <param name="_Nombre">Nombre del Usuario. Busca fragmentos de palabra. En caso de no
    querer.
    /// usar este filtro, se debe enviar <code>null</code>.</param>
    /// <param name="_Activo">Indica si el Usuario está activo o no. En caso de no querer.
    /// <param name="_Desde">Primera fila para recibir.</param>
    /// <param name="_Cuantos">Número máximo de resultados.</param>
    /// usar este filtro, se debe enviar <code>null</code>.</param>
    /// <returns></returns>
    public IList<Usuario> BuscarUsuariosPorNombreYEstado(string _Nombre, bool? _Activo, int
    Desde, int _Cuantos) // Nuevo +
    {
        return m_ManejadorUsuario.BuscarUsuariosPorNombreYEstado(_Nombre, _Activo, _Cuantos,
    Desde);
    }

    /// <summary>
    /// Busca los usuarios que tengan cierto fragmento de palabra en el nombre de ingreso.
    /// </summary>
    /// <param name="_NombreIngreso">Nombre ingreso(Completo o fragmento) a buscar.</param>
    /// <returns>Lista de usuarios.</returns>
    public IList<Usuario> BuscarUsuariosPorNombreIngreso(string _NombreIngreso)
    {
        return m_ManejadorUsuario.BuscarUsuariosPorNombreIngreso(_NombreIngreso);
    }

    /// <summary>
    /// Devuelve la cantidad de usuarios que existen.
    /// </summary>
    /// <returns>La cantidad de usuarios.</returns>
    public long ContarTodosUsuarios()
    {
        return m_ManejadorUsuario.ContarTodosUsuarios();
    }
}
```

FachadaAD.cs

```
}

/// <summary>
/// Devuelve la cantidad de Usuarios activos.
/// </summary>
/// <returns>Cantidad de Usuarios activos</returns>
public long ContarUsuariosActivos()
{
    return m_ManejadorUsuario.ContarUsuariosActivos();
}

/// <summary>
/// Devuelve la cantidad de Usuarios inactivos.
/// </summary>
/// <returns>Cantidad de Usuarios inactivos</returns>
public long ContarUsuariosInactivos()
{
    return m_ManejadorUsuario.ContarUsuariosInactivos();
}

/// <summary>
/// Cuenta los usuarios por nombre y estado(activo/inactivo).
/// </summary>
/// <param name="_Nombre">Nombre del usuario a contar.</param>
/// <param name="_Activo">Estado del usuario(activo/inactivo).</param>
/// <returns>Cantidad de usuarios.</returns>
public long ContarUsuariosPorNombreYEstado(string _Nombre, bool? _Activo) // Nuevo
{
    return m_ManejadorUsuario.ContarUsuariosPorNombreYEstado(_Nombre, _Activo);
}

#endregion

#region Rol

/// <summary>
/// Devuelve una colección con todos los usuarios ordenados por la propiedad nombre en forma
ascendente.
/// </summary>
/// <returns>Lista de todos los usuarios ordenados por nombre.</returns>
public IList<Rol> LeerTodosRoles()
{
    return m_ManejadorRol.LeerTodosRoles();
}

/// <summary>
/// Lee un usuario de la base de datos.
/// </summary>
/// <param name="_Usuariold">Identificador del usuario que se quiere leer.</param>
/// <returns>Usuario cargado de la base de datos.</returns>
public Rol LeerRol(long _RolId)
{
}
```


FachadaAD.cs

```
return m_ManejadorRol.LeerRol(_RolId);  
}
```

```
#endregion
```

```
#endregion Métodos
```