

ESCUELA POLITÉCNICA DEL EJÉRCITO

DPTO. DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
AUTOMATIZADO DE DIGITALIZACIÓN DE
DOCUMENTOS (SADO) PARA SOLUCIONES
INTELIGENTES**

Previa a la obtención del Título de:

INGENIERO DE SISTEMAS E INFORMÁTICA

**POR: GABRIELA REBECA AGUILAR BAQUERO
OMAR CRISTÓBAL ARBOLEDA TERÁN**

SANGOLQUÍ, Septiembre del 2011

DECLARACIÓN

Nosotros, Gabriela Rebeca Aguilar Baquero y Omar Cristóbal Arboleda Terán, declaramos que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que se ha consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica del Ejército (ESPE), puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Gabriela Rebeca Aguilar Baquero

Omar Cristóbal Arboleda Terán

CERTIFICACIÓN

Certifico que el presente trabajo, fue realizado en su totalidad por los Srs. GABRIELA REBECA AGUILAR BAQUERO y OMAR CRISTÓBAL ARBOLEDA TERÁN como requerimiento parcial a la obtención del Título de INGENIERO DE SISTEMAS E INFORMÁTICA, bajo mi supervisión.

Sangolquí, Septiembre del 2011

ING.MAURICIO LOACHAMIN

DEDICATORIA

Dedico este trabajo a mis Padres *Luis Aguilar e Irene Baquero* y mis hermanos *Luis, Belén* quienes con sus consejos, confianza, ayuda y amor incondicional me han permitido crecer como mujer y ser una profesional excelente.

A mi esposo *Diego Fuentes* que siempre ha creído en mí y me ha brindado apoyo, amor y fuerza para alcanzar mis objetivos.

A mi hija *María Paz Fuentes* que cada día me inspira a luchar, y ser constante para cumplir todos los objetivos que me permitan brindarle un futuro mejor.

“Para todo hay tiempo en esta vida”

Gabriela Rebeca Aguilar Baquero

DEDICATORIA

Dedico este trabajo a mi Madre *Rina Elizabeth “Angelitos” Terán Morales* y a mi novia *Soffa Flores Albuja* quienes con sus confianza, ayuda, amor e **“INSISTENCIA”** me han permitido crecer como persona y como profesional.

“Solo dame una oportunidad, que el resto lo ganaré por mi esfuerzo”

Omar Cristóbal Arboleda Terán

AGRADECIMIENTO

Le doy gracias a mi Dios por permitirme vivir cada día y ser la persona que soy y por ayudarme en todos los momentos de mi vida.

A mis padres por darme la oportunidad de recibir una Educación Superior, por su inmenso amor, entereza, cuidado, dedicación, amistad y consejo diario.

A mi esposo e hija por brindarme su amor, entereza, comprensión, alegría y que me dan animo y fuerza para seguir cumpliendo mis metas y anhelos.

A mi compañero de tesis, que siempre ha estado ahí para lograr juntos el gran anhelo de convertirnos en los nuevos Ingenieros de la Patria.

A mi Director y Codirector de tesis quienes con su experiencia y profesionalismo supieron encaminar de la mejor manera esta obra, y a todas las personas que de una u otra manera colaboraron en su culminación.

Gabriela Rebeca Aguilar Baquero

AGRADECIMIENTO

Quiero Agradecer a mi madre y hermano por apoyar de manera incondicional este sueño, respetando mis decisiones, inculcándome valores éticos y morales; los cuales se han plasmado en mi personalidad, y en la práctica de mis hechos.

A mi novia que gracias a su empuje pude culminar este proyecto, esta etapa de mi vida, y que sin su apoyo incondicional no lo hubiera cumplido.

A mis Familiares y Amigos que a lo largo de este tiempo han sido un apoyo emocional, encontrado siempre una mano en la cual apoyarme en los momentos difíciles.

.

Omar Cristóbal Arboleda Terán

ÍNDICE DE CONTENIDOS

CAPÍTULO I	18
1 GENERALIDADES.....	18
1.1 Introducción	18
1.2 Justificación	19
1.3 Objetivos.....	21
1.3.1 Objetivo General.....	21
1.3.2 Objetivos Específicos	21
1.4 Alcance.....	22
1.5 Herramientas	24
CAPÍTULO II	28
2 MARCO TEÓRICO	28
2.1 Sistema de Digitalización de Documentos.....	28
2.1.1 ¿Qué son Sistemas Digitales?	28
2.1.2 Características de un Sistema Digital.....	28
2.1.3 Características de un Documento Digital	29
2.1.3.1 Características Funcionales	29
2.1.3.2 Mutabilidad e Identidad en los Documentos Digitales	31
2.1.3.3 Reconocimiento de Caracteres Ópticos(OCR).....	32
2.1.4 Usos de Sistemas de Digitalización de Documentos.....	33
2.1.5 Ventajas y Desventajas de los Sistemas Digitales de Documentos, Digitalización de Documentos y Archivos de Papel	34
2.1.5.1 Ventajas Sistema Digital	34
2.1.5.2 Desventajas Sistema Digital.....	34
2.1.5.3 Ventajas Digitalización de documentos.....	35
2.1.5.4 Desventajas del archivo en papel.....	36
2.2 Metodología para el desarrollo de software.....	36
2.2.1 Metodología de Diseño de Hipermmedia Orientada a Objetos (OOHDM).....	36
2.2.1.1 Definición.....	36
2.2.1.2 Características Fundamentales.....	37
2.2.1.3 Etapas ó Fases	38
2.2.1.4 Definición de Requerimientos.....	39
2.2.1.5 Ventajas y Desventajas.....	49
2.2.1.6 Criterios de Selección de OOHDM.....	51
2.2.2 Lenguaje Unificado De Modelado (UML).....	51
2.2.2.1 Definición.....	51
2.2.2.2 Diagramas	52
2.2.2.3 Diagramas de Caso de Uso	52
2.2.2.4 Diagramas de Secuencia	53
2.2.2.5 Diagramas de Colaboración	55
2.2.2.6 Diagramas de Objeto	56
2.2.2.7 Diagramas de Clase.....	57
2.2.2.8 Diagramas de Estado.....	61

2.2.2.9	Diagramas de Actividad	62
2.2.2.10	Diagramas de Despliegue	64
2.2.2.11	Diagramas de Componente.....	65
2.2.3	Metodología OOADM y el Lenguaje de Modelado UML.....	66
2.3	Indexación de Documentos	67
2.3.1.1	Características principales de los Subsistemas de Indexación de Documentos	68
2.4	LDAP	69
2.4.1.1	Origen de Influencias	70
2.4.1.2	Visión general del protocolo	71
2.4.1.3	Estructura de directorio	73
2.4.1.4	URLs de LDAP	74
2.4.1.5	Variantes	75
2.4.1.6	Otros modelos de datos	76
2.4.1.7	Uso.....	76
2.4.1.8	Implementaciones	77
2.5	Active Directory	80
2.5.1.1	Estructura.....	80
2.5.1.2	Funcionamiento.....	81
2.5.1.3	Intercambio entre dominios	82
2.5.1.4	Confianza transitiva.....	82
2.5.1.5	Direccionamientos a recursos	83
2.5.1.6	Diferencias entre Windows NT y Active Directory	84
2.5.1.7	Interfaces de programación.....	84
2.5.1.8	Requisitos de instalación.....	85
2.5.1.9	Alternativas.....	85
2.6	Información general y conceptual sobre .NET Framework.....	86
2.6.1.1	Características de Common Language Runtime.....	89
2.6.1.2	Biblioteca de clases de .NET Framework.....	91
2.6.1.3	Common Language Runtime.....	91
2.6.1.4	C# y .NET Framework.....	94
2.6.1.5	Lenguaje C#.....	95
2.6.1.6	Arquitectura de la plataforma .NET Framework	96
CAPÍTULO III		99
3	DESARROLLO DEL SISTEMA SADO	99
3.1	Introducción.....	99
3.2	Antecedentes.....	99
3.2	Análisis de la Situación Actual	99
3.3	Estrategias nivel de plan de negocio	100
3.3.1	Análisis FODA	100
3.3.2	Matriz de Resumen FODA	102
3.3.3	Cruces FODA	104
3.3.4	Estrategias	107
3.3.4.1	Estrategias competitivas Actuales.....	107
3.3.4.2	Estrategias Funcionales	109
3.4	Formulación y Análisis.....	109
3.5	Especificación de Requerimientos.....	110
3.5.1	Introducción.....	111
3.5.1.1	Propósito	111

3.5.2	Identificación de Roles y Tareas.....	111
3.5.3	Especificación de Escenarios.....	114
3.5.4	Especificación de casos de uso por Actor	116
3.6	Requerimientos No Funcionales.....	126
3.7	Diagramas de Secuencia.....	126
3.8	Diseño Conceptual	127
3.8.1	Diseño de Base de Datos.....	127
3.8.2	Diagramas de Clase.....	128
3.9	Diseño Navegacional.....	136
3.9.1	Esquema de Contextos Navegacionales.....	136
3.10	Diseño de Interfaz Abstracta.....	139
3.10.1	Vista de Datos Abstractos.....	140
3.11	Diagramas de Estado.....	143
3.12	Diseño Estético	144
3.12.1	Características de la Plantilla.....	144
3.13	Consideraciones de Diseño Gráfico.....	144
3.14	Diagramas de Despliegue.....	145
3.15	Arquitectura.....	145
CAPÍTULO IV		146
4	IMPLEMENTACIÓN SADO.....	146
4.1	Introducción	146
4.2	Proceso de Instalación	146
4.2.1	Proceso de Instalación y Configuración de Aplicaciones	146
4.3	Implementación del Adobe Capture en el Servidor	147
4.3.1	Configuración del Adobe Capture.....	150
4.4	Active Directory	155
4.4.1	Instalación de Active Directory	155
4.4.2	Configuration Active Directory	164
4.4.3	Creación de Grupos	165
4.4.4	Creación de Grupos de Usuarios	167
4.4.5	Creación de Usuarios.....	169
4.5	Print Server.....	173
4.5.1	Instalación y Configuración de Print Server.....	173
4.6	File Server	175
4.6.1	Implementación del FileServer	175
4.6.2	Configuración de File Server	179
4.7	Base de Datos	183
4.7.1	Instalación de la Base de Datos	183
4.7.2	Configuración de la Base de Datos	191
4.8	Implementación de SADO	195
4.8.1	SADO LDAP Migrator.....	195
4.8.2	SADO	196
4.8.3	Administración de Documentos.....	198
4.8.4	Consulta de Documentos	199
4.8.5	SADO Vigilante.....	201
CAPÍTULO V		204
CONCLUSIONES Y RECOMENDACIONES		204
5.1	Conclusiones	204

5.2 Recomendaciones	205
BIBLIOGRAFÍA	207
GLOSARIO DE TÉRMINOS.....	209
ANEXOS	211

ÍNDICE DE TABLAS

Tabla 1.1	Servidor de aplicaciones	25
Tabla 1.2	Servicios.....	25
Tabla 1.3	Para la Programación.....	26
Tabla 1.4	Capa de Base de Datos	27
Tabla 2.1	Etapas de la Metodología OOHDm	39
Tabla 3.1	Análisis de FODA	101
Tabla 3.2	Matriz de Resumen de Fortalezas.....	102
Tabla 3.3	Matriz de Resumen de Oportunidades	102
Tabla 3.4	Matriz de Resumen de Debilidades.....	103
Tabla 3.5	Matriz de Resumen de Amenazas.....	103
Tabla 3.6	Matriz de Resumen de Fortalezas contra Oportunidades	104
Tabla 3.7	Matriz de Resumen de Debilidades contra Amenazas	105
Tabla 3.8	Matriz de Resumen de Fortalezas contra Amenazas	106
Tabla 3.9	Matriz de Resumen de Debilidades contra Amenazas	107
Tabla 3.10	Gestionar Usuarios.....	118
Tabla 3.11	Gestionar Seguridades.....	119
Tabla 3.12	Gestionar Administración de Documentos	120
Tabla 3.13	Gestionar Histórico de Documentos	121
Tabla 3.14	Gestionar Reportes	122
Tabla 3.15	Utilizar Servicio de Históricos	123
Tabla 3.16	Utilizar Servicio Administración de Documentos	124
Tabla 3.17	Utiliza el Servicio de Históricos	125

ÍNDICE DE FIGURAS

Figura 2.1	Esquema del Diseño Navegacional.....	45
Figura 2.2	Ejemplo de un Caso de Uso.....	53
Figura 2.3	Ejemplo de un Diagrama de Secuencia	54
Figura 2.4	Ejemplo de un Diagrama de Colaboración.....	55
Figura 2.5	Ejemplo de Diagrama de Objetos.....	57
Figura 2.6	Ejemplo de una Clase	58
Figura 2.7	Ejemplo de un Diagrama de Clases	60
Figura 2.8	Ejemplo de un Diagrama de Estado.....	62
Figura 2.9	Ejemplo de un Diagrama de Actividades.....	63
Figura 2.10	Ejemplo de un Diagrama de Despliegue	64
Figura 2.11	Ejemplo de un Diagrama de Componentes.....	66
Figura 2.12	OHDM y UML.....	67
Figura 2.13	Indexación.....	69
Figura 2.14	Componentes .Net Framework.....	88
Figura 2.15	Ejecución de C#	98
Figura 3.1	Actores - SADO.....	112
Figura 3.2	Casos de Uso para Administrador SADO	117
Figura 3.3	Casos de Uso para Editor SADO	123
Figura 3.4	Casos de Uso para Usuario SADO	125
Figura 3.5	Diagrama de Base de Datos SADODB	127
Figura 3.6	Diseño Conceptual – Componente de Archivo.....	128
Figura 3.7	Diseño Conceptual – Componente de Perfiles.....	128
Figura 3.8	Diseño Conceptual – Componente.....	129
Figura 3.9	Diseño Conceptual – Grupos	129
Figura 3.10	Diseño Conceptual – Usuario por Grupos.....	130
Figura 3.11	Diseño Conceptual – Programas.....	130
Figura 3.12	Diseño Conceptual – Menú	131
Figura 3.13	Diseño Conceptual – Tipo Documento.....	131
Figura 3.14	Diseño Conceptual – Documentos.....	132
Figura 3.15	Diseño Conceptual – Solicitudes.....	132
Figura 3.16	Diseño Conceptual – PlugIn	133
Figura 3.17	Diseño Conceptual –FrmMain.....	133
Figura 3.18	Diseño Conceptual – SadoDB.....	134

Figura 3.19	Diseño Conceptual – IPlugin	134
Figura 3.20	Diseño Conceptual – IPluginHost.....	135
Figura 3.21	Diseño Conceptual – IPlugin	135
Figura 3.22	Esquema de Contexto - Menú Principal	137
Figura 3.23	Esquema de Contexto – Menú Principal –Administración.....	137
Figura 3.24	Esquema de Contexto – Menú Principal – Consultas	138
Figura 3.25	Esquema de Contexto – Menu Horizontal.....	138
Figura 3.26	Esquema de Contexto – Enlaces Por Actor	139
Figura 3.27	Vista Abstracta del Nodo Aplicaciones.....	140
Figura 3.28	Vista Abstracta del Nodo Cabecera	140
Figura 3.29	Vista Abstracta del Nodo Menú Horizontal Administrador.....	141
Figura 3.30	Vista Abstracta del Nodo Menú del Editor.....	141
Figura 3.31	Vista Abstracta del Nodo Menú del Usuario.....	142
Figura 3.32	Vista Abstracta del Nodo Contenido.....	142
Figura 3.33	Diagrama de Estado – SADO	143
Figura 3.34	Diseño Estético – Organización de los Elementos.....	144
Figura 3.35	Diagrama de despliegue	145
Figura 3.36	Diagrama de Arquitectura	145
Figura 4.1	Imagen de Bienvenida a la Instalación.....	147
Figura 4.2	Imagen de Inicio de Instalación.....	148
Figura 4.3	Imagen de Selección de Idioma de Instalación.....	148
Figura 4.4	Imagen de Términos de Licencia.....	149
Figura 4.5	Imagen de Selección de Ubicación de la Instalación.....	149
Figura 4.6	Imagen de confirmación de la Instalación.....	150
Figura 4.7	Imagen de Configuración de Adobe Capture.....	151
Figura 4.8	Imagen de creación de trabajo e ingreso de pasos a seguir.....	151
Figura 4.9	Imagen de Capturar de Imagen.....	152
Figura 4.10	Imagen de Mejora de Imagen	152
Figura 4.11	Imagen de OCR	152
Figura 4.12	Imagen de Exportar a PDF.....	153
Figura 4.13	Imagen de Guardado en la Unidad de red.....	153
Figura 4.14	Imagen de Propiedades de Guardado en Disco.....	153
Figura 4.15	Imagen de Revision de las Propiedades de los Trabajos.....	154
Figura 4.16	Imagen de WorkGroup de Acrobat Capture	155
Figura 4.17	Imagen de estado de la instalación.....	156

Figura 4.18 Imagen de Resultado de Instalación.	156
Figura 4.19 Imagen de Bienvenida al Wizard de Active Directory.....	157
Figura 4.20 Imagen de creación de Nuevo Bosque de Dominio.	158
Figura 4.21 Imagen de nombre de dominio.....	158
Figura 4.22 Imagen de Estado de la validación del nuevo dominio.....	159
Figura 4.23 Imagen de Selección de Versión de Active Directory.....	159
Figura 4.24 Imagen de recomendaciones adicionales de roles para servidor	160
Figura 4.25 Imagen de Advertencia de IP Dinámica	161
Figura 4.26 Imagen de Selección de Ubicación de Almacenamiento.....	161
Figura 4.27 Imagen de Ingreso de Password de Administrador.....	162
Figura 4.28 Imagen de Resumen de configuraciones	162
Figura 4.29 Imagen de estado de la configuración.....	163
Figura 4.30 Imagen de Finalización de la instalación.....	164
Figura 4.31 Imagen de Sugerencia de Reiniciar el Servidor.	164
Figura 4.32 Imágenes de creación de las unidades Organizacionales Generales (Grupos).....	165
Figura 4.33 Imágenes de creación de las unidades Organizacionales Generales (Usuarios).....	165
Figura 4.34 Imagen del Active Directory Users and Computers de nuestra organización.....	166
Figura 4.35 Imagen de Creación del Grupo Global.	166
Figura 4.36 Imagen de la Organización actual dentro del Directorio.....	167
Figura 4.37 Imagen de Creación de Grupo para el Usuario.....	168
Figura 4.38 Imagen de Búsqueda de Usuario.....	168
Figura 4.39 Imagen del Grupo para el Usuario.	169
Figura 4.40 Imagen de Creación de Usuario.....	170
Figura 4.41 Imagen de Asignación de Password.	170
Figura 4.42 Imagen de Finalización de la creación de Usuario.....	170
Figura 4.43 Imagen de Grupos a los que pertenece el Usuario.	171
Figura 4.44 Imagen de creación del Grupo del Usuario.	171
Figura 4.45 Imagen final del Grupo del Usuario con su respectivo usuario asignado.....	172
Figura 4.46 Imagen de Jerarquía desde el Grupo Global.	172
Figura 4.47 Imagen de Bienvenida a la Instalación del Nuevo Rol.	173
Figura 4.48 Imagen de Selección de Funciones del Rol.	174
Figura 4.49 Imagen de Confirmación de la Instalación.	174

Figura 4.50 Imagen de Progreso de Instalación Print Server	175
Figura 4.51 Imagen de selección de Nuevo Rol de File Service.	175
Figura 4.52 Imagen de Selección de Funciones del File Server.	176
Figura 4.53 Imagen de Configuración de Unidad de Storage del FileServer..	177
Figura 4.54 Imagen de Configuración de Reportes sobre el FileServer.....	177
Figura 4.55 Imagen de Configuración de Indexación de Unidades.	178
Figura 4.56 Imagen de Resumen de la Instalación del File Server.	178
Figura 4.57 Imagen de Resumen de la Instalación.	179
Figura 4.58 Carpeta dentro del almacenamiento.	180
Figura 4.59 Opciones de Compartido de la Carpeta.	180
Figura 4.60 Configuración del recurso compartido.....	181
Figura 4.61 Permisos dentro del Dominio.	181
Figura 4.62 Ubicación del Recurso	182
Figura 4.63 Parámetros de almacenamiento.....	182
Figura 4.64 Imagen del File Server Resource Manager con el Recursos Monitoreados.....	183
Figura 4.65 Imagen de Instalación Nueva	183
Figura 4.66 Imagen de Chequeo de Prerrequisitos.....	184
Figura 4.67 Imagen de Key de Instalación.	184
Figura 4.68 Imagen de Aceptación de Licencia.....	185
Figura 4.69 Imagen de Instalación de archivos de ayuda.	185
Figura 4.70 Imagen de chequeo de prerrequisitos de instalación.	186
Figura 4.71 Imagen de selección del tipo de instalación.	186
Figura 4.72 Imagen de Selección de Componentes a instalar.	187
Figura 4.73 Imagen de la selección de la instancia y de la ubicación de la instalación.	188
Figura 4.74 Imagen de Selección de Cuentas para Servicios.	188
Figura 4.75 Imagen de selección de Tipo de autenticación y Administrador..	189
Figura 4.76 Imagen de Revisión de la Configuración de la Instalación.	190
Figura 4.77 Imagen de Resumen de la Instalación.	190
Figura 4.78 Imagen de Ventana Principal del SQL Management Studio.	191
Figura 4.79 Imagen de creación de la base de datos.....	192
Figura 4.80 Imagen de Opciones de creación de la Base de datos.	192
Figura 4.81 Imagen de la vista del Script de estructura y programabilidad de SADO.....	193
Figura 4.82 Imagen de Ejecución del Script.....	194

Figura 4.83 Imagen de Tablas de la Base de Datos	194
Figura 4.84 Imagen de vista principal del SADO LDAP Migrator	196
Figura 4.85 Imagen de Inicio del Sistema.	196
Figura 4.86 Imagen de la Vista Principal del Sistema.	197
Figura 4.87 Imagen del módulo de Administración de Documentos.	198
Figura 4.88 Imagen del Módulo de Consulta de documentos.	200
Figura 4.89 Imagen del sub-módulo de búsqueda de documentos.....	200
Figura 4.90 Imagen de Vista del Servicio SADO Vigilante.	201

CAPÍTULO I

GENERALIDADES

1.1 Introducción

El Sistema Automatizado de Digitalización de Documentos (SADO) ayudará a simplificar el almacenamiento de varios documentos que se encuentran en forma física, los cuales pueden llegar a extraviarse, a ser alterado, SADO propone garantizar la fiabilidad en cuanto a la existencia de la misma. Se utilizará el Producto Adobe Capture, con esto se busca mejorar la productividad a través de la optimización del tiempo de administración de los documentos logrando así garantizar el almacenamiento y organización de la información en formato digital utilizando OCR (Optical character recognition) que Adobe Capture facilita.

El valor que la información representa actualmente para las empresas, negocios, escuelas, etc. Se convierte en unos de los factores más importantes que se deben resguardar por lo cual surgen las necesidades de gestionar, organizar y asegurar la misma por lo cual se busca obtener una herramienta que resuelva este tipo de necesidades y sobre todo que el usuario mantenga la iteración y el control a la información que el maneja y permitirle que esta información se guarde en diferentes formatos y subsiguiente darle un formato más general independiente de plataformas y aplicaciones. El archivo puede ser enviado utilizando correo electrónico, también se

utilizará la indexación para obtener una búsqueda más adecuada y el versionamiento que brindará apoyo a la empresa en su proceso de auditoría, y ofrecerá un mejor control de histórico de dicho documento.

SADO además de brindar la administración de documentos, también garantizará principales características como:

- Implementación Escalable
- Seguro
- Inicio de sesión único
- Tolerancia a Fallas
- Control de Usuarios

Por lo cual se tiene la certeza de que el producto puede satisfacer las necesidades de los usuarios de la empresa Soluciones Inteligentes.

1.2 Justificación

Debido al constante cambio de la tecnología, tanto el hardware como el software se devalúan rápidamente produciendo que las empresas, universidades e instituciones públicas y privadas se vean obligadas a mantenerse a la vanguardia de la misma.

Por ello, el presente plan propone el Desarrollo del Sistema Automatizado de Digitalización de Documentos (SADO), lo que permitirá la simplificación en el almacenamiento de distintos tipos de documentos, evitando la pérdida de los mismos debido a problemas o eventualidades externas los que causan pérdidas a las empresas.

El propósito es utilizar como base el Producto Adobe Capture y desarrollar un sistema que cuente con la administración de documentos y conversión en su parte operativa, la expansión de la aplicación se tiene prevista mediante la implementación de componentes adicionales, y así lograr un producto de calidad y fiabilidad.

Se utilizará la metodología OOHDM para el análisis, diseño e implementación de la Aplicación, debido a que permitirá visualizar y alinear nuestra aplicación con los objetivos institucionales requeridos.

Después el Capture facilitará el OCR y la digitalización de contenidos de SADO publicándolos de manera muy fácil dentro de nuestra red; logrando así crear una aplicación realmente funcional tanto en el front-end como en el back-end.

Se usará Windows Server 2008 por que se incrementa la flexibilidad y la confiabilidad en la infraestructura del servidor, a la vez que ofrece a los desarrolladores una plataforma de aplicaciones y una Web más robustas para la creación de aplicaciones y servicios conectados.

SQL Server 2008 no brinda una plataforma confiable y fácilmente adaptable a los cambios que se producen constantemente en el mercado además reduce las vulnerabilidades de datos

Adobe Capture es una herramienta eficaz que va a permitir convertir grandes volúmenes de documentos en el formato deseado, ofreciendo así producto profesional ya que por su potente reconocimiento óptico de caracteres convierte a documentos electrónicos de alta calidad y fiabilidad la empresa Soluciones Digitales Microsoft Visual Studio.NET 2008 permite crear rápidamente aplicaciones con la más alta calidad, ofrece herramientas que ayuda con la integración entre en bases de datos y diseñadores gráficos con atractivas implementaciones para el usuario.

1.3 Objetivos

1.3.1 Objetivo General

Realizar el Análisis, Diseño e Implementación del Sistema Automatizado de Digitalización de Documentos (SADO) a través de la Implementación de un sistema basado en Adobe Captura y componentes, que permitan brindar servicios requeridos por los clientes internos.

1.3.2 Objetivos Específicos

- Realizar un análisis FODA para el desarrollo del SADO con el fin de evaluar la situación actual de la empresa y establecer hacia donde se quiere llegar con la aplicación.

- Estudiar la Arquitectura del Adobe Capture para determinar e implementar servicios complementarios en la construcción de la aplicación.
- Desarrollar el sistema y automatización de la Aplicación montándonos como base en el Adobe Capture con los nuevos servicios, con el fin de facilitar la labor de digitalización de Documentos usando plantillas diseñadas previamente.
- Implementar SADO con administración de plantillas y documentos con el propósito de brindar búsquedas óptimas, versionamientos, controles de acceso, garantizando la organización de los mismos.

1.4 Alcance

El presente plan tiene por objeto el Desarrollo del Sistema Automatizado de Digitalización de Documentos (SADO) mediante la utilización del Adobe Capture para la digitalización de Documentos.

Se ha considerado dividir su creación en 2 fases:

La primera se refiere al **análisis de requerimientos**, los cuales serán organizados de acuerdo a la metodología ¡OOHDM!.

La segunda se refiere al ***Diseño, Desarrollo e Implementación del SADO***, dentro del cual se pretende cubrir aspectos tales como gestión, operabilidad, usabilidad.

El Diseño se refiere a:

- Utilización de la metodología OOHDM para la creación de un correcto diseño.
- Diseño de componentes.
- Diseño y creación de Vistas/Formularios.
- Definición de estándares para la aplicación.

El Desarrollo se refiere a:

- Desarrollo de la Administración de Plantillas.
- Desarrollo de Administración de Documentos en donde se contempla los siguientes módulos.

1. *Búsqueda de documentos:* En donde se empleara la indexación con el fin de tener una disponibilidad de la información y optimizar el proceso de búsqueda.

2. *Control de Acceso:* Para este módulo se trabaja con perfiles de usuario, donde se contralará el manejo que este perfil tiene hacia el documento considerando (modificación, lectura, envió, impresión, etc.) del Documentos.

3. *Correo:* En este módulo se implementara el envió de documento.

4. *Versionamiento:* En este modulo se realizara el control de todo tipo de documentos, donde cada versionamiento tendrá su justificación o observación.

La Implementación se refiere a:

- Implantación de plataforma de Adobe Capture.
- Implementación del SADO

Finalmente se procederá a la configuración e instalación de equipo y software necesarios y el levantamiento de servicios para generar las pruebas necesarias con el fin de que se cumplan las expectativas de rendimiento y usabilidad que los usuarios requieren.

1.5 Herramientas

Con respecto a las herramientas para el desarrollo del sitio Web se definen de acuerdo a las tres capas que posee la aplicación:

Servidor de Aplicaciones

Nombre	Descripción
Sistema Operativo Windows Server 2008 Standard	Windows Server 2008 es un sistema operativo de propósitos múltiples capaz de manejar gran cantidad de funciones de servidor, de manera centralizada o distribuida. Más seguro, fiable y con mayor disponibilidad que cualquiera de sus antecesores, mejorado y ampliado para

	<p>aprovechar mejor los beneficios de la plataforma .NET.</p> <p>Algunas de sus capacidades:</p> <ul style="list-style-type: none"> • Servidor de archivos e impresión. • Servidor web y aplicaciones Web. • Servidor de correo. • Terminal Server • Servidor de acceso remoto/red privada virtual (VPN). • Servidor de directorio, Sistema de dominio (DNS), y servidor DHCP.
--	--

Tabla 1.1 Servidor de aplicaciones

Servicios

Nombre	Descripción
Acrobat Capture 3.0	Convierte grandes volúmenes de documentos de papel en archivos en formato PDF de Adobe en los que se pueden efectuar búsquedas.

Tabla 1.2 Servicios

Para la Programación

Nombre	Descripción
Microsoft Visual Studio.NET 2008	Es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones
.Net framework 3.x	<p>.NET es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.</p> <p>Este es el lenguaje insignia de .NET Framework (marco de trabajo .NET) y pretende reunir las ventajas de lenguajes como C, C++ y Visual Basic en uno solo. El CLR es el verdadero núcleo del framework de .NET, entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando el conjunto de servicios del sistema operativo .</p>

Tabla 1.3 Para la Programación

Para la capa de Base de Datos

Nombre	Descripción
Microsoft SQL Server Express 2008	<p>SQL Server Express es una versión de SQL Server 2005 diseñada para ayudar a los desarrolladores a construir aplicaciones robustas y fiables ofreciendo una sencilla pero potente base de datos que es además gratuita.</p> <p>SQL Server 2005 Express Edition proporciona muchas nuevas funcionalidades mejoradas, como son:</p> <ul style="list-style-type: none">• Una base de datos robusta para crear aplicaciones dinámicas• Fuerte soporte de XML• Herramientas y características para mejorar la gestión y facilidad de uso• Instalación y setup sencillos• Escalabilidad y rendimientos• Gran integración con Visual Studio 2005• Robusta seguridad

Tabla 1.4 Capa de Base de Datos

CAPÍTULO II

MARCO TEÓRICO

2.1 Sistema de Digitalización de Documentos

2.1.1 ¿Qué son Sistemas Digitales?

Este siglo se caracteriza por el gran manejo de información en su gran variedad de formatos, presentaciones por lo cual en el caso particular de poseer información impresa o física esta presenta gran riesgo de pérdida o deterioro, difícil acceso, no admite concurrencia, por lo cual la principal solución para evitar este tipo de suceso es digitalizar dicha información. De ahí que en base a esta necesidad se crearon *Los Sistemas de Digitalización de Documentos*. En otras palabras es aquel sistema que incorpora toda la información de soporte físico, convirtiéndolos en formatos electrónicos lo que conlleva cubrir los problemas que poseen los documentos físicos. “Sin necesidad de entrar en polémicas muy grandes, ya sean políticas o burocráticas, este procedimiento (la digitalización) vendría a facilitar en gran medida las cosas, es una actividad que empieza a darse y que vendría a ayudar a la conformación de la historia en los países”, comentó el director del Archivo General de la Nación de México.

2.1.2 Características de un Sistema Digital

Un sistema digital tiene como base los documentos digitales por lo cual se explicará las características que un documento digital.

2.1.3 Características de un Documento Digital

Los documentos digitales tienen características que los diferencian de los documentos tradicionales:

2.1.3.1 Características Funcionales

a. Carácter Interactivo de los Documentos Digitales

Un documento cuando se encuentra impreso posee un estado de pasivo ya que solo se podría aplicar la acción de leer carece de interactividad.

El documento electrónico en cambio permite que el usuario pregunte por contenidos, ingrese comentarios, modifique o agregue contenidos, cambiar aspectos de formato. Y así la interactividad puede establecer un cierto grado de comunicación con el documento electrónico y el lector simula una iteración de mensaje y respuesta en eso consiste la interactividad

b. Carácter multimediático de los documentos digitales

Los documentos digitales permiten combinar distintas formas de presentar información en un sólo documento. El documento digital puede, así, combinar texto con sonido; sonido con imágenes; imágenes con texto, o las tres cosas simultáneamente. Un documento electrónico puede describir conceptualmente un proceso, ilustrarlo con animaciones en video, presentar sus consecuencias con fotografías y narrarlo paralelamente a la lectura del texto. Es a esta posibilidad de utilizar "múltiples" medios informativos, lo que denomino "carácter multimediático" de los documentos digitales.

La condición multimediática de los documentos electrónicos trae como consecuencia que sea posible concebir nuevos géneros. El documento electrónico permite combinar en un sólo ambiente las posibilidades del cuento o la poesía, con la ilustración de esos géneros con audiovisuales. O viceversa, combinar documentos audiovisuales con textos ampliamente explicativos o sugestivos. O simplemente combinar los medios expresivos de nuevas maneras para crear nuevos géneros.

c. Carácter hipertextual de los documentos digitales

Los documentos tradicionales obligan a lecturas lineales parece ser una limitación más sentida por los autores que por los usuarios del documento tradicional. El autor, permanentemente siente la necesidad de apoyar sus planteamientos; de complementarlos con los elaborados por otros; de referirse a datos que no caben en la estructura de su discurso; de hablar por intermedio de terceros. De allí las notas a pie de página, las citas, las referencias y bibliografías.

Un documento digital, es posible establecer relaciones entre palabras, frases, imágenes, textos o sonidos ubicados en un documento, e información textual, visual o auditiva, ubicada en cualquier otro documento al cual se pueda acceder electrónicamente; es decir, con tal que se encuentre en el mismo disco, computadora o red. Es esta característica la que permite que se pueda navegar de una "Página Web" a otra a través de Internet.

Las relaciones hipertextuales pueden haber sido hechas por el autor original del documento o por un lector o consultante a posteriori, lo cual, al crear nuevas rutas de navegación lo transforma en coautor del discurso que se conforme durante el acto de navegar. De esta manera, al leer navegando entre páginas y documentos, el lector

construye su propio documento. Construye un hiperdocumento único para él o incluso único para la ocasión de la lectura. La hipertextualidad y la hipermedialidad son características definitivamente transformadoras de las formas comunicacionales modernas.

d. Carácter “omniaccesible” de los documentos digitales

El documento tradicional, físicamente considerado, siempre tendrá obstáculos para su accesibilidad total. Los documentos digitales esta limitación es superada. La posibilidad de establecer "hiperrelaciones" entre documentos ubicados en distintos sitios Web, no importa la región o país donde se encuentren, permite que los documentos electrónicos, ubicados no importa donde, si están en un medio conectado en red con Internet, puedan ser accedidos desde cualquier otro lugar. Son, en consecuencia, accesibles desde todas partes; son "omniaccesibles".

2.1.3.2 Mutabilidad e Identidad en los Documentos Digitales

a. Carácter “mutable” de los documentos digitales

El documento digital se produce en el mismo medio en que se publica: un medio electrónico. Esto facilita enormemente efectuar cambios en él. No es necesario reproducirlo por entero una vez que se le hayan efectuado cambios al documento, como es el caso en un documento impreso o audiovisual tradicional por en consecuencia, una gran mutabilidad presenta los documentos digitales.

b. “Volatilidad” de los documentos digitales

Un documento electrónico es fácilmente publicable para que el mundo entero lo pueda consultar. Pero así como es fácil publicarlo, es igualmente fácil sacarlo de circulación.

Esta característica es especialmente importante a la hora de utilizar documentos electrónicos accesibles en línea, como apoyo documental en artículos u otro tipo de trabajo académico o científico. Al citarlos es necesario dejar claro en qué fecha se consultó; incluso, a qué hora se consultó.

2.1.3.3 Reconocimiento de Caracteres Ópticos(OCR)

La tecnología OCR proporciona a los sistemas de reproducción por escáner y sistemas de imágenes la habilidad de convertir imágenes de caracteres en letra de máquina, en caracteres capaces de ser interpretados o reconocidos por una computadora. Así, las imágenes de caracteres en letra de máquina son extraídas de un mapa de bits de la imagen reproducida por el escáner.

Básicamente, el sistema que utilizan es el siguiente:

- El sistema de digitalización no hace mucho consistía en que el OCR aislaba la imagen correspondiente a un carácter y la comparaba con una base de caracteres para determinar su correspondiente código ASCII o bien Unicode.
- Una vez reconocido lo transformaba y lo trasladaba al documento OCR resultante. Este sistema tenía el inconveniente de que trabajaba con un

número limitado de fuentes, por lo que el resultado obtenido no siempre era el deseado.

- Además, en textos muy largos y con inclusión de frases en cursiva, por ejemplo, el proceso era bastante lento.

Actualmente los sistemas de digitalización son capaces de reconocer no solo el texto en sí, sino también el estilo y formato de este, aunque dentro de unas limitaciones, haciendo necesario que posteriormente se edita el texto resultante es cierto que también pueden reconocer textos escritos manualmente, pero en este caso siempre a condición de que estén escritos claramente, a ser posible con letra de molde, y que esta se asemeje lo más posible a un tipo de letra existente.

2.1.4 Usos de Sistemas de Digitalización de Documentos

Los Sistemas de Digitalización de Documentos se usan en diversas instituciones como bibliotecas, empresas públicas o privadas, universidades.

Una de las ventajas que comparten todas las empresas que implementan un sistema digitalización de documentos es la solución que brinda de administración y digitalización de documentos a empresas de todo tipo, cumpliendo así todas, disponibilidad y consulta, contribuyendo a incrementar la seguridad, productividad, rentabilidad, transparencia y orden en la información de su cliente

Para la empresa de Soluciones Inteligentes su principal objetivo es facilitar la gestión de documentos a sus empleados como son formularios o facturas documentos en

general pero gracias a la implementación del sistema SADO se busca un ahorro de recursos y tiempo

2.1.5 Ventajas y Desventajas de los Sistemas Digitales de Documentos, Digitalización de Documentos y Archivos de Papel

2.1.5.1 Ventajas Sistema Digital

- Los requisitos se definen en forma progresiva, en lugar de hacerlos al inicio.
- Se mantiene el control directo sobre la digitalización de documentos y toda la gama de funciones derivadas.
- Se prevé una seguridad consistente y de calidad.
- Se asegura una manipulación adecuada y la accesibilidad respecto al fondo documental.

2.1.5.2 Desventajas Sistema Digital

- Gran inversión y tiempo prolongado en la etapa inicial del proyecto.
- No hay precio fijo por documento digitalizado.
- La institución paga los gastos (incluyendo los costos de tiempo de inactividad, capacitación / experiencia del personal y obsolescencia tecnológica) en vez de pagar por los productos.
- Capacidad de producción (personal y equipamiento) limitada.

2.1.5.3 Ventajas Digitalización de documentos

- Elimina el extravío de documentos
- Permite un rápido y fácil acceso a la información.
- Resguarda la información en múltiples formatos electrónicos.
- Controla el acceso a la información por niveles de seguridad. Firmas digitales.
- Mejora los tiempos de acceso para la búsqueda de documentos
- Permite almacenar todo tipo de documentos (tamaño, formato, color, etc.).
- Resguarda la documentación. Optimiza el espacio físico convirtiendo pilas de papel en un simple CD-ROM o almacenado en una maquina.
- Protege la información teniendo varios respaldos de la información.
- Agiliza y eficiencia de taza el servicio a clientes
- Ordena de manera lógica los expedientes.
- Permite una fácil distribución de documentos vía Internet o intranet
- Protege los documentos contra la acción de agentes externos y deterioro de los mismos
- Reduce costos de operación.
- Incrementa la productividad.
- Permite simultaneidad de usuarios al mismo documento.
- Diferentes sistemas operativos pueden tener acceso a los archivos digitales.
- Bajo costo de almacenamiento

- Elimina costos de impresión, fotocopiado de documentos

2.1.5.4 Desventajas del archivo en papel

- Elaboración de Índices, gastos de almacenaje,
- Múltiples Archivos, Depuración de los documentos, Localización,
- Manipulación, Archivamiento,
- Pérdida de tiempo en la búsqueda de documentos,
- Retraso en firmas y autorizaciones,
- Elevados costos, Extravío de documentos, Exceso en fotocopias,
- Duplicidad de trabajo, Archivos duplicados,
- Falta de seguridad y confiabilidad, Uso cooperativo,
- Altos costos de administración, Velocidad de respuesta, Ecología,
- Seguimiento y control de documentos.

2.2 Metodología para el desarrollo de software

2.2.1 Metodología de Diseño de Hipermedia Orientada a Objetos (OOHDM)

2.2.1.1 Definición

OOHDM es una metodología orientada a objetos y ampliamente aceptada para el desarrollo de aplicaciones hipermedia, y en particular de aplicaciones Web. Se basa en dos premisas fundamentales:

- La utilización de casos de uso para la fácil captura y definición de requisitos por parte de los usuarios y clientes no expertos; de tal manera que los analistas posteriormente puedan generar los user interaction diagrams (UIDs) o modelos gráficos que representan la interacción entre el usuario y el sistema, sin considerar aspectos específicos de la interfaz.
- Comenzar el diseño del sistema con el fin de que las necesidades de interacción del sitio Web sean las deseadas por los usuarios.

2.2.1.2 Características Fundamentales

- *Apropiada para un comportamiento complejo.*

OOHDM provee las mejores herramientas para controlar el desarrollo de una aplicación que tiene un complejo comportamiento.

- *Separación del diseño con respecto al desarrollo.*

Permite que la complejidad del desarrollo de software sea menor ya que ésta ocurre a diferentes niveles: “dominios de aplicación sofisticados (financieros, médicos, geográficos, etc.); la necesidad de proveer acceso de navegación simple a grandes cantidades de datos, y por último la aparición de nuevos dispositivos para los cuales se deben construir interfaces *Web* fáciles de usar”.¹

¹ Tomado de: Darío Andrés Silva, Construyendo Aplicaciones Web con una Metodología de Diseño Orientado a Objetos, 2002, pp 2

2.2.1.3 Etapas ó Fases

OOHDM propone el desarrollo de aplicaciones Web a través de cinco etapas donde se combinan notaciones gráficas UML (Lenguaje de Modelado Unificado) con otras propias de la metodología y son:

- Definición u Obtención de Requerimientos
- Diseño o Modelo Conceptual
- Diseño Navegacional
- Diseño de Interfaces Abstractas
- Implementación.

En cada etapa se crean un conjunto de modelos orientados a objetos que describen un diseño particular, tal como se indica en la tabla 2.1.

El éxito de esta metodología es la clara identificación de los tres diferentes niveles de diseño en forma independiente de la implementación.

Etapas	Productos	Formalismos	Mecanismos	Descripción
<i>Obtención de Requerimientos</i>	Casos de Uso (actores, escenarios)	Plantillas del formato del documento, Diagramas de Interacción de Usuario (UIDs)	Técnicas de Observación, Entrevistas	Se crea un documento que describe actividades y requerimientos de los usuarios
<i>Diseño Conceptual</i>	Clases, subsistemas, relaciones, atributos	Modelos Orientados a Objetos	Clasificación, agregación, generalización y especialización	Se modela la semántica del dominio de la aplicación
<i>Diseño Navegacional</i>	Nodos, enlaces, estructuras de acceso, contextos	Vistas Orientadas a Objetos, Cartas de navegación	Clasificación, agregación, generalización	Se tiene en cuenta el perfil del usuario y las tareas. Se enfatiza en

	navegacionales, transformaciones de navegación	orientadas a objetos, Clases de Contexto	y especialización	los aspectos cognitivos. Se crea la estructura de navegación de la aplicación
<i>Diseño de Interfaz Abstracta</i>	Objetos de la interfaz abstracta, respuestas a eventos externos, transformaciones de la interfaz	Vistas Abstractas de Datos (ADV), Diagramas de Configuración, Cartas de navegación de los ADVs	Mapeado entre la navegación y los objetos visibles	Se modelizan los objetos visibles. Se describe la interfaz para los objetos de navegación. Se define el aspecto de los objetos de la interfaz
<i>Implementación</i>	Aplicación en funcionamiento	Los soportados por el entorno	Los que provea el entorno	Se realiza la puesta en producción del sistema

Tabla 2.1 Etapas de la Metodología OOHDM

2.2.1.4 Definición de Requerimientos

Se basa en los diagramas de casos de uso², los cuales son diseñados por escenarios con la finalidad de obtener de manera clara los requerimientos y acciones del sistema y de los usuarios.³

Describe el alcance del sistema definiendo los requerimientos funcionales y los no funcionales.

Primero que todo es necesaria la recopilación de requerimientos mediante entrevistas, observaciones o cualquier otra técnica que facilite el trabajo a los desarrolladores y que los usuarios o clientes se sientan cómodos.

² **Caso de Uso:** Descripción a detalle de las actividades y procesos necesarios para el desarrollo de un sistema o aplicación.

³ **Usuarios:** Es un ente humano que usa al sistema. Un mismo usuario puede actuar como instancias en varios actores diferentes, es decir, puede jugar diferentes roles.

En este punto, se hace necesario identificar los actores⁴ y las tareas que ellos deben realizar.

Luego, se determinan los escenarios para cada tarea y tipo de actor. Los casos de uso que surgen a partir de aquí, serán luego representados mediante los Diagramas de Interacción de Usuario (UIDs), los cuales proveen de una representación gráfica concisa de la interacción entre el usuario y el sistema durante la ejecución de alguna tarea. Con este tipo de diagramas se capturan los requisitos de la aplicación de manera independiente de la implementación.

Ésta es una de las fases más importantes, debido a que es aquí donde se realiza la recopilación de datos, para ello se debe proporcionar las respuestas a las siguientes preguntas:

- ¿Cuáles son los tópicos principales que serán atendidos?
- ¿Cómo los tópicos están relacionados entre sí?
- ¿Qué categoría de usuarios serán atendidos?
- ¿Cuáles son las tareas principales que serán abordadas?
- ¿Qué tareas corresponden a qué categoría de usuarios?
- ¿Los recursos disponibles son competitivos con la información levantada?⁵

⁴ **Actores:** Son objetos que se encuentran fuera del sistema a modelar. Representan entes que tienen necesidad de intercambiar información con el sistema; pueden ser instanciados por usuarios, dispositivos u otros sistemas.

⁵ **Preguntas claves tomadas de:** OOHDM Object Oriented Hypermedia Design Methodology, Agosto 2006

Con estas preguntas se puede recaudar de cierta manera las bases necesarias para la construcción de una aplicación hipermedial exitosa, sin embargo mientras mayor sea el nivel de profundidad de la recolección de datos, existirá mayor probabilidad de cubrir todos los requerimientos solicitados por los usuarios.

OOHDM propone dividir esta etapa en 5 subetapas:

- Identificación de Roles y Tareas
- Especificación de escenarios
- Especificación de casos de uso
- Especificación de UIs
- Validación de Casos de Uso y UIs

2.2.1.4.1 *Identificación de Roles y Tareas*

Roles

Se determina los posibles roles que podrían tomar los potenciales usuarios de la aplicación, ya que los usuarios pueden tomar diferentes roles importantes en cada intercambio de información con el sistema.

Tareas

Para cada rol se debe identificar las tareas que deberá soportar la aplicación.

2.2.1.4.2 Especificación de Escenarios

Los escenarios son descripciones narrativas de cómo la aplicación será utilizada. En esta subetapa, cada usuario deberá especificar textual o verbalmente los escenarios que describen su tarea.

2.2.1.4.3 Especificación de Casos de Uso

Un caso de uso es la descripción a detalle de las actividades y procesos necesarios para utilizar la aplicación. Representan la interacción entre el usuario y el sistema, agrupando las tareas especificadas en los escenarios existentes.

Por ello, es muy importante identificar cuál es la información relevante en cada uno de los escenarios, para luego generar un caso de uso coherente.

2.2.1.4.4 Especificación de UIs

Los UIs permiten representar en forma gráfica, rápida y sencilla los casos de uso generados en la etapa anterior.

Para obtener un UIs desde un caso de uso, la secuencia de información intercambiada entre el usuario y el sistema debe ser identificada y organizada en las interacciones. Identificar la información de intercambio es crucial ya que es la base para la definición de los UIs.

2.2.1.4.5 Validación de Casos de Uso y UIs

El desarrollador deberá interactuar con cada usuario para validar los casos de uso y UIs obtenidos, mostrando y explicando cada uno de ellos para ver si el o los usuarios están de acuerdo. El usuario deberá interceder sólo en aquellos casos de uso y UIs en que participa.

2.2.1.4.6 Diseño Conceptual

Está representado por los objetos del dominio, las relaciones y colaboraciones existentes establecidas entre ellos; es decir, está formado por clases, relaciones y subsistemas. Este diseño se define de acuerdo a los UIs establecidos.

Las **clases** son descritas y usadas como en los modelos orientados a objetos tradicionales; conectadas por relaciones y los objetos son instancias de las clases. Sin embargo, los atributos pueden ser de múltiples tipos para representar perspectivas diferentes de las mismas entidades del mundo real.

Se puede usar un modelo de datos semántico estructural (como el modelo de entidades y relaciones). Utiliza una notación similar a UML.

Importante:

- Las clases son usadas en el diseño navegacional para derivar nodos, y las relaciones para construir enlaces o vínculos.

- Si la aplicación crece el diseño conceptual debe estar preparado para ser extendido, tal como cualquier diseño orientado a objetos.

2.2.1.4.7 *Diseño Navegacional*

Un modelo navegacional es construido como una vista sobre un diseño conceptual, admitiendo la construcción de modelos diferentes de acuerdo con los diferentes perfiles de usuarios. Cada modelo navegacional provee de una vista subjetiva del modelo conceptual.

Su objetivo es permitir a la aplicación ejecutar todas las tareas requeridas por el usuario, es decir, unificar una serie de tareas para obtener el diseño navegacional de la aplicación.

El diseño navegacional es expresado en dos esquemas: el esquema de clases navegacionales y el esquema de contextos navegacionales como se puede verificar en la Figura 2.1.

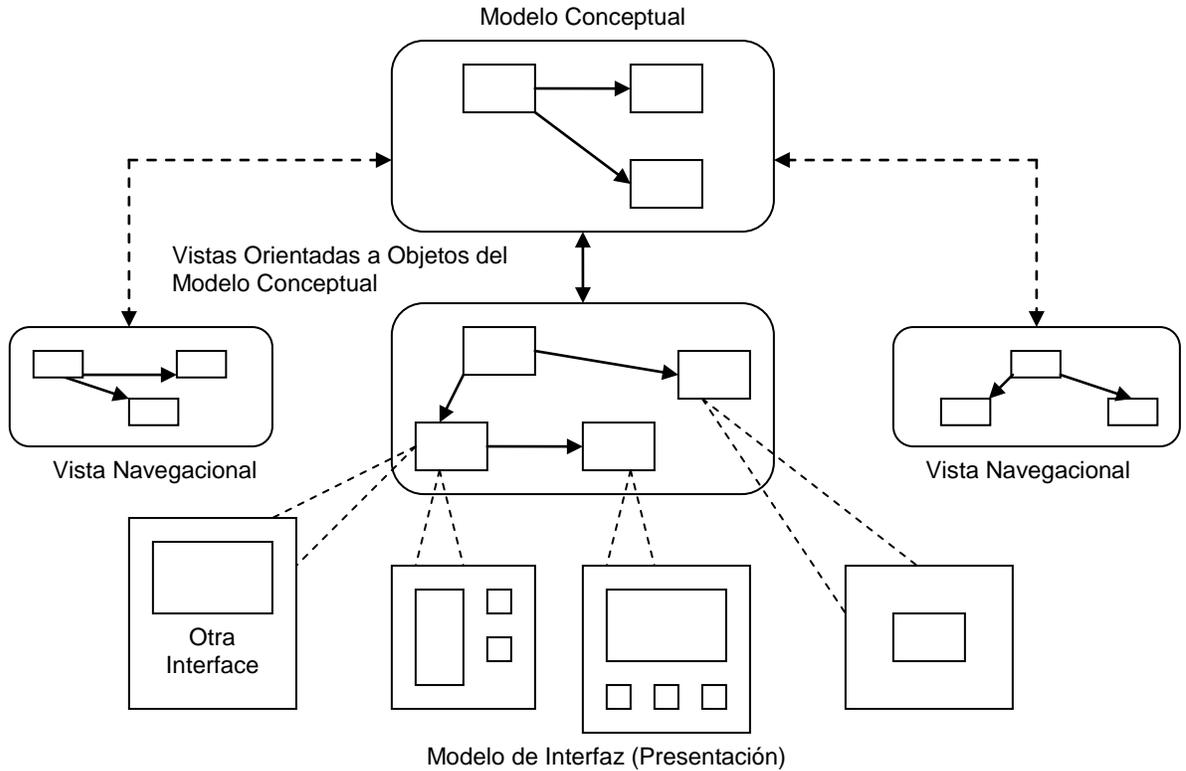


Figura 2.1 Esquema del Diseño Navegacional

2.2.1.4.8 *Esquema de Clases Navegacionales*

Las posibles vistas del hiperdocumento se generan a través de las llamadas clases navegacionales que son tipos predefinidos de clases conformados por: nodos, enlaces, anclas⁶ y estructuras de acceso que son los índices o recorridos guiados, que representan los posibles caminos de acceso a los nodos.

Nodos

- Contenedores de información básicos

⁶ **Ancla:** Los atributos de los nodos que activan navegaciones son llamados anclas.

- Se definen como vistas orientadas a objetos de las clases conceptuales definidas en la fase de Diseño Conceptual
- Contienen atributos simples y enlaces

Enlaces

- Identifican relaciones
- Implementan las relaciones definidas en el esquema conceptual
- Las clases de los enlaces especifican sus atributos, comportamiento y los objetos fuente y destino
- Representan las posibles formas de comenzar la navegación

Estructuras de Acceso

- Actúan como índices o diccionarios
- Útiles para ayudar al usuario final a encontrar la información deseada
- Ejemplos: menús, índices y tours guiados

2.2.1.4.9 *Esquema de Contextos Navegacionales*

Un esquema de contexto navegacional es el que organiza el espacio navegacional en conjuntos convenientes o subespacios que pueden ser recorridos en un orden particular y que deberían ser definidos como caminos para ayudar al usuario a lograr la tarea deseada que es obtener la información requerida, junto con los enlaces que estarán disponibles cuando se acceda a un objeto en un contexto determinado.

Contexto navegacional

- Es un conjunto de nodos, enlaces, clases de contextos y otros contextos navegacionales (contextos anidados).
- Inducido de clases navegacionales
- Se definen por extensión o enumerando sus miembros
- Un índice o un tour guiado definen contextos de navegación

2.2.1.4.10 *Diseño de Interfaces Abstractas*

En esta etapa se define la forma en la cual los objetos navegacionales pueden aparecer, es decir, especifica la estructura y el comportamiento de la interfase del sistema hipermedia con el usuario. Este modelo es abstracto y, por tanto, independiente de la implementación final del sistema.

Aquí los objetos de interfaz activarán la navegación y el resto de funcionalidades de la aplicación, esto es, se describirán los objetos de interfaz y se los asociará con objetos de navegación. La separación entre el diseño navegacional y el diseño de interfaz abstracta permitirá construir diferentes interfaces para el mismo modelo navegacional.

Este diseño se basa en tres diagramas que se complementan entre sí, y son:

- Diagrama de Vista de Datos Abstractos (ADV)
- Diagrama de Configuración
- Diagrama de Estado

2.2.1.4.11 Diagrama De Vista De Datos Abstractos (ADV)

Este diagrama incluye una vista (ADV) por cada clase navegacional (nodo, enlace o estructura de acceso) que fue establecida durante la fase de Diseño Navegacional.

Se compone de una serie de cajas⁷ que representan las diferentes clases de objetos que aparecerán ante el usuario.

2.2.1.4.12 Diagrama de Configuración

En este diagrama se representan principalmente:

- Los eventos externos provocados por el usuario, como *Clic o DobleClic del Ratón* que maneja un ADV.
- Los servicios que ofrece el ADV (como "visualización").
- Las relaciones estáticas entre las ADVs.

2.2.1.4.13 Diagrama de Estado

Como en el caso de los modelos OO⁸ de Ingeniería del Software, se representa el comportamiento dinámico del sistema hipertextual mediante el establecimiento de un diagrama de transición de estados para cada ADVs, en el que se reflejan los posibles

⁷ **Caja:** Es un ADV

⁸ **OO:** Orientados a Objetos

estados por los que puede pasar cada objeto de la interface (*oculto, desactivado, ampliado, reducido, normal, etc.*) y los eventos que originan los cambios de estado.⁹

2.2.1.4.14 Implementación

En esta fase se debe implementar el diseño, ya que hasta el momento todos los modelos fueron construidos en forma independiente de la plataforma de implementación y por tanto, el desarrollador ya ha identificado la información que será mostrada, cuenta con una idea básica de cómo se verán las interfaces y tiene un completo conocimiento del dominio del problema.

Para comenzar con la implementación el desarrollador deberá elegir dónde almacenará los objetos y con qué lenguaje o herramienta desarrollará las interfaces; para después continuar con la creación de las mismas, de acuerdo a la secuencia de requerimientos solicitados.

2.2.1.5 Ventajas y Desventajas

Ventajas

- Clara identificación de los tres diferentes niveles de diseño en forma independiente de la implementación.
- Su forma de representación gráfica es bastante completa y permite representar en forma precisa elementos propios de las aplicaciones

⁹ Tomado de la página: *Crea tu Página Web*, wmaestro.com/webmaestro/docs/portada.html

hipermedia, tales como nodos, anclas, vínculos, imágenes, estructuras de acceso y contextos.

- En la etapa de diseño navegacional se pueden crear enlaces entre nodos cualesquiera que permiten una verdadera interoperabilidad entre los mismos.
- El desarrollador puede entender y lograr en cada etapa lo que el usuario realmente necesita, gracias a que en el análisis y diseño, el usuario es parte fundamental en la validación del producto obtenido.
- Al generar una cantidad considerable de documentación a través de sus distintas etapas, permite llevar un control del desarrollo de las mismas y tener la posibilidad de realizar una rápida detección, corrección de errores y mantención.
- La utilización de UIDs permite representar en forma clara, rápida y precisa los casos de uso obtenidos.

Desventajas

- Requiere de cierto conocimiento e investigación para aprender la metodología, debido a los modelos que utiliza.
- El diseño pierde un poco de continuidad del modelo navegacional al diseño de interfaz, dado que se pasa a utilizar otro tipo de modelo.
- En ciertos casos OOHDM podría exagerar la cantidad de reglas y pasos (a veces complicados de seguir) para realizar distintos mapeos entre un diagrama y otro por lo cual el desarrollador podría perderse y olvidar detalles fundamentales a ser especificados.

- El diseño navegacional posee una gran cantidad de diagramas que muchas veces entregan información similar a la entregada por los UIDs y las ADVs.

2.2.1.6 Criterios de Selección de OOHDM

OOHDM es una metodología de diseño hipermedial, que utiliza el enfoque orientado a objetos, extendiéndolo e integrándolo con técnicas de representación gráfica de relaciones entre objetos y de contextos navegacionales que son ricos en representación estructural y semántica.

2.2.2 Lenguaje Unificado De Modelado (UML)

2.2.2.1 Definición

UML (Unified Modeling Language) o Lenguaje Unificado de Modelado es un lenguaje gráfico para visualizar, especificar, construir y documentar los elementos de un sistema, establece un conjunto de notaciones, diagramas estándar para modelar sistemas orientados a objetos describiendo la semántica esencial de estos diagramas y los símbolos en ellos utilizados.

UML representa una recopilación de mejores técnicas de ingeniería que se han probado ser exitosas en el desarrollo de sistemas grandes y complejos, es por ello que UML puede usarse con una gran variedad de metodologías o procesos de desarrollo de software, siendo un lenguaje para especificar y no un método o un proceso.

2.2.2.2 Diagramas

UML versión 2.0 cuenta con los siguientes tipos de diagramas que permiten modelar el problema y su solución:

- Diagramas de Casos de Uso
- Diagrama de Secuencia
- Diagrama de Colaboración
- Diagrama de Objeto
- Diagrama de Clase
- Diagrama de Estado
- Diagrama de Actividad
- Diagrama de Despliegue
- Diagrama de Componente

2.2.2.3 Diagramas de Caso de Uso

Un caso de uso es una descripción de las iteraciones que se producen entre un usuario y un sistema de cómputo, especifican el comportamiento deseado del sistema por parte del usuario para llevar a cabo una tarea concreta, pero no especifica cómo lo hace.

Un diagrama de caso del uso es una colección de actores¹⁰, de casos del uso, y de sus relaciones.

¹⁰ **Actor:** Representa un conjunto coherente de roles que juegan los usuarios de los casos de uso al interactuar con el sistema

Están basados en el lenguaje natural, es decir, es accesible por los usuarios. Los diagramas del caso del uso están conectados de cerca con escenarios. Un escenario es un ejemplo de qué sucede cuando alguien obra recíprocamente con el sistema.

A continuación un ejemplo de caso de uso en la Figura 2.2.

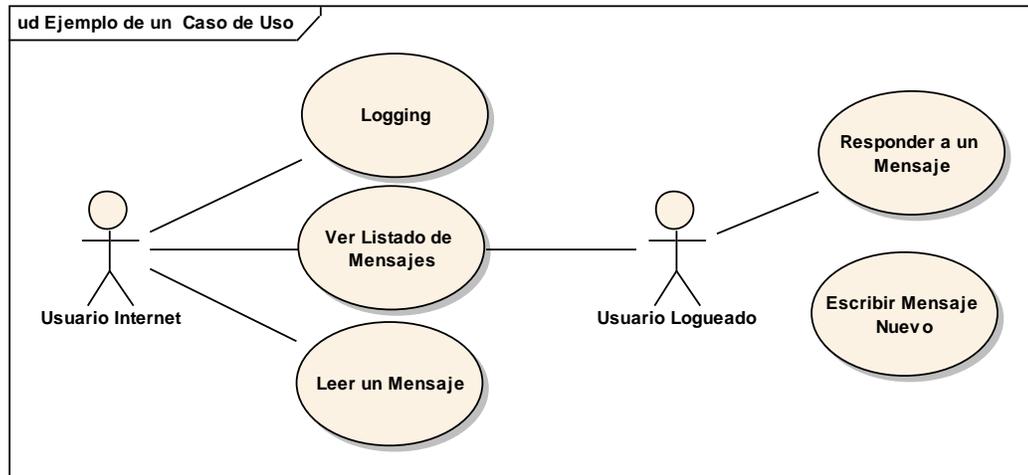


Figura 2.2 Ejemplo de un Caso de Uso

2.2.2.4 Diagramas de Secuencia

Muestran una interacción ordenada según la secuencia temporal de eventos de un caso de uso. Es un diagrama de la interacción que detalla cómo se realizan las operaciones: se envían qué mensajes y cuando. Se organizan según el tiempo, el cual mientras progresa se recorre el diagrama hacia abajo de la página.

Los diagramas de secuencia se centran en la temporalidad de los mensajes que se producen entre los objetos e incluyen: Objetos, línea de tiempo, mensajes con

argumentos, ciclo de vida de los objetos, información devuelta por un método y especificación de procesos concurrentes.

El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba hacia abajo. A continuación un ejemplo de diagrama de Secuencia en la Figura 2.3.

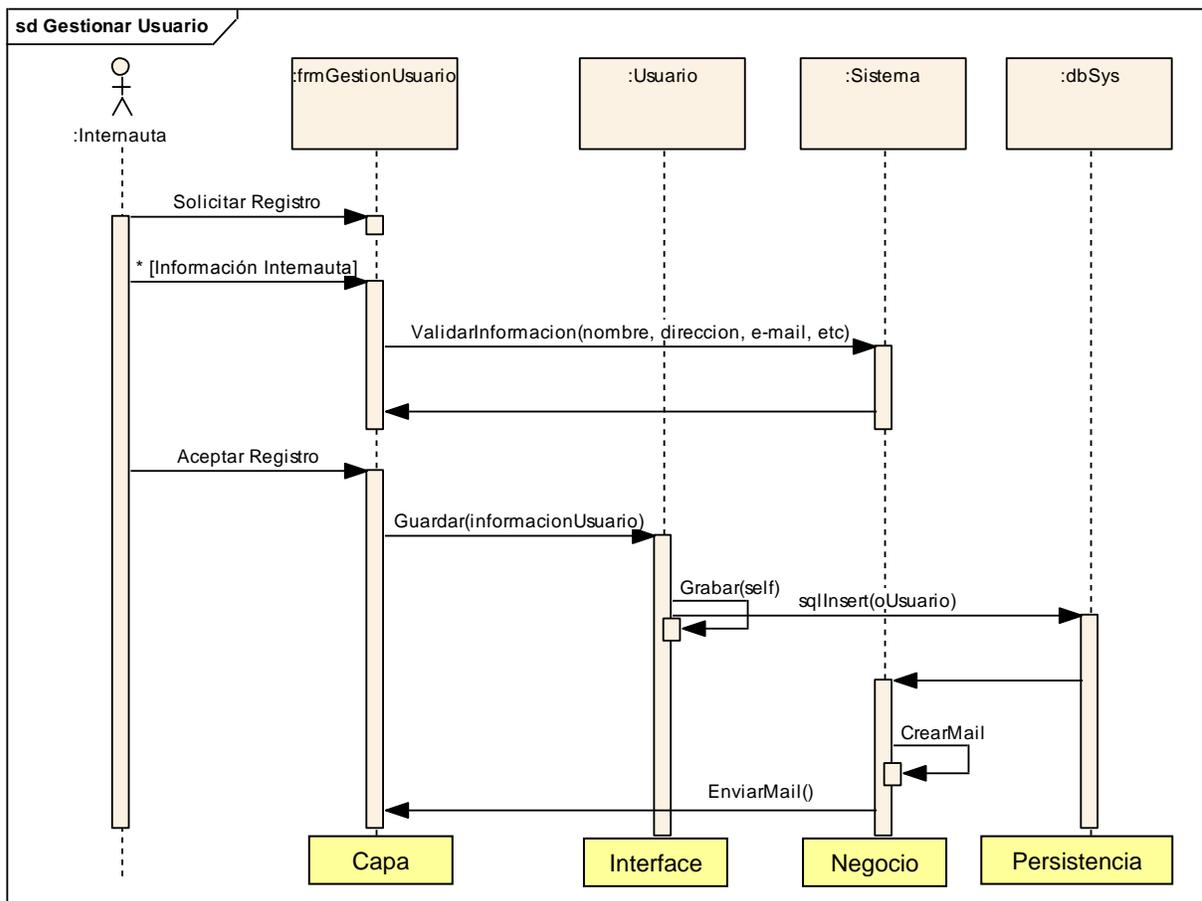


Figura 2.3 Ejemplo de un Diagrama de Secuencia

2.2.2.5 Diagramas de Colaboración

Estos diagramas forman parte de los Diagramas de Interacción. Transportan la misma información que los de Secuencia, pero se centran en las relaciones entre los roles de los objetos en vez del tiempo en que los mensajes son enviados.

La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia.

Su utilidad radica en mostrar la implementación de una operación. La colaboración muestra los parámetros y las variables locales de la operación, así como asociaciones más permanentes. También se lo utiliza para indicar cómo se conectan estáticamente los objetos. A continuación un ejemplo de diagrama de Colaboración en la Figura 2.4.

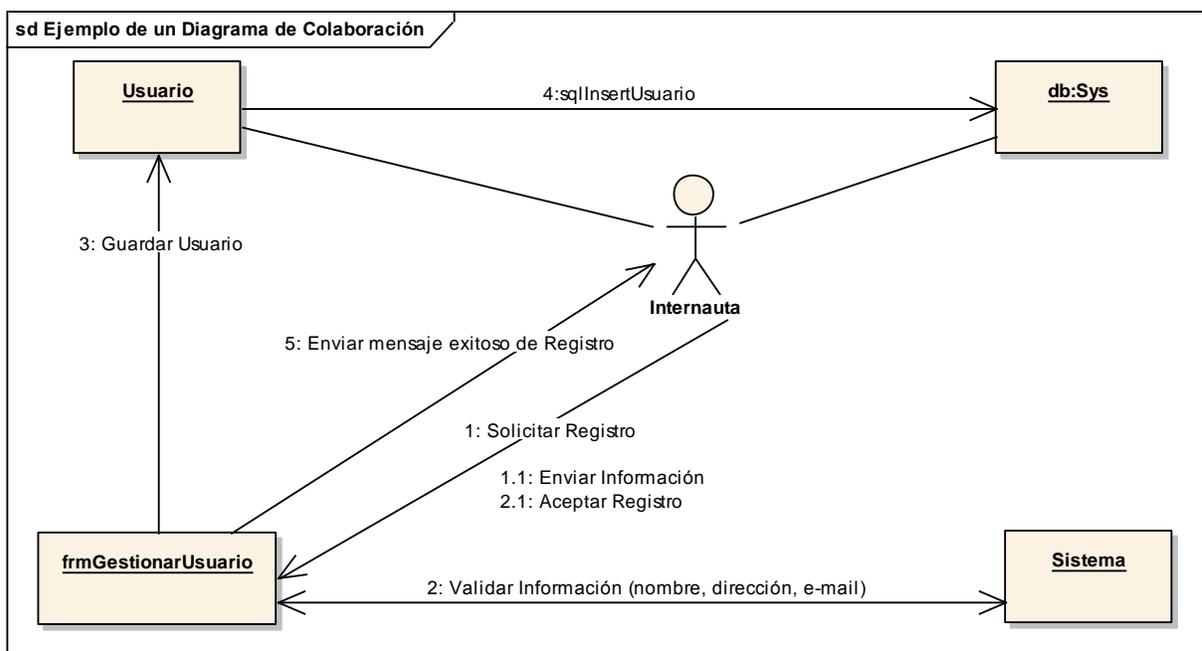


Figura 2.4 Ejemplo de un Diagrama de Colaboración

2.2.2.6 Diagramas de Objeto

Representan un conjunto de objetos y sus relaciones en un momento concreto de tiempo de ejecución. Modelan las instancias de los elementos contenidos en los diagramas de clases. Son la base para los diagramas de colaboración

Al representar una situación concreta del dominio, se distinguen de los otros tipos de diagramas en su contenido particular, el cual posee:

- Objetos
- Enlaces

Los diagramas de objetos pueden contener paquetes o subsistemas, los cuales se usan para agrupar los elementos de un modelo con partes más grandes. Un diagrama de objetos es esencialmente una instancia de un diagrama de clases o la parte estática de un diagrama de interacción.

Se emplean para visualizar, especificar, construir y documentar la existencia de ciertas instancias en el sistema, junto con las relaciones entre ellas. Los diagramas de objetos son útiles para modelar estructuras de datos complejas, como se puede ver en la Figura 2.5.

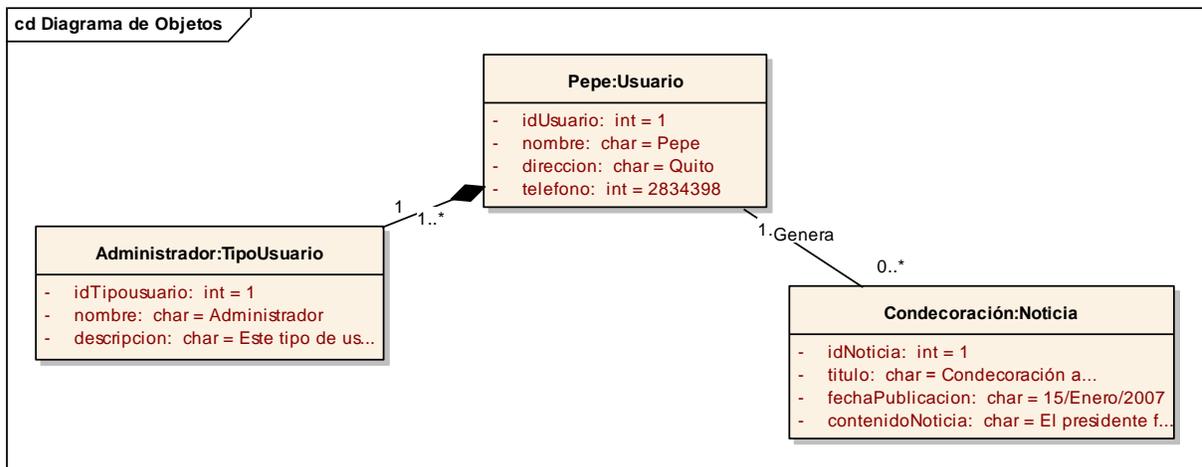


Figura 2.5 Ejemplo de Diagrama de Objetos

2.2.2.7 Diagramas de Clase

El Diagrama de Clases¹¹ es el diagrama principal para el análisis y diseño. Modela la vista estática estructural del sistema.

La definición de clase incluye definiciones para atributos y operaciones. El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones.

Cada diagrama de clases tiene clases, asociaciones, y multiplicidades. Adicionalmente la navegabilidad y los roles son artículos opcionales puestos en un diagrama para proporcionar claridad.

¹¹ **El Diagrama de Clases:** Es el más utilizado en los modelos de sistemas OO; son los “planos” principales [Booch].

Clases en UML

Los atributos de una clase no deberían ser manipulables directamente por el resto de objetos. Por esta razón se crearon niveles de visibilidad:

- (-) Privado: es el más fuerte. Esta parte es totalmente invisible
- (#) Los atributos/operaciones protegidos están visibles para las clases amigas y para las clases derivadas de la original.
- (+) Los atributos/operaciones públicos son visibles a otras.

A continuación un ejemplo de la clase Ventana en la Figura 2.6.

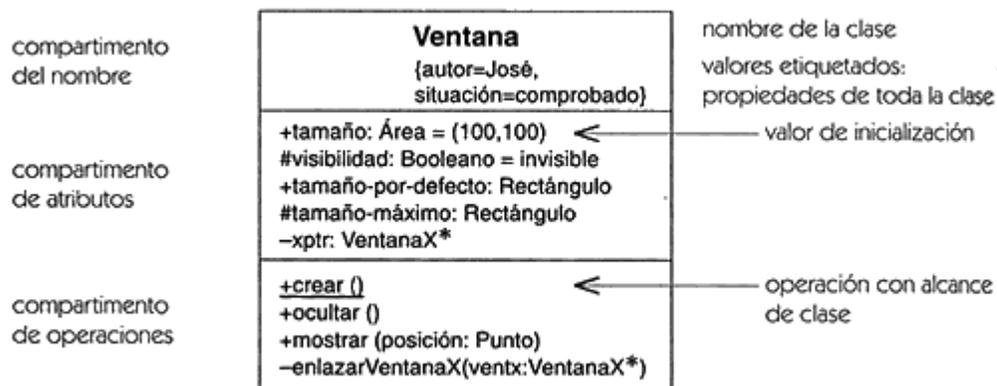


Figura 2.6 Ejemplo de una Clase¹²

¹² Figura tomada de la página: Diagramas de Clase, <http://www.creangel.com/uml/clases.php#>

Relaciones entre clases

Los enlaces entre objetos pueden representarse entre las respectivas clases y sus formas de relación son:

- Asociación y Agregación (vista como un caso particular de asociación)
- Generalización/Especialización.

Las relaciones de Agregación y Generalización forman jerarquías de clases.

Asociación

La asociación expresa una conexión bidireccional entre objetos. Las asociaciones representan relaciones entre instancias de clases. Puede determinarse por la especificación de multiplicidad (mínima...máxima)

- Uno y sólo uno
- 0..1 Cero o uno
- M..N Desde M hasta N (enteros naturales)
- 0..* Cero o muchos
- 1..* Uno o muchos (al menos uno)

Una asociación tiene dos extremos. Un extremo puede tener un nombre del rol para clarificar la naturaleza de la asociación.

Las asociaciones pueden denotar la navegabilidad mediante el sentido de las flechas. Las asociaciones sin flechas de navegabilidad son bidireccionales.

Agregación

Es una relación de componente, una asociación en la cual una clase pertenece a una colección. UML ofrece un tipo de agregación más poderosa que se denomina composición. El objeto parte puede pertenecer a un todo único.

Generalización

Consiste en factorizar las propiedades comunes de un conjunto de clases en una clase más general. Los nombres usados: clase padre - clase hija. Otros nombres: superclase - subclase, clase base - clase derivada. Las subclasses heredan propiedades de sus clases padre, es decir, atributos y operaciones (y asociaciones) de la clase padre están disponibles en sus clases hijas.

La Generalización y Especialización son equivalentes en cuanto al resultado: la jerarquía y herencia establecidas. La especialización es una técnica muy eficaz para la extensión y reutilización.

Como se puede apreciar en la Figura 2.7.

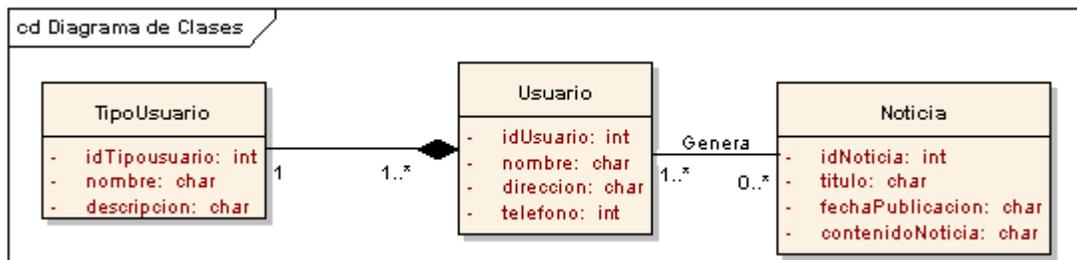


Figura 2.7 Ejemplo de un Diagrama de Clases

2.2.2.8 Diagramas de Estado

Representan la secuencia de estados por los que pasa un caso de uso, un objeto a lo largo de su vida, o todo el sistema. Describen todos los estados posibles en los que puede entrar un objeto en particular y la manera en que cambia el estado del objeto, como resultado de los eventos que llegan a él.

Se utilizan principalmente para representar el comportamiento de las instancias de una clase y están formados por:

Estados: Situación en la vida de un objeto, satisfaciendo una condición, realizando una actividad o esperando algún evento. Los estados están caracterizados parcialmente por los valores de algunos atributos del objeto.

Transiciones: Se cambia de un estado a otro, porque se produce un evento o se satisface una condición.

Las acciones se asocian con las transiciones y se consideran como procesos que suceden con rapidez y no son interrumpibles. Las actividades se asocian con los estados y pueden tardar más. Una actividad puede ser interrumpida por algún evento. Como se puede apreciar en la Figura 2.8.

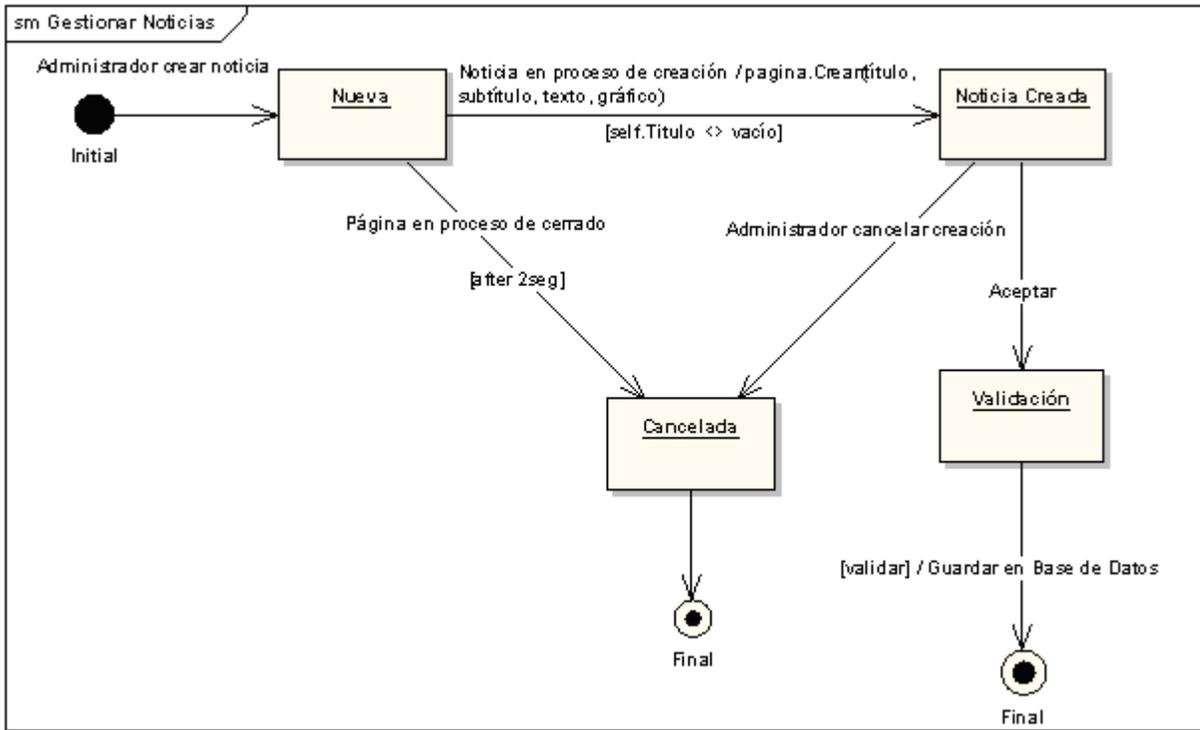


Figura 2.8 Ejemplo de un Diagrama de Estado

2.2.2.9 Diagramas de Actividad

Tiene un comportamiento dinámico, el cual sirve para visualizar, especificar, construir y documentar la dinámica de un conjunto de objetos o simplemente para modelar el flujo de control de una operación. Es utilizado en conjunción con los diagramas de Caso de Uso para que el equipo de desarrollo entienda cómo es utilizado el sistema y cómo reacciona en determinados eventos.

Elementos de un Diagrama de Actividades

- **Inicio:** El inicio del diagrama es representado por un círculo de color negro sólido.
- **Actividad:** Es la acción que será realizada por el sistema. Su representación es un ovalo.
- **Transición:** Ocurre cuando se lleva a cabo el cambio de una actividad a otra. Su representación es una línea con una flecha en su terminación para indicar dirección.

Como se puede apreciar en el ejemplo de Diagrama de Actividades en la Figura 2.9.

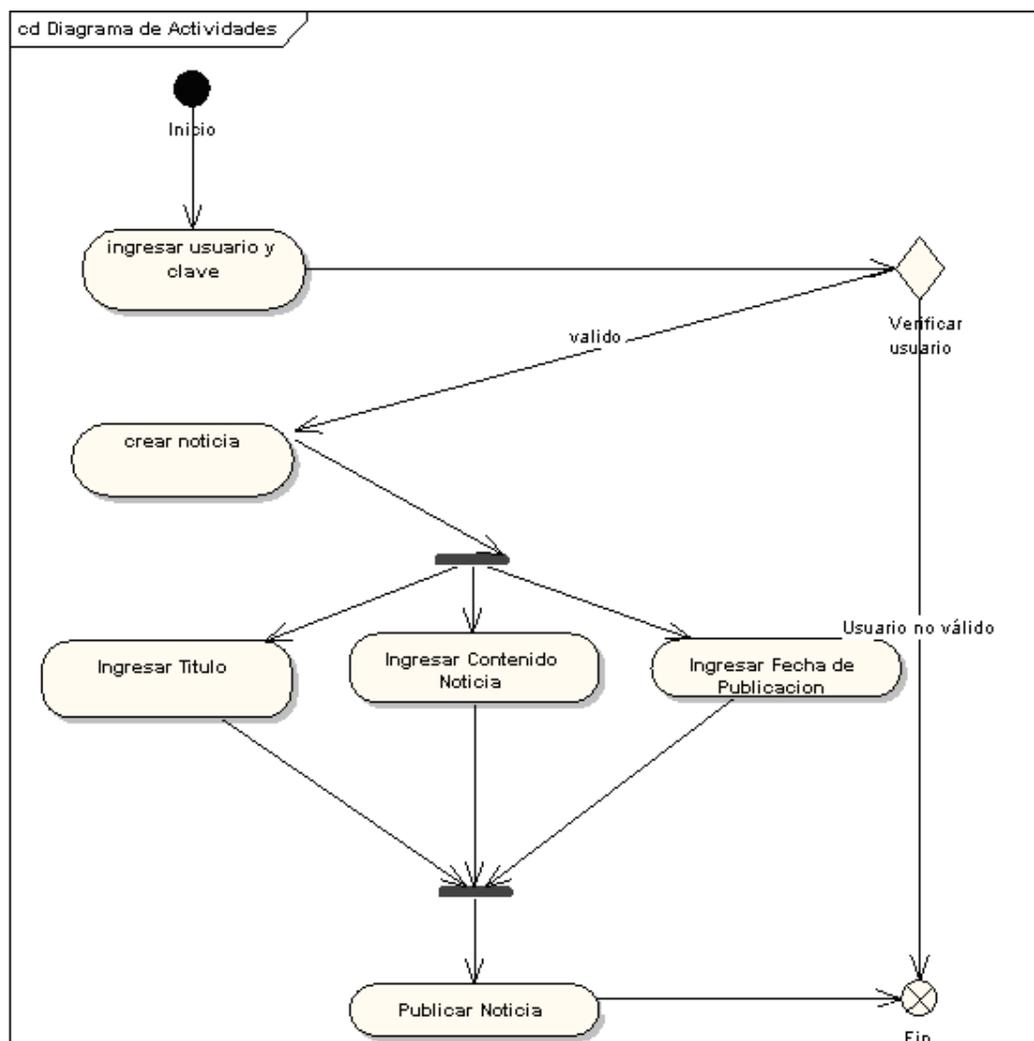


Figura 2.9 Ejemplo de un Diagrama de Actividades

2.2.2.10 Diagramas de Despliegue

Se utilizan para modelar la topología hardware del sistema, forman parte de la vista física del sistema.

Los Diagramas de Despliegue muestran la disposición física de los distintos nodos¹³ que componen un sistema y el reparto de los componentes sobre dichos nodos. Como se puede apreciar en la Figura 2.10.

Con UML, estos diagramas se utilizan para visualizar los aspectos estáticos de estos nodos físicos y sus relaciones y para especificar sus detalles para la construcción.

Usos comunes de los diagramas de despliegue:

- Para modelar sistemas: empotrados, cliente/servidor, completamente distribuidos.

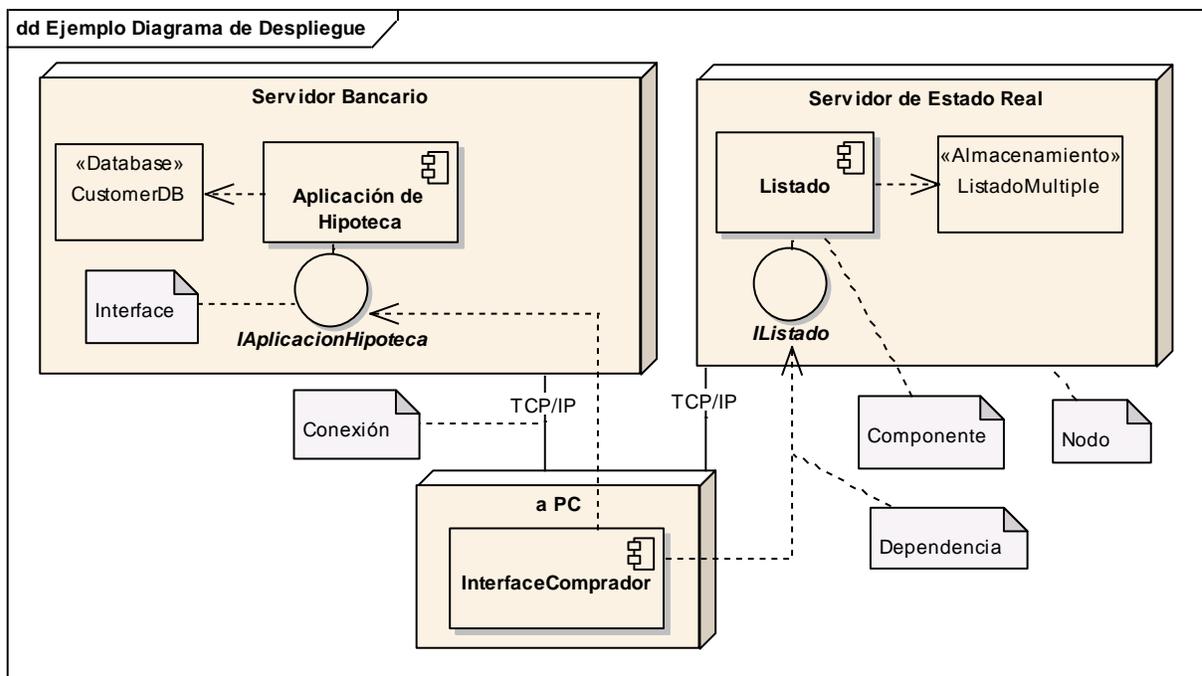


Figura 2.10 Ejemplo de un Diagrama de Despliegue

¹³ **Nodo:** Es un elemento físico que representa un recurso con capacidad computacional.

2.2.2.11 Diagramas de Componente

Sirve para representar la vista física de un sistema, la cual modela la estructura de implementación de la aplicación, su organización en componentes y su despliegue en nodos de ejecución. Esta vista establece las correspondencias entre las clases, los componentes de implementación y los nodos.

Estos diagramas describen los elementos físicos del sistema y sus relaciones:

Los componentes representan una parte física en el sistema los cuales conforman un conjunto de interfaces y proporciona una implementación de clases.

Las relaciones de dependencia se utilizan para indicar que un componente utiliza los servicios ofrecidos por otro componente.

El diagrama también puede usarse para mostrar interfaces y las dependencias de llamada entre componentes, usando flechas con líneas discontinuas desde los componentes a las interfaces de otros componentes, como se puede apreciar en la Figura 2.11.

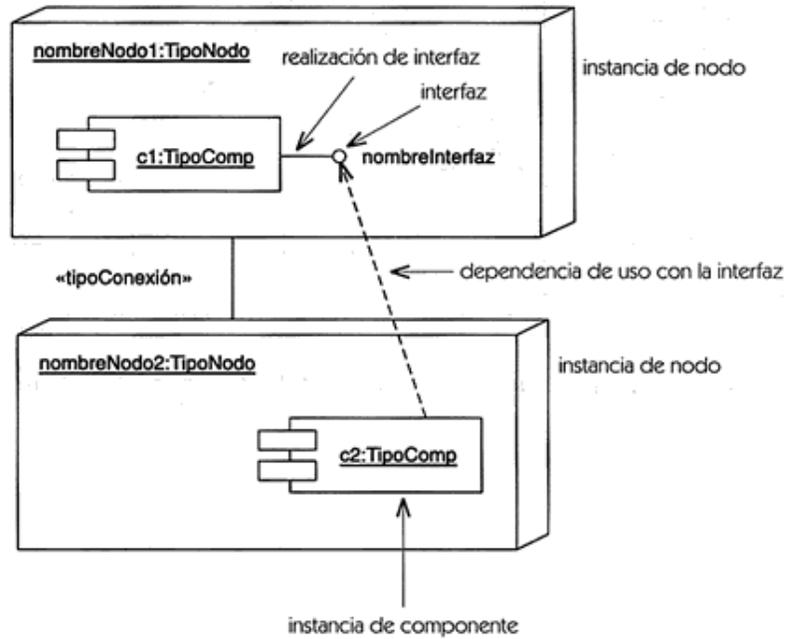


Figura 2.11 Ejemplo de un Diagrama de Componentes¹⁴

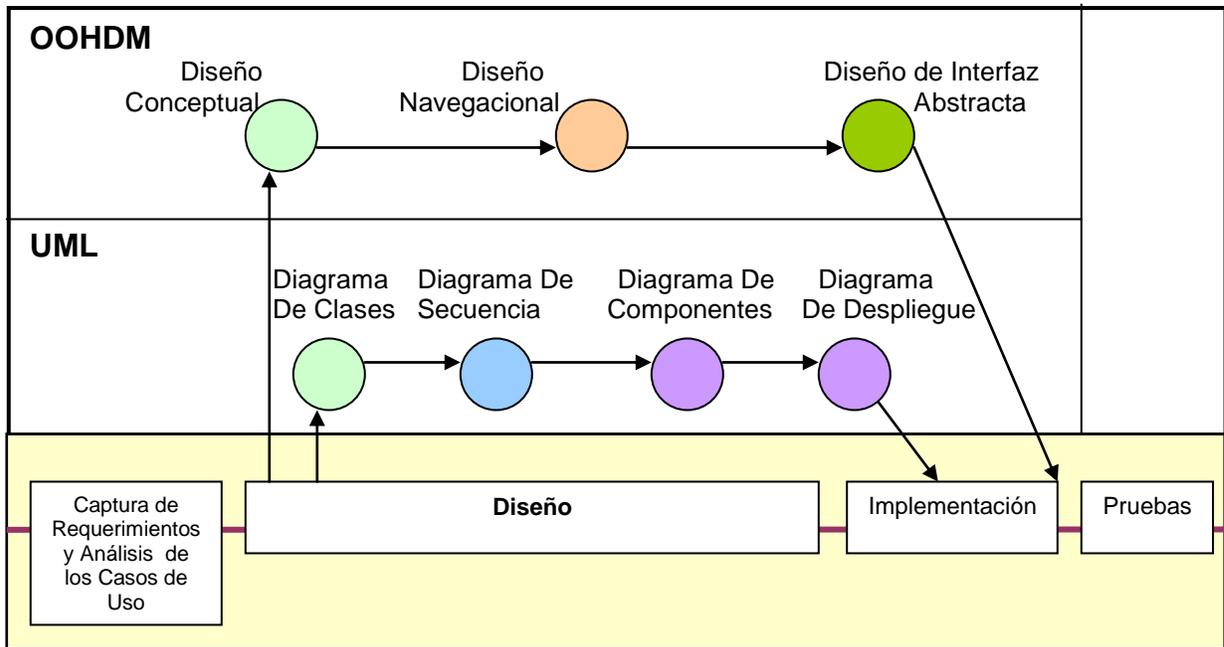
2.2.3 Metodología OOHDM y el Lenguaje de Modelado UML

Con el fin de obtener y garantizar un producto de calidad tanto visual como estructural, se utilizará la pirámide del Diseño la metodología OOHDM y el lenguaje de modelado (UML), que poseen un objetivo en común, el desarrollo de una aplicación de una forma concienzuda y metódica.

En la siguiente gráfica se muestra más claramente la relación entre OOHDM y UML

Figura 2.12:

¹⁴ Figura tomada de la página: Diagrama de Componentes, <http://www.creangel.com/uml/componente.php#>



— Línea de Vida de un Sistema Web

Figura 2.12 OHDM y UML

2.3 Indexación de Documentos

Si bien los documentos más numerosos se han almacenado ya en el Subsistema de Archivo Masivo, siempre habrá en los diferentes directorios de la red una gran cantidad de archivos de todo tipo, los que por diferentes razones, principalmente porque están en elaboración, convendrá mantener allí.

Para indexar todo tipo de documentos en una red y facilitar su recuperación, se han desarrollado los Subsistemas de Indexación de Documentos.

Un Subsistema moderno de Indexación de Documentos consiste, esencialmente, en un Índice de todos los documentos residentes en los diferentes directorios de una red que desean incorporar al mismo.

2.3.1.1 Características principales de los Subsistemas de Indexación de Documentos

- No guardan los documentos mismos, sólo los índices del texto contenido en ellos. El índice normalmente ocupa un espacio equivalente a un tercio de los documentos indexados. El índice puede estar centralizado o junto a los documentos.
- Los documentos pueden ser de cualquier tipo, siempre que tengan texto asociado, tales como:
- Documentos de aplicaciones de escritorio tales como procesadores de texto, planillas electrónicas, correo electrónico, etc.
- Documentos mixtos texto/imagen, tales como Adobe Portable Document Format (PDF) o documentos HTML y sus imágenes asociadas.
- Registros de una Base de Datos Relacional, etc.
- Los documentos se indexan por el texto ASCII que contengan.
- Los documentos se pueden visualizar directamente, o transformados a HTML, o bien abriendo la aplicación que los creó.
- Los sistemas modernos de este tipo tienen acceso Inter/Intranet.
- Pueden acceder simultáneamente diversos índices para una consulta.

Los Subsistemas de indexación no deben confundirse con los Subsistemas de Archivo. Mientras el objetivo de los primeros es la recuperación de documentos dispersos en una red, el objetivo de los segundos consiste en almacenar masivamente los documentos mismos.

A continuación se indica en el siguiente grafico 2.13, un ejemplo de indexación en donde en 3 pasos se indica básicos se explica el concepto de indexación.

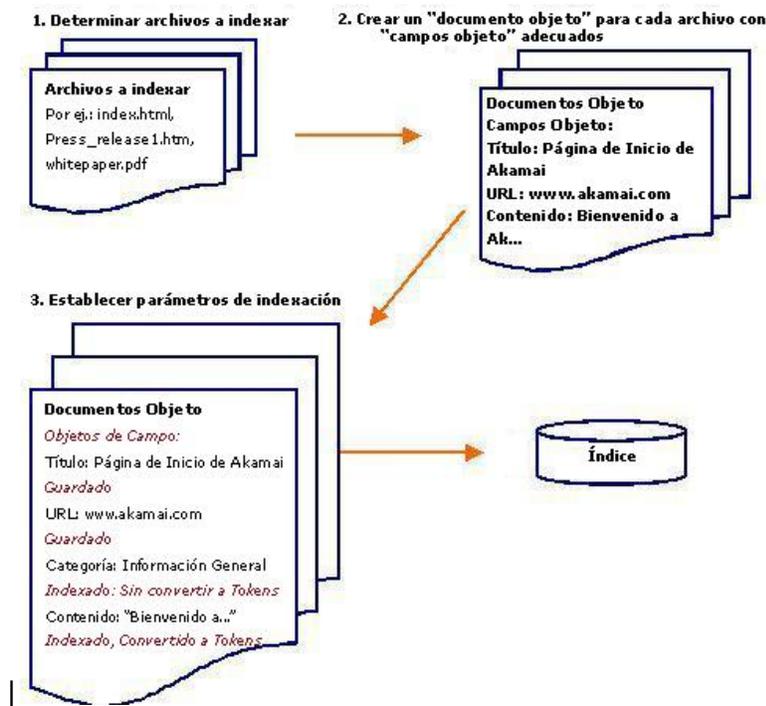


Figura 2.13 Indexación

2.4 LDAP

LDAP (Lightweight Directory Access Protocol, Protocolo Ligero de Acceso a Directorios) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) a la que pueden realizarse consultas.

Un directorio es un conjunto de objetos con atributos organizados en una manera lógica y jerárquica. El ejemplo más común es el directorio telefónico, que consiste en una serie de nombres (personas u organizaciones) que están ordenados alfabéticamente, con cada nombre teniendo una dirección y un número de teléfono adjuntos.

Un árbol de directorio LDAP a veces refleja varios límites políticos, geográficos y/o organizacionales, dependiendo del modelo elegido. Los despliegues actuales de LDAP tienden a usar nombres de Sistema de Nombres de Dominio (DNS por sus siglas en inglés) para estructurar los niveles más altos de la jerarquía. Conforme se desciende en el directorio pueden aparecer entradas que representan personas, unidades organizacionales, impresoras, documentos, grupos de personas o cualquier cosa que representa una entrada dada en el árbol (o múltiples entradas).

Habitualmente, almacena la información de autenticación (usuario y contraseña) y es utilizado para autenticarse aunque es posible almacenar otra información (datos de contacto del usuario, ubicación de diversos recursos de la red, permisos, certificados, etc). A manera de síntesis, LDAP es un protocolo de acceso unificado a un conjunto de información sobre una red.

La versión actual es LDAPv3, la cual es especificada en una serie de Internet Engineering Task Force (IETF) Standard Track Request for Comments (RFCs) como se detalla en el documento RFC 4510.

2.4.1.1 Origen de Influencias

Las compañías de telecomunicaciones introdujeron el concepto de servicios de directorio a Tecnologías de Información y Redes de Computadoras, así su comprensión de los requerimientos de directorios era bien desarrollado después de 70 años de producir y manejar directorios de teléfonos. La culminación de este esfuerzo fue la especificación X.500, un conjunto de protocolos producido por la Unión Internacional de Telecomunicaciones (ITU por sus siglas en inglés) en la década de 1980.

Los servicios de directorio X.500 fueron accedidos tradicionalmente vía DAP (Directory Access Protocol), que requería la pila de protocolos OSI (Open Systems

Interconnection). LDAP fue originalmente dirigido a ser un protocolo alternativo y ligero para acceder a servicios de directorio X.500 a través de la pila de protocolos más simple (y ahora más difundido) TCP/IP. Este modelo de acceso a directorio fue imitado de los protocolos DIXIE Directory Assistance Service.

Servidores de directorio LDAP independientes pronto fueron implementados, así como los servidores de directorio que soportaban DAP y LDAP. El último se hizo popular en empresas debido a que eliminaba cualquier necesidad de desplegar una red OSI. Ahora, los protocolos de directorio X.500 incluyendo DAP pueden ser usados directamente sobre TCP/IP.

El protocolo fue creado originalmente por Tim Howes (University of Michigan), Steve Kille (Isode Limited), y Wengyik Yeong (Performance Systems International) hacia 1993. Un desarrollo más completo ha sido hecho por la Internet Engineering Task Force.

En las primeras etapas de ingeniería de LDAP, éste era conocido como Lightweight Directory Browsing Protocol, o LDBP. Posteriormente fue renombrado dado que el ámbito del protocolo había sido expandido para incluir no sólo navegación en el directorio y funciones de búsqueda, sino también funciones de actualización de directorio.

LDAP ha influenciado protocolos posteriores de Internet, incluyendo versiones posteriores de X.500, XML Enabled Directory (XED), Directory Service Markup Language (DSML), Service Provisioning Markup Language (SPML), y Service Location Protocol (SLP).

2.4.1.2 Visión general del protocolo

Un cliente inicia una sesión de LDAP conectándose a un servidor LDAP, por defecto en el puerto TCP 389. El cliente luego envía una petición de operación al servidor, y el servidor envía respuestas. Con algunas excepciones, el cliente no necesita

esperar una respuesta antes de enviar la siguiente petición, y el servidor puede responder en cualquier orden.

El cliente puede requerir las siguientes operaciones:

- Start TLS — usar la extensión Transport Layer Security (TLS) LDAPv3 para una conexión segura
- Bind — autenticarse y especificar una versión del protocolo LDAP
- Search — buscar y/o obtener entradas de directorio
- Compare — probar si una entrada nombrada contiene un valor de atributo dado
- Add — Añadir una nueva entrada
- Delete — Borrar una entrada
- Modify — Modificar una entrada
- Modify Distinguished Name (DN) — Modificar o renombrar una entrada
- Abandon — abortar una petición previa
- Extended Operation — operación genérica usada para definir otras operaciones
- Unbind — cerrar la conexión (no es el inverso de Bind)

Además, el servidor puede enviar "notificaciones no solicitadas" que no son respuestas a ninguna petición, por ejemplo antes de que se termine el tiempo de conexión.

Un método alternativo común para asegurar las comunicaciones LDAP es usar un túnel SSL. Esto es denotado en las URLs de LDAP usando el esquema de URLs "ldaps". El puerto por defecto para LDAP sobre SSL es 636. El uso de LDAP sobre SSL fue común en LDAP Version 2 (LDAPv2) pero nunca fue estandarizado en una especificación formal. Su uso es considerado obsoleto al igual que LDAPv2, que ha sido retirado oficialmente en 2003.

LDAP es definido en términos de ASN.1, y los protocolos del mensaje están codificados en el formato binario BER. Sin embargo, utiliza representaciones textuales para un número de campos y tipos ASN.1.

2.4.1.3 Estructura de directorio

El protocolo accede a directorios LDAP, que siguen la edición de 1993 del modelo X.500:

- Un directorio es un árbol de entradas de directorio.
- Una entrada consta de un conjunto de atributos.
- Un atributo tiene un nombre (un *tipo de atributo* o *descripción de atributo*) y uno o más valores. Los atributos son definidos en un *esquema* (véase luego).
- Cada entrada tiene un identificador único: su *Nombre distinguido* (*Distinguished Name*, DN). Este consta de su *Relative Distinguished Name* (RDN) construido por algunos atributos en la entrada, seguidos del DN de la entrada del padre. Pensar en el nombre distinguido como un completo nombre de archivo y el nombre distinguido relativo como el nombre de archivo relativo en un folder.

Tener cuidado con el hecho de que un nombre distinguido puede cambiar en el tiempo de vida de una entrada, por ejemplo, cuando las entradas son movidas en el árbol. Para hacer más confiables e identificar de manera no ambigua las entradas un UUID podría ser proporcionado en el conjunto de los *atributos operacionales* de la entrada.

Una entrada puede lucirse como esta cuando es representada en el formato LDAP Data Interchange Format (LDIF) (LDAP por sí mismo es un protocolo binario):

```
dn: cn=John Doe,dc=example,dc=com
cn: John Doe
givenName: John
sn: Doe
telephoneNumber: +1 888 555 6789
telephoneNumber: +1 888 555 1232
mail: john@example.com
```

```
manager: cn=Barbara Doe,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
```

Explicación: dn es el nombre de la entrada; no es un atributo ni tampoco parte de la entrada. "cn=John Doe" es el nombre distinguido relativo, y "dc=example,dc=com" es el nombre distinguido de la entrada del padre, donde dc indica domain component (componente de dominio). Las otras líneas presentan los atributos en la entrada. Los nombres de atributos son generalmente cadenas mnemotécnicas, como "cn" para common name (nombre común), "dc" para domain component (componente de dominio), "mail" para dirección de e-mail y "sn" para surname (apellido).

Un servidor aloja un subárbol comenzando por una entrada específica, por ejemplo "dc=example,dc=com" y sus hijos. Los servidores también pueden almacenar referencias a otros servidores, con lo cual un intento de acceso a "ou=department,dc=example,dc=com" puede retornar una *referencia* o *continuación de referencia* a un servidor que aloja esa parte del árbol de directorio. El cliente luego puede contactar al otro servidor. Algunos servidores también soportan *encadenamiento* (chaining), que implica que el servidor contacta al otro servidor y devuelve el resultado al cliente.

LDAP raramente define un ordenamiento: El servidor puede devolver los valores de un atributo, los atributos en una entrada y las entradas encontradas por una operación de búsqueda en cualquier orden. Esto sigue la definición formal - una entrada es definida como un conjunto de atributos, y un atributo es un conjunto de valores, y los conjuntos no necesitan estar ordenados.

2.4.1.4 URLs de LDAP

Un formato URL de LDAP existe para descubrir qué clientes soportan en variedad de grados, y qué servidores retornan como referentes y referencias de continuación un ejemplo de URL LDAP:

```
ldap://host:port/DN?attributes?scope?filter?extensions
```

La mayoría de los componentes, que son descritos debajo, son opcionales.

- *host* es el FQDN o dirección IP del servidor LDAP donde se realiza la consulta.
- *port* es el puerto de red del servidor LDAP.
- *DN* es el nombre distinguido a usar como base de búsqueda.
- *attributes* es una lista separada con comas de atributos a devolver.
- *scope* especifica el ámbito de búsqueda y puede ser "base" (por defecto), "one" o "sub".
- *filter* es un filtro de búsqueda. Por ejemplo (*objectClass=**) como es definido en RFC 4515.
- *extensions* son extensiones al formato URL de LDAP.

Por ejemplo, "ldap://ldap.example.com/cn=John%20Doe,dc=example,dc=com" refiere a todos los usuarios en la entrada de John Doe en ldap.example.com, mientras "ldap:///dc=example,dc=com??sub?(givenName=John)" busca por la entrada en el servidor por defecto (notar el triple slash, omitiendo el host, y la marca de doble pregunta, omitiendo los atributos). Así como en otros URLs, los caracteres especiales deben ser codificados con signos de porcentaje.

Hay un esquema de URL similar y no estándar para LDAP sobre SSL, ldaps:. Esto no debe confundirse con LDAP sobre TLS, que se puede conseguir usando la operación StartTLS usando el esquema normal ldap:.

2.4.1.5 Variantes

Varias de las operaciones de servidor son dejadas al implementador o administrador para que él decida cómo serán realizadas. En consecuencia, los servidores pueden contar con soporte para una amplia variedad de escenarios.

Por ejemplo, el almacenamiento de datos en el servidor no es especificado -el servidor puede usar archivos de texto plano, bases de datos, o sólo ser una puerta de enlace para otro servidor. El control de acceso no es estándar, aunque se han

realizado trabajos en él y existen modelos comúnmente usados. Las claves de usuarios pueden ser almacenadas en sus entradas del directorio o en otro lugar. El servidor puede rechazar realizar operaciones que desee, e imponer varias limitaciones.

La mayoría de secciones de LDAP son extensibles. Por ejemplo: Uno puede definir nuevas operaciones. Los *controles* pueden modificar peticiones y respuestas, por ejemplo para solicitar resultados ordenados de búsqueda. Nuevos ámbitos de búsqueda y métodos de enlace pueden ser definidos. Los atributos pueden tener *opciones* que podrían modificar su semántica.

2.4.1.6 Otros modelos de datos

Así como LDAP ha tenido su momento cumbre, los proveedores lo han proporcionado como un protocolo de acceso a otros servicios. La implementación luego da forma a los datos para imitar el modelo LDAP/X.500, pero con cuanto rigidez este modelo es seguido varía entre implementaciones. Por ejemplo, existe software para acceder a bases de datos SQL a través de LDAP, a pesar de que LDAP no se presta para que esto ocurra.¹ Los servidores X.500 pueden soportar también LDAP.

De forma similar, los datos que fueron previamente alojados en otros tipos de almacenamiento de datos son a veces movidos a directorios LDAP. Por ejemplo, los usuarios de Unix y la información de grupos puede almacenarse en LDAP y accedidos vía módulos de PAM y NSS. LDAP es utilizado en ocasiones por otros servicios para conseguir autenticación.

2.4.1.7 Uso

2.4.1.7.1 Estructura de nombres

Desde que un servidor LDAP puede devolver referencias a otros servidores para efectuar nuevas peticiones que el servidor mismo no puede devolver, una estructura

de nombres para las entradas LDAP es requerida para que sea posible encontrar un servidor alojando un nombre distinguido dado. Desde que una estructura ya existe en el DNS, los nombres de alto nivel de los servidores a veces ofrecen nombres de DNS simulados, así como se hacía en X.500.

Si una organización tiene el nombre de dominio `example.org`, su entrada de más alto nivel en LDAP tendrá generalmente como nombre distinguido `dc=example,dc=org` (donde *dc* significa componente de dominio). Si el servidor LDAP es también denominado `ldap.example.org`, el nivel más alto de la organización de la URL del LDAP URL se convierte en `ldap://ldap.example.org/dc=example,dc=org`.

Bajo el alto nivel, los nombres de entradas generalmente reflejan la estructura organizacional interna o las necesidades, en lugar de nombres de DNS.

2.4.1.7.2 Terminología

La terminología de LDAP que puede encontrarse es en ocasiones engorrosa. Esto en parte se debe a malentendidos, otros ejemplos son debido a orígenes históricos, otros surgen cuando se usó con servicios distintos de X.500 que usan terminología distinta. Por ejemplo, "LDAP" es a veces utilizado para referirse al protocolo, otras veces para el protocolo y los datos. Un "directorío LDAP" puede ser los datos o también el punto de acceso. Un "atributo" puede ser el tipo de atributo, o los contenidos de un atributo en un directorío o la descripción de un atributo (un tipo de atributo con *opciones*). Un enlace (bind) anónimo es un método distinto de un enlace no autenticado, aunque ambos producen estados de autenticación anónima, por lo cual ambos términos son usados para ambas variantes. El atributo "uid" debe almacenar nombres de usuarios en lugar de identificadores numéricos de usuarios.

2.4.1.8 Implementaciones

Existen diversas implementaciones y aplicaciones reales del protocolo LDAP.

2.4.1.8.1 *Active Directory*

Active Directory es el nombre utilizado por Microsoft (desde Windows 2000) como almacén centralizado de información de uno de sus dominios de administración.

Un Servicio de Directorio es un depósito estructurado de la información de los diversos objetos que contiene el Active Directory, en este caso podrían ser impresoras, usuarios y equipos.

Bajo este nombre se encuentra realmente un esquema (definición de los campos que pueden ser consultados) LDAP versión 3, lo cual permite integrar otros sistemas que soporten el protocolo. En este LDAP se almacena información de usuarios, recursos de la red, políticas de seguridad, configuración, asignación de permisos, etc.

2.4.1.8.2 *Novell Directory Services*

También conocido como eDirectory es la implementación de Novell utilizada para manejar el acceso a recursos en diferentes servidores y computadoras de una red. Básicamente está compuesto por una base de datos jerárquica y orientada a objetos, que representa cada servidor, computadora, impresora, servicio, personas, etc. entre los cuales se crean permisos para el control de acceso, por medio de herencia. La ventaja de esta implementación es que corre en diversas plataformas, por lo que puede adaptarse fácilmente a entornos que utilicen más de un sistema operativo.

2.4.1.8.3 *iPlanet - Sun ONE Directory Server*

Basado en la antigua implementación de Netscape, iPlanet se desarrolló cuando AOL adquirió Netscape Communications Corporation y luego conjuntamente con Sun Microsystems comercializaron software para servidores, entre ellos el iPlanet

Directory Server, su implementación de LDAP... Actualmente se denomina Sun ONE Directory Server.

2.4.1.8.4 *OpenLDAP*

Se trata de una implementación libre del protocolo que soporta múltiples esquemas por lo que puede utilizarse para conectarse a cualquier otro LDAP.

Tiene su propia licencia, la OpenLDAP Public License. Al ser un protocolo independiente de la plataforma, varias distribuciones GNU/Linux y BSD lo incluyen, al igual que AIX, HP-UX, Mac OS X, Solaris, Windows (2000/XP) y z/OS.

OpenLDAP tiene cuatro componentes principales:

- slapd - demonio LDAP autónomo.
- slurpd - demonio de replicación de actualizaciones LDAP autónomo.
- Rutinas de biblioteca de soporte del protocolo LDAP.
- Utilidades, herramientas y clientes.

2.4.1.8.5 *Red Hat Directory Server*

Directory Server es un servidor basado en LDAP que centraliza configuración de aplicaciones, perfiles de usuarios, información de grupos, políticas así como información de control de acceso dentro de un sistema operativo independiente de la plataforma.

Forma un repositorio central para la infraestructura de manejo de identidad, Red Hat Directory Server simplifica el manejo de usuarios, eliminando la redundancia de datos y automatizando su mantenimiento.

2.4.1.8.6 *Apache Directory Server*

Apache Directory Server (ApacheDS), es un servidor de directorio escrito completamente en Java por Alex Karasulu y disponible bajo la licencia de Apache Software, es compatible con LDAPv3 certificado por el Open Group, soporta otros

protocolos de red tal como Kerberos y NTP, además provee Procedimientos Almacenados, triggers y vistas; características que están presente en las Base de Datos Relacionales pero que no estaban presentes en el mundo LDAP.

2.4.1.8.7 Open DS

Basado en los estándares LDAPv3 y DSMLv2, OpenDS surgió como un proyecto interno de SUN, aunque posteriormente se puso a disposición de la comunidad. Está desarrollado en JAVA y precisa de un entorno de ejecución (Java Runtime Environment) para funcionar. Es multiplataforma.

La primera versión estable fue liberada en julio de 2008.

2.5 Active Directory

Active Directory (AD) es el término utilizado por Microsoft para referirse a su implementación de servicio de directorio en una red distribuida de computadores. Utiliza distintos protocolos (principalmente LDAP, DNS, DHCP, kerberos, entre otros...).

Su estructura jerárquica permite mantener una serie de objetos relacionados con componentes de una red, como usuarios, grupos de usuarios, permisos y asignación de recursos y políticas de acceso.¹⁵

2.5.1.1 Estructura

Active Directory está basado en una serie de estándares llamados (X.500), aquí se encuentra una definición lógica a modo jerárquico.

Dominios y subdominios se identifican utilizando la misma notación de las zonas DNS, razón por la cual Active Directory requiere uno o más servidores DNS que permitan el direccionamiento de los elementos pertenecientes a la red, como por

¹⁵ Tomado de http://es.wikipedia.org/wiki/Active_directory#cite_note-0

ejemplo el listado de equipos conectados; y los componentes lógicos de la red, como el listado de usuarios.

Un ejemplo de la estructura descendente (o herencia), es que si un usuario pertenece a un dominio, será reconocido en todo el árbol generado a partir de ese dominio, sin necesidad de pertenecer a cada uno de los subdominios.

A su vez, los árboles pueden integrarse en un espacio común denominado bosque (que por lo tanto no comparten el mismo nombre de zona DNS entre ellos) y establecer una relación de «trust» o confianza entre ellos. De este modo los usuarios y recursos de los distintos árboles serán visibles entre ellos, manteniendo cada estructura de árbol el propio Active Directory.

2.5.1.2 Funcionamiento

Su funcionamiento es similar a otras estructuras de LDAP (Lightweight Directory Access Protocol), ya que este protocolo viene implementado de forma similar a una base de datos, la cual almacena en forma centralizada toda la información relativa a un dominio de autenticación. La ventaja que presenta esto es la sincronización presente entre los distintos servidores de autenticación de todo el dominio.

A su vez, cada uno de estos objetos tendrá atributos que permiten identificarlos en modo unívoco (por ejemplo, los usuarios tendrán campo «nombre», campo «email», etcétera, las impresoras de red tendrán campo «nombre», campo «fabricante», campo «modelo», campo "usuarios que pueden acceder", etc). Toda esta información queda almacenada en Active Directory replicándose de forma automática entre todos los servidores que controlan el acceso al dominio.

De esta forma, es posible crear recursos (como carpetas compartidas, impresoras de red, etc) y conceder acceso a estos recursos a usuarios, con la ventaja que estando todos estos objetos memorizados en Active Directory, y siendo esta lista de objetos replicada a todo el dominio de administración, los eventuales cambios serán visibles en todo el ámbito. Para decirlo en otras palabras, Active Directory es una

implementación de servicio de directorio en una red distribuida centralizada que facilita el control, la administración y la consulta de todos los elementos lógicos de una red (como pueden ser usuarios, equipos y recursos).

2.5.1.3 Intercambio entre dominios

Para permitir que los usuarios de un dominio accedan a recursos de otro dominio, Active Directory usa un *trust* (en español, relación de confianza). El trust es creado automáticamente cuando se crean nuevos dominios. Los límites del trust no son marcados por dominio, sino por el bosque al cual pertenece. Existen trust transitivos, donde los trust de Active Directory pueden ser un acceso directo (un dos dominios en árboles diferentes, transitivo, una o dos vías), bosque (transitivo, una o dos vías), reino (transitivo o no transitivo, una o dos vías), o externo (no transitivo, una o dos vías), para conectarse a otros bosques o dominios que no son de Active Directory. Active Directory usa el protocolo V5 de Kerberos, aunque también soporta NTLM y usuarios webs mediante autenticación SSL / TLS.

2.5.1.4 Confianza transitiva

Las Confianzas transitivas son confianzas automáticas de dos vías que existen entre dominios en Active Directory. Se definen como:

2.5.1.4.1 Confianza explícita

Las *Confianzas explícitas* son aquellas que establecen las relaciones de forma manual para entregar una ruta de acceso para la autenticación. Este tipo de relación puede ser de una o dos vías, dependiendo de la aplicación.

Las Confianzas explícitas se utilizan con frecuencia para acceder a dominios compuestos por ordenadores con Windows NT 4.0.

2.5.1.4.2 Confianza de Acceso Directo

La *Confianza de acceso directo* es, esencialmente, una confianza explícita que crea accesos directos entre dos dominios en la estructura de dominios. Este tipo de relaciones permite incrementar la conectividad entre dos dominios, reduciendo las consultas y los tiempos de espera para la autenticación.

2.5.1.4.3 Confianza entre bosques

La *Confianza entre bosques* permite la interconexión entre bosques de dominios, creando relaciones transitivas de doble vía. En Windows 2000, las confianzas entre bosques son de tipo explícito, al contrario de Windows Server 2003.

2.5.1.5 Direccionamientos a recursos

Los direccionamientos a recursos de Active Directory son estándares con la Convención Universal de Nombrado (UNC), Localizador Uniforme de Recursos (URL) y nombrado de LDAP.

Cada objeto de la red posee un nombre de distinción (en inglés, Distinguished name (DN)), así una impresora llamada *Imprime* en una Unidad Organizativa (en inglés, Organizational Units, OU) llamada *Ventas* y un dominio *foo.org*, puede escribirse de las siguientes formas para ser direccionado:

- en DN sería *CN=Imprime,OU=Ventas,DC=foo,DC=org*, donde
 - CN es el nombre común (en inglés, Common Name)
 - DC es clase de objeto de dominio (en inglés, Domain object Class).
- En forma canónica sería *foo.org/Ventas/Imprime*

Los otros métodos de direccionamiento constituyen una forma local de localizar un recurso

- Distinción de Nombre Relativo (en inglés, Relative Distinguished Name (RDN)), que busca un recurso sólo con el Nombre Común (CN).

- Globally Unique Identifier (GUID), que genera una cadena de 128 bits que es usado por Active Directory para buscar y replicar información

Ciertos tipos de objetos poseen un Nombre de Usuario Principal (en inglés, User Principal Name (UPN)) que permite el ingreso abreviado a un recurso o un directorio de la red. Su forma es *objetodered@dominio*

2.5.1.6 Diferencias entre Windows NT y Active Directory

A diferencia del anterior sistema de administración de dominios de Windows NT Server, que preveía únicamente el dominio de administración, Active Directory permite también crear estructuras jerárquicas de dominios y subdominios, facilitando la estructuración de los recursos según su localización o función dentro de la organización a la que sirven. Otra diferencia importante es el uso de estándares como X.500 y LDAP para el acceso a la información.

2.5.1.7 Interfaces de programación

Las interfaces de servicio de Active Directory (ADSI) entregan al programador una interfaz orientada a objetos, facilitando la creación de programas de directorios mediante algunas herramientas compatibles con lenguajes de alto nivel, como Visual Basic, sin tener que lidiar con los distintos espacios de nombres.

Mediante las ADSI se permite crear programas que realizan un único acceso a varios recursos del entorno de red, sin importar si están basados en LDAP u otro protocolo. Además, permite generar secuencias de comandos para los administradores.

También se puede desarrollar la Interfaz de mensajería (MAPI), que permite generar programas MAPI.

2.5.1.8 Requisitos de instalación

Para crear un dominio hay que cumplir, por lo menos, con los siguientes requisitos recomendados:

- Tener cualquier versión Server de Windows 2000 o 2003 (Server, Advanced Server o Datacenter Server), en el caso de 2003 server, tener instalado el service pack 1 en la máquina.
- Protocolo TCP/IP instalado y configurado manualmente, es decir, sin contar con una dirección asignada por DHCP,
- Tener un servidor de nombre de DNS, para resolver la dirección de los distintos recursos físicos presentes en la red
- Poseer más de 250 MB en una unidad de disco formateada en NTFS.

2.5.1.9 Alternativas

Samba es un programa de código libre, que tiene disponible un controlador de dominios compatible con Windows NT 4.

El programa de código libre Mandriva Directory Server ofrece una interfaz web para manejar el controlador de dominios de Samba y el servicio de directorios de LDAP.

Otra alternativa es Novell eDirectory, que es Multiplataforma: se puede correr sobre cualquier sistema operativo: Linux, AIX, Solaris, Novell Netware, UNIX e integra LDAP v.3 Nativo. Es el precursor en materia de estructuras de Directorio, ya que fue introducido en 1990 con la versión de Novell Netware 4.0. Aunque AD de Microsoft alcanzó mayor popularidad, todavía no puede igualar la fiabilidad y calidad de eDirectory y su capacidad Multiplataforma.

Sun Java ES Directory Server y OpenDS son otras alternativas basadas en java. El primero es un producto de Sun Microsystems y el segundo una alternativa de código abierto.

Una alternativa que integra OpenLDAP, Heimdal kerberos, Samba y además certificación digital y Bind9 (modificado para usar LDAP como backend) es WBSAgnitio.

2.6 Información general y conceptual sobre .NET Framework

.NET Framework es un componente integral de Windows que admite la creación y la ejecución de la siguiente generación de aplicaciones y servicios Web XML. El diseño de .NET Framework está enfocado a cumplir los objetivos siguientes:

- Proporcionar un entorno coherente de programación orientada a objetos, en el que el código de los objetos se pueda almacenar y ejecutar de forma local, ejecutar de forma local pero distribuida en Internet o ejecutar de forma remota.
- Proporcionar un entorno de ejecución de código que reduzca lo máximo posible la implementación de software y los conflictos de versiones.
- Ofrecer un entorno de ejecución de código que fomente la ejecución segura del mismo, incluso del creado por terceras personas desconocidas o que no son de plena confianza.
- Proporcionar un entorno de ejecución de código que elimine los problemas de rendimiento de los entornos en los que se utilizan secuencias de comandos o intérpretes de comandos.
- Ofrecer al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en el Web.
- Basar toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se puede integrar con otros tipos de código.

.NET Framework contiene dos componentes principales: Common Language Runtime y la biblioteca de clases de .NET Framework. Common Language Runtime es el fundamento de .NET Framework. El motor en tiempo de ejecución se puede considerar como un agente que administra el código en tiempo de ejecución y proporciona servicios centrales, como la administración de memoria, la

administración de subprocesos y la interacción remota, al tiempo que aplica una seguridad estricta a los tipos y otras formas de especificación del código que fomentan su seguridad y solidez. De hecho, el concepto de administración de código es un principio básico del motor en tiempo de ejecución. El código destinado al motor en tiempo de ejecución se denomina código administrado, a diferencia del resto de código, que se conoce como código no administrado. La biblioteca de clases, el otro componente principal de .NET Framework, es una completa colección orientada a objetos de tipos reutilizables que se pueden emplear para desarrollar aplicaciones que abarcan desde las tradicionales herramientas de interfaz gráfica de usuario (GUI) o de línea de comandos hasta las aplicaciones basadas en las innovaciones más recientes proporcionadas por ASP.NET, como los formularios Web Forms y los servicios Web XML.

.NET Framework puede alojarse en componentes no administrados que cargan Common Language Runtime en sus procesos e inician la ejecución de código administrado, con lo que se crea un entorno de software en el que se pueden utilizar características administradas y no administradas. En .NET Framework no sólo se ofrecen varios hosts de motor en tiempo de ejecución, sino que también se admite el desarrollo de estos hosts por parte de terceros.

Por ejemplo, ASP.NET aloja el motor en tiempo de ejecución para proporcionar un entorno de servidor escalable para el código administrado. ASP.NET trabaja directamente con el motor en tiempo de ejecución para habilitar aplicaciones de ASP.NET y servicios Web XML, que se tratan más adelante en este tema.

Internet Explorer es un ejemplo de aplicación no administrada que aloja el motor en tiempo de ejecución (en forma de una extensión de tipo MIME). Al usar Internet Explorer para alojar el motor en tiempo de ejecución, puede incrustar componentes administrados o controles de Windows Forms en documentos HTML. Al alojar el motor en tiempo de ejecución de esta manera se hace posible el uso de código móvil administrado (similar a los controles de Microsoft® ActiveX®), pero con mejoras

significativas que sólo el código administrado puede ofrecer, como la ejecución con confianza parcial y el almacenamiento aislado de archivos.

En la ilustración siguiente se muestra la relación de Common Language Runtime y la biblioteca de clases con las aplicaciones y el sistema en su conjunto. En la ilustración se representa igualmente cómo funciona el código administrado dentro de una arquitectura mayor.

Como se puede apreciar en la Figura 2.14 se explica los componentes del .Net Framework.

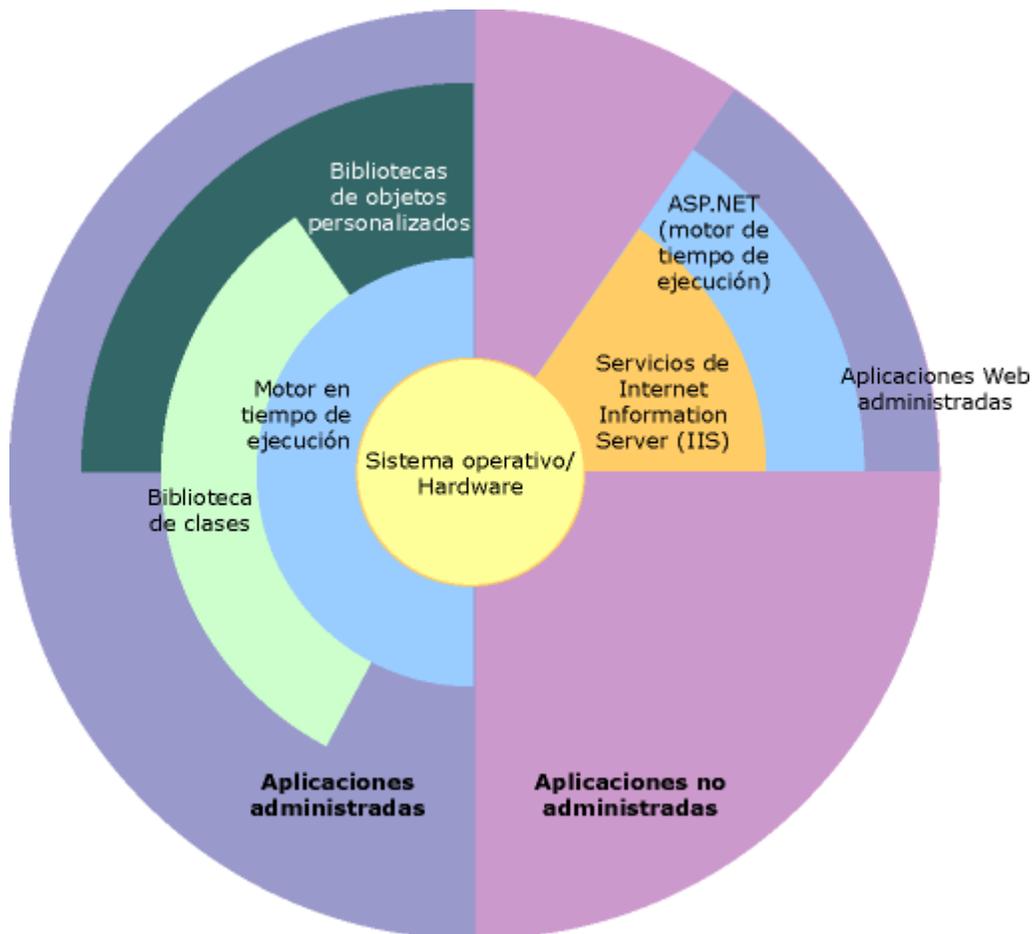


Figura 2.14 Componentes .Net Framework

En las secciones siguientes se describen con más detalle los componentes y características principales de .NET Framework.

2.6.1.1 Características de Common Language Runtime

Common Language Runtime administra la memoria, ejecución de subprocesos, ejecución de código, comprobación de la seguridad del código, compilación y demás servicios del sistema. Estas características son intrínsecas del código administrado que se ejecuta en Common Language Runtime.

Con respecto a la seguridad, los componentes administrados reciben grados de confianza diferentes, en función de una serie de factores entre los que se incluye su origen (como Internet, red empresarial o equipo local). Esto significa que un componente administrado puede ser capaz o no de realizar operaciones de acceso a archivos, operaciones de acceso al Registro y otras funciones delicadas, incluso si se está utilizando en la misma aplicación activa.

El motor en tiempo de ejecución impone seguridad en el acceso al código. Por ejemplo, los usuarios pueden confiar en que un archivo ejecutable incrustado en una página Web puede reproducir una animación en la pantalla o entonar una canción, pero no puede tener acceso a sus datos personales, sistema de archivos o red. Por ello, las características de seguridad del motor en tiempo de ejecución permiten que el software legítimo implementado en Internet sea excepcionalmente variado.

El motor en tiempo de ejecución impone la solidez del código mediante la implementación de una infraestructura estricta de comprobación de tipos y código denominada CTS (Common Type System, Sistema de tipos común). CTS garantiza que todo el código administrado es autodescriptivo. Los diferentes compiladores de lenguajes de Microsoft y de terceros generan código administrado que se ajusta a CTS. Esto significa que el código administrado puede usar otros tipos e instancias administrados, al tiempo que se aplica inflexiblemente la fidelidad y seguridad de los tipos.

También, el entorno administrado del motor en tiempo de ejecución elimina muchos problemas de software comunes. Por ejemplo, el motor en tiempo de ejecución controla automáticamente la disposición de los objetos, administra las referencias a éstos y los libera cuando ya no se utilizan. Esta administración automática de la memoria soluciona los dos errores más comunes de las aplicaciones: la pérdida de memoria y las referencias no válidas a la memoria.

Igualmente, el motor en tiempo de ejecución aumenta la productividad del programador. Por ejemplo, los desarrolladores pueden crear aplicaciones en el lenguaje que prefieran y seguir sacando todo el provecho del motor en tiempo de ejecución, la biblioteca de clases y los componentes escritos en otros lenguajes por otros colegas. El proveedor de un compilador puede elegir destinarlo al motor en tiempo de ejecución. Los compiladores de lenguajes que se destinan a .NET Framework hacen que las características de .NET Framework estén disponibles para el código existente escrito en dicho lenguaje, lo que facilita enormemente el proceso de migración de las aplicaciones existentes.

Aunque el motor en tiempo de ejecución está diseñado para el software del futuro, también es compatible con el software actual y el software antiguo. La interoperabilidad entre el código administrado y no administrado permite que los desarrolladores continúen utilizando los componentes COM y las DLL que necesiten.

El motor en tiempo de ejecución está diseñado para mejorar el rendimiento. Aunque Common Language Runtime proporciona muchos servicios estándar de motor en tiempo de ejecución, el código administrado nunca se interpreta. Una característica denominada compilación JIT (Just-In-Time) permite ejecutar todo el código administrado en el lenguaje máquina nativo del sistema en el que se ejecuta. Mientras tanto, el administrador de memoria evita que la memoria se pueda fragmentar y aumenta la zona de referencia de la memoria para mejorar aún más el rendimiento.

Por último, el motor en tiempo de ejecución se puede alojar en aplicaciones de servidor de gran rendimiento, como Microsoft® SQL Server™ e Internet Information

Services (IIS). Esta infraestructura permite utilizar código administrado para escribir lógica empresarial, al tiempo que se disfruta del superior rendimiento de los mejores servidores empresariales del sector que puedan alojar el motor en tiempo de ejecución.

2.6.1.2 Biblioteca de clases de .NET Framework

La biblioteca de clases de .NET Framework es una colección de tipos reutilizables que se integran estrechamente con Common Language Runtime. La biblioteca de clases está orientada a objetos, lo que proporciona tipos de los que su propio código administrado puede derivar funciones. Esto ocasiona que los tipos de .NET Framework sean sencillos de utilizar y reduce el tiempo asociado con el aprendizaje de las nuevas características de .NET Framework. Además, los componentes de terceros se pueden integrar sin dificultades con las clases de .NET Framework.

Por ejemplo, las clases de colección de .NET Framework implementan un conjunto de interfaces que puede usar para desarrollar sus propias clases de colección. Éstas se combinarán fácilmente con las clases de .NET Framework.

Como en cualquier biblioteca de clases orientada a objetos, los tipos de .NET Framework permiten realizar diversas tareas de programación comunes, como son la administración de cadenas, recolección de datos, conectividad de bases de datos y acceso a archivos. Además de estas tareas habituales, la biblioteca de clases incluye tipos adecuados para diversos escenarios de desarrollo especializados.

2.6.1.3 Common Language Runtime

Los compiladores y las herramientas exponen la funcionalidad en tiempo de ejecución y permiten escribir código con las ventajas que proporciona este entorno de ejecución administrado. El código desarrollado con un compilador de lenguaje orientado al tiempo de ejecución se denomina código administrado. Este código se beneficia de características como: la integración entre lenguajes, el control de excepciones entre lenguajes, la seguridad mejorada, la compatibilidad con la

implementación y las versiones, un modelo simplificado de interacción y servicios de creación de perfiles y depuración.

Para permitir al motor en tiempo de ejecución proporcionar servicios al código administrado, los compiladores de lenguajes deben emitir metadatos que describen los tipos, los miembros y las referencias del código. Los metadatos se almacenan con el código; cada archivo ejecutable portable (PE) de Common Language Runtime cargable contiene metadatos. El motor en tiempo de ejecución utiliza los metadatos para localizar y cargar clases, colocar instancias en memoria, resolver invocaciones a métodos, generar código nativo, exigir mecanismos de seguridad y establecer los límites del contexto en tiempo de ejecución.

El tiempo de ejecución controla automáticamente la disposición de los objetos y administra las referencias a éstos, liberándolos cuando ya no se utilizan. Los objetos cuya duración se administra de esta forma se denominan datos administrados. La recolección de elementos no utilizados elimina pérdidas de memoria así como otros errores habituales de programación. Con un código administrado se pueden utilizar datos administrados, datos no administrados o estos dos tipos de datos en una aplicación .NET. Framework. Como los compiladores de lenguajes proporcionan sus propios tipos, como tipos primitivos, no siempre se sabe (o no es necesario saber) si los datos se están administrando.

Common Language Runtime facilita el diseño de los componentes y de las aplicaciones cuyos objetos interactúan entre lenguajes distintos. Los objetos escritos en lenguajes diferentes pueden comunicarse entre sí, lo que permite integrar sus comportamientos de forma precisa. Por ejemplo, puede definir una clase y, a continuación, utilizar un lenguaje diferente para derivar una clase de la clase original o llamar a un método de la clase original. También se puede pasar al método de una clase una instancia de una clase escrita en un lenguaje diferente. Esta integración entre lenguajes diferentes es posible porque los compiladores y las herramientas de lenguajes orientados al motor en tiempo de ejecución utilizan un sistema de tipos común definido por el motor en tiempo de ejecución, y los lenguajes siguen las reglas

en tiempo de ejecución para definir nuevos tipos, así como para crear, utilizar, almacenar y enlazar tipos.

Como parte de los metadatos, todos los componentes administrados contienen información sobre los componentes y los recursos utilizados en su creación. El motor en tiempo de ejecución utiliza esta información para garantizar que el componente o la aplicación contiene las versiones especificadas de todo lo necesario, por lo que hay menos posibilidades de que la ejecución del código se interrumpa debido a una dependencia inadecuada. La información de registro y los datos de estado ya no se almacenan en el Registro, donde puede ser difícil establecer y mantener datos. En su lugar, la información sobre tipos definidos por el usuario (y sus dependencias) se almacena con el código como metadatos y, de este modo, las tareas de réplica y eliminación de componentes es mucho menos complicada.

Las herramientas y los compiladores de lenguajes exponen la funcionalidad del motor en tiempo de ejecución de forma que resulte útil e intuitiva para los programadores. Esto significa que algunas características en tiempo de ejecución pueden ser más evidentes en un entorno que en otro. El funcionamiento del motor en tiempo de ejecución depende de las herramientas y los compiladores utilizados. Por ejemplo, un programador de Visual Basic observará que con Common Language Runtime, el lenguaje Visual Basic contiene más características orientadas a objetos que antes. Algunas de las ventajas del motor en tiempo de ejecución son:

- Mejoras en el rendimiento.
- Capacidad para utilizar fácilmente componentes desarrollados en otros lenguajes.
- Tipos extensibles que proporciona una biblioteca de clases
- Nuevas características del lenguaje como herencia, interfaces y sobrecarga para la programación orientada a objetos; compatibilidad con el uso de subprocesos libres que permite la creación de multiprocesos; aplicaciones escalables; compatibilidad con los atributos personalizados y el control de excepciones estructurado.

Si utiliza Microsoft® Visual C++® .NET, puede escribir código administrado utilizando Visual C++, que proporcionan las ventajas de un entorno de ejecución administrado, así como el acceso a características eficaces y a tipos de datos informativos que ya le resultan familiares. Otras características del motor en tiempo de ejecución son:

- Integración entre lenguajes diferentes y, en especial, herencia entre lenguajes.
- Recolección de elementos no utilizados, que administra la duración de los objetos de modo que no es necesario el recuento de referencias.
- Objetos autodescriptivos que hacen innecesario el Lenguaje de definición de interfaces (IDL).
- Capacidad para compilar una vez y ejecutar código en cualquier CPU y sistema operativo que sea compatible con el motor en tiempo de ejecución.

También se puede escribir código administrado en lenguaje C#, que proporciona las siguientes ventajas:

- Diseño completo orientado a objetos
- Seguridad de tipos muy sólida
- Buena combinación entre la simplicidad de Visual Basic y la eficacia de C++.
- Recolección de elementos no utilizados.
- Sintaxis y palabras clave similares en C y en C++.
- Utilice delegados para una mayor seguridad y protección de tipos, en vez de punteros a funciones. Los punteros a funciones están disponibles mediante la utilización de la palabra clave **unsafe** de C# y de la opción **/unsafe** del compilador de C# (Csc.exe) para datos y código no administrados.

2.6.1.4 C# y .NET Framework

C# es un lenguaje orientado a objetos elegante y con seguridad de tipos que permite a los desarrolladores generar diversas aplicaciones sólidas y seguras que se ejecutan en .NET Framework. Puede utilizar este lenguaje para crear aplicaciones cliente para Windows tradicionales, servicios web XML, componentes distribuidos,

aplicaciones cliente-servidor, aplicaciones de base de datos, y muchas tareas más. Visual C# 2008 proporciona un editor de código avanzado, diseñadores de interfaz de usuario prácticos, un depurador integrado y muchas otras herramientas para facilitar el desarrollo de aplicaciones basado en la versión 3.0 del lenguaje C# y en la versión 3.5 de .NET Framework.

2.6.1.5 Lenguaje C#

La sintaxis de C# es muy expresiva, pero también es sencilla y fácil de aprender. La sintaxis de C# basada en signos de llave podrá ser reconocida inmediatamente por cualquier persona familiarizada con C, C++ o Java. Los desarrolladores que conocen cualquiera de estos lenguajes pueden empezar a trabajar de forma productiva en C# en un plazo muy breve. La sintaxis de C# simplifica muchas de las complejidades de C++ y proporciona características eficaces tales como tipos de valores que admiten valores NULL, enumeraciones, delegados, expresiones lambda y acceso directo a memoria, que no se encuentran en Java. C# admite métodos y tipos genéricos, que proporcionan mayor rendimiento y seguridad de tipos, e iteradores, que permiten a los implementadores de clases de colección definir comportamientos de iteración personalizados que el código cliente puede utilizar fácilmente. En C# 3.0, las expresiones Language-Integrated Query (LINQ) convierten la consulta con establecimiento inflexible de tipos en una construcción de lenguaje de primera clase.

Como lenguaje orientado a objetos, C# admite los conceptos de encapsulación, herencia y polimorfismo. Todas las variables y métodos, incluido el método **Main** que es el punto de entrada de la aplicación, se encapsulan dentro de definiciones de clase. Una clase puede heredar directamente de una clase primaria, pero puede implementar cualquier número de interfaces. Los métodos que reemplazan a los métodos virtuales en una clase primaria requieren la palabra clave **override** como medio para evitar redefiniciones accidentales. En C#, una estructura es como una clase sencilla; es un tipo asignado en la pila que puede implementar interfaces pero que no admite la herencia.

Además de estos principios básicos orientados a objetos, C# facilita el desarrollo de componentes de software a través de varias construcciones de lenguaje innovadoras, entre las que se incluyen las siguientes:

- Firmas de métodos encapsulados denominadas *delegados*, que habilitan notificaciones de eventos con seguridad de tipos.
- Propiedades, que actúan como descriptores de acceso para variables miembro privadas.
- Atributos, que proporcionan metadatos declarativos sobre tipos en tiempo de ejecución.
- Comentarios en línea de documentación XML.
- Language-Integrated Query (LINQ) que proporciona funciones de consulta integradas en una gran variedad de orígenes de datos.

Si necesita interactuar con otro software de Windows, como objetos COM o archivos DLL nativos de Win32, podrá hacerlo en C# mediante un proceso denominado "interoperabilidad". La interoperabilidad habilita los programas de C# para que puedan realizar prácticamente las mismas tareas que una aplicación C++ nativa. C# admite incluso el uso de punteros y el concepto de código "no seguro" en los casos en que el acceso directo a la memoria es totalmente crítico.

El proceso de generación de C# es simple en comparación con el de C y C++, y es más flexible que en Java. No hay archivos de encabezado independientes, ni se requiere que los métodos y los tipos se declaren en un orden determinado. Un archivo de código fuente de C# puede definir cualquier número de clases, estructuras, interfaces y eventos.

2.6.1.6 Arquitectura de la plataforma .NET Framework

Los programas de C# se ejecutan en .NET Framework, un componente que forma parte de Windows y que incluye un sistema de ejecución virtual denominado Common Language Runtime (CLR) y un conjunto unificado de bibliotecas de clases. CLR es la implementación comercial de Microsoft de la Common Language

Infrastructure (CLI), norma internacional que constituye la base para crear entornos de ejecución y desarrollo en los que los lenguajes y las bibliotecas trabajan juntos sin problemas.

El código fuente escrito en C# se compila en un lenguaje intermedio (IL) conforme con la especificación CLI. El código de lenguaje intermedio y recursos tales como mapas de bits y cadenas se almacenan en disco en un archivo ejecutable denominado ensamblado, cuya extensión es .exe o .dll generalmente. Un ensamblado contiene un manifiesto que proporciona información sobre los tipos, la versión, la referencia cultural y los requisitos de seguridad del ensamblado.

Cuando se ejecuta un programa de C#, el ensamblado se carga en CLR, con lo que se pueden realizar diversas acciones en función de la información del manifiesto. A continuación, si se cumplen los requisitos de seguridad, CLR realiza una compilación Just In Time (JIT) para convertir el código de lenguaje intermedio en instrucciones máquina nativas. CLR también proporciona otros servicios relacionados con la recolección automática de elementos no utilizados, el control de excepciones y la administración de recursos. El código ejecutado por CLR se denomina algunas veces "código administrado", en contraposición al "código no administrado" que se compila en lenguaje máquina nativo destinado a un sistema específico. En el diagrama siguiente se muestran las relaciones en tiempo de compilación y tiempo de ejecución de los archivos de código fuente de C#, las bibliotecas de clases de .NET Framework, los ensamblados y CLR.

A continuación un Diagrama de la Ejecución de C# en la Figura 2.15.

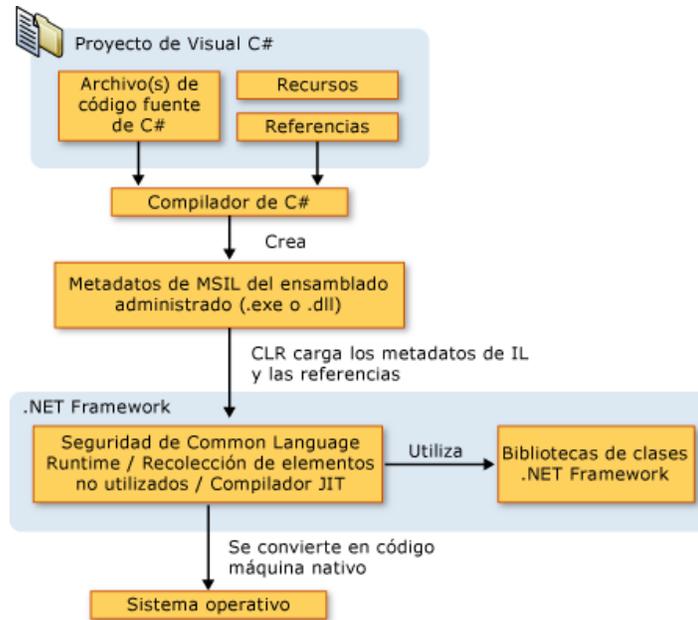


Figura 2.15 Ejecución de C#

La interoperabilidad del lenguaje es una característica clave de .NET Framework. Como el código de lenguaje intermedio generado por el compilador de C# cumple la especificación de tipos común (CTS), este código generado en C# puede interactuar con el código generado en las versiones .NET de Visual Basic, Visual C++, Visual J# o cualquiera de los más de 20 lenguajes conformes con CTS. Un único ensamblado puede contener varios módulos escritos en diferentes lenguajes .NET, y los tipos admiten referencias entre sí como si estuvieran escritos en el mismo lenguaje.

Además de los servicios en tiempo de ejecución, .NET Framework también incluye una amplia biblioteca de más de 4.000 clases organizadas en espacios de nombres que proporcionan una gran variedad de funciones útiles para la entrada y salida de archivos, la manipulación de cadenas, el análisis XML, los controles de los formularios Windows Forms y muchas tareas más. La aplicación de C# típica utiliza continuamente la biblioteca de clases de .NET Framework para el tratamiento de las tareas comunes de "infraestructura".

CAPÍTULO III

DESARROLLO DEL SISTEMA SADO

3.1 Introducción

A continuación en este Capítulo se realizara el análisis FODA de la empresa y se obtendrá los requerimientos para diseñar el aplicativo, con la ayuda de la Metodología OOHDM y el Lenguaje de Modelado UML.

3.2 Antecedentes

Soluciones Inteligentes es una empresa de desarrollo de software que posee la necesidad de prestar a sus clientes un sistema de digitalización masiva, para evitar el trabajo manual que se ha venido realizando.

3.2 Análisis de la Situación Actual

Actualmente el proceso de preservación y archivo de documentación para la empresa Soluciones Inteligentes se realiza manualmente por lo cual no se garantiza que este expuesto a modificaciones, de igual manera el realizar este proceso de almacenaje físico requiere de demasiado tiempo el momento de clasificar cada uno de los documentos por categoría y fecha.

Los documentos se preservan en un histórico físico, pero no garantiza la seguridad requerida para evitar el deterioro de estos documentos, por lo cual surge la necesidad de implementar este sistema.

Anteriormente Soluciones Inteligentes quiso dar un control adecuado al manejo y administración de los documentos, por lo cual contrató la asesoría de una empresa, la que recomendó la compra de la licencia Adobe Capture pero no tomaron en cuenta el limitante que presenta este producto que solo sirve para digitalizar los documentos y no les ofrece la administración, búsqueda y seguridad.

Adicionalmente la empresa que prestó asesoría a Soluciones Inteligentes jamás realizó la instalación y/o configuración del software, y el proyecto con el capture no se llegó a culminar.

Tomando en cuenta el antecedente anterior Soluciones Inteligentes decide contactarnos para que se realice la consultoría sobre esta solución y posteriormente encarga el diseño y desarrollo de la aplicación en base a los requerimientos definidos en su proyecto original de digitalización de Documentos.

3.3 Estrategias nivel de plan de negocio

3.3.1 Análisis FODA

En la siguiente tabla se muestra el análisis FODA que se aplicó a la empresa Soluciones Inteligentes.

FORTALEZAS	DEBILIDADES
<ul style="list-style-type: none"> • Infraestructura adecuada y equipamiento y tecnología de última generación. • La empresa posee un gran prestigio por el servicio entregado a clientes. • Excelencia y compromiso con los clientes. • Nivel adecuado de recursos Financieros. • Tecnología Propia. • Soporte técnico ofrecido antes de la venta y después de la venta. • Procesos estandarizados y normados. • Se cuenta con un grupo de criterios y estándares nacionales para establecer calidad , competitividad e impacto social 	<ul style="list-style-type: none"> • Estancarse con un solo producto y no ofrecer diversidad a los clientes. • Falta un sistema que administre y gestión gastos. • Falta de alianzas estratégicas y convenio con los competidores que beneficien. • La infraestructura posiblemente colapse con el aumento laboral, no sea la adecuada. • No posee un área bien estructura de marketing. • No posee una cadena de valor. • No tiene personal preparado para un Sistema de Digitalización.
OPORTUNIDADES	AMENAZAS
<ul style="list-style-type: none"> • Tendencia por parte de las empresas al uso de IT (Tecnologías de la información) • Abrirse puertas a empresas del sector público con un producto innovador. • Ahorro de tiempo y recursos en el almacenamiento y búsqueda de documentos. • La mayoría de las empresas públicas cuentan con infraestructura obsoleta y anticuada. • Poseer gran variedad de sistemas que puedan satisfacer las necesidades del cliente. • La baja creación de competencia por las barreras de entrada. 	<ul style="list-style-type: none"> • Dependencias tecnológicas por parte de potencias mundiales. • Presencia de servicios y productos sustitutos. • Existe empresas de competencia en desarrollo de software. • Ingreso de competidores con estructuras de costos menores. • Contratar asesoría para un tipo de proyecto específico. • Presupuesto mínimo por parte de las empresas para la inversión en el área IT.

Tabla 3.1 Análisis de FODA

3.3.2 Matriz de Resumen FODA

A continuación se indica las matrices de resumen basadas en el análisis FODA que permitirá definir las estrategias a aplicar:

MATRICES DE RESUMEN				
#	MATRICES RESUMEN E IMPACTO DE ASPECTOS ESTRATEGICOS JERARQUIZADOS	IMPACTO		
	FORTALEZAS	ALTO	MEDIO	BAJO
	1	Infraestructura adecuada y equipamiento y tecnología de última generación.	X	
2	La empresa posee un gran prestigio a través de servicio entregado a clientes	X		
3	Excelencia y compromiso con los clientes	X		
4	Nivel adecuado de recursos Financieros		X	
5	Tecnología Propia		X	
6	Soporte técnico ofrecido antes de la venta y después de la venta	X		
7	Procesos estandarizados y normados		X	
8	Se cuenta con un grupo de criterios y estándares nacionales para establecer calidad , competitividad e impacto social		X	

Tabla 3.2 Matriz de Resumen de Fortalezas

MATRICES DE RESUMEN				
#	MATRICES RESUMEN E IMPACTO DE ASPECTOS ESTRATEGICOS JERARQUIZADOS	IMPACTO		
	OPORTUNIDADES	ALTO	MEDIO	BAJO
	1	Tendencia por parte de las empresas al uso de IT (Tecnologías de la información)	X	
2	Abrirse puertas a empresas del sector público con un producto innovador.	X		
3	Necesidad de los empleadores y empleados de facilitar el trabajo de históricos de documentos	X		
4	La mayoría de las empresas públicas cuentan con infraestructura obsoleta y anticuada	X		
5	Poseer gran variedad de sistemas que puedan satisfacer las necesidades del cliente		X	
6	La baja creación de competencia por las barreras de entrada	X		

Tabla 3.3 Matriz de Resumen de Oportunidades

MATRICES DE RESUMEN				
#	MATRICES RESUMEN E IMPACTO DE ASPECTOS ESTRATEGICOS JERARQUIZADOS	IMPACTO		
		DEBILIDADES	ALTO	MEDIO
1	Estancarse con un solo producto y no ofrecer diversidad a los clientes.	X		
2	Falta un sistema que administre y gestión gastos		X	
3	Falta de alianzas estratégicas y convenio con los competidores que beneficien	X		
4	La infraestructura posiblemente colapse con el aumento laboral, no sea la adecuada			X
5	No posee un área bien estructura de marketing	X		
6	No posee una cadena de valor			X
7	No tiene personal preparado para realizar un Sistema de Digitalización	X		

Tabla 3.4 Matriz de Resumen de Debilidades

MATRICES DE RESUMEN				
#	MATRICES RESUMEN E IMPACTO DE ASPECTOS ESTRATEGICOS JERARQUIZADOS	IMPACTO		
		AMENAZAS	ALTO	MEDIO
1	Dependencias tecnológicas por parte de potencias mundiales	X		
2	Presencia de servicios y productos sustitutos	X		
3	Existen empresas de competencia en desarrollo de software.	X		
4	Ingreso de competidores con estructuras de costos menores	X		
5	Contratar asesoría para un tipo de proyecto específico		X	
6	Presupuesto mínimo por parte de las empresas para la inversión en el área IT		X	

Tabla 3.5 Matriz de Resumen de Amenazas

3.3.3 Cruces FODA

<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border: 1px solid black; padding: 5px; width: 150px;"> PONDERACION ALTA=5 MEDIA=3 BAJA=1 </div> <div style="writing-mode: vertical-rl; transform: rotate(180deg); font-weight: bold; font-size: 1.2em;">OPORTUNIDADES</div> </div> <div style="border: 1px solid black; padding: 5px; width: 150px;"> PMf = 25 PMc = 46,43 </div> <div style="font-weight: bold; font-size: 1.2em;">FORTALEZAS</div>	Tendencia por parte de las empresas al uso de IT (Tecnologías de la información)	Abrirse puertas a empresas del sector público con un producto innovador	Necesidad de los empleadores y empleados de facilitar el trabajo de históricos de documentos	La mayoría de las empresas públicas cuentan con infraestructura obsoleta y anticuada	Poseer gran variedad de sistemas que puedan satisfacer las necesidades del cliente	La baja creación de competencia por las barreras de entrada	TOTAL
	Infraestructura adecuada y equipamiento y tecnología de última generación.	5	5	3	5	5	3
La empresa posee un gran prestigio a través de servicio entregado a clientes	5	5	1	5	5	1	22
Excelencia y compromiso con los clientes	5	5	3	5	5	1	24
Nivel adecuado de recursos Financieros	5	5	1	3	5	1	20
Tecnología Propia	5	5	1	1	5	5	22
Soporte técnico ofrecido antes de la venta y después de la venta	5	5	3	5	5	1	24
Procesos estandarizados y normados	5	5	3	5	5	5	28
Se cuenta con un grupo de criterios y estándares nacionales para establecer calidad , competitividad e impacto social	5	5	3	3	5	1	22
TOTAL	40	40	18	32	40	18	188

Tabla 3.6 Matriz de Resumen de Fortalezas contra Oportunidades

<div style="border: 1px solid black; padding: 5px; width: fit-content;"> PONDERACION ALTA=5 MEDIA=3 BAJA=1 </div> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> PMf = 30,33 PMc = 22,75 </div> DEBILIDADES	AMENAZAS						TOTAL
	Dependencias tecnológicas por parte de potencias mundiales	Presencia de servicios y productos sustitutos	Existe empresas de competencia en desarrollo de software	Ingreso de competidores con estructuras de costos menores	Contratar asesoría para un tipo de proyecto específico	Presupuesto mínimo por parte de las empresas para la inversión en el área IT	
Estancarse con un solo producto y no ofrecer diversidad a los clientes	5	5	1	1	3	1	16
Falta un sistema que administre y gestión gastos	5	5	1	5	5	5	26
Falta de alianzas estratégicas y convenio con los competidores que beneficien	5	5	3	3	5	1	22
La infraestructura posiblemente colapse con el aumento laboral, no sea la adecuada	5	5	1	5	5	5	26
No posee un área bien estructura de marketing	5	5	3	3	3	1	20
No posee una cadena de valor	5	5	1	1	1	1	14
No tiene personal preparado para realizar un Sistema de Digitalización	5	5	1	1	1	1	14
TOTAL	35	35	15	19	23	15	138

Tabla 3.7 Matriz de Resumen de Debilidades contra Amenazas

<div style="border: 1px solid black; padding: 5px; width: fit-content;"> PONDERACION ALTA=5 MEDIA=3 BAJA=1 </div> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> PMf = 30,77 PMc = 50 </div> FORTALEZAS	AMENAZAS						TOTAL
	Dependencias tecnológicas por parte de potencias mundiales	Presencia de servicios y productos sustitutos	Existe empresas de competencia en desarrollo de software	Ingreso de competidores con estructuras de costos menores	Contratar asesoría para un tipo de proyecto específico	Presupuesto mínimo por parte de las empresas para la inversión en el área IT	

Infraestructura adecuada y equipamiento y tecnología de última generación.	5	5	5	5	3	5	28
La empresa posee un gran prestigio a través de servicio entregado a clientes	5	5	3	5	1	5	24
Excelencia y compromiso con los clientes	5	5	3	3	3	5	24
Nivel adecuado de recursos Financieros	5	5	5	5	5	5	30
Tecnología Propia	5	5	5	5	5	1	26
Soporte técnico ofrecido antes de la venta y después de la venta	5	5	3	3	5	5	26
Procesos estandarizados y normados	5	5	5	3	5	5	28
Se cuenta con un grupo de criterios y estándares nacionales para establecer calidad , competitividad e impacto social	5	5	1	3	5	5	24
TOTAL	40	40	30	32	32	36	274

Tabla 3.8 Matriz de Resumen de Fortalezas contra Amenazas

<div style="border: 1px solid black; padding: 5px; width: fit-content;"> PONDERACION ALTA=5 MEDIA=3 BAJA=1 </div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-top: 10px;"> PMf = 26,67 PMc = 22,86 </div> <p style="text-align: center; color: blue; font-weight: bold;">DEBILIDADES</p> <p style="text-align: center; color: blue; font-weight: bold;">OPORTUNIDADES</p>	Tendencia por parte de las empresas al uso de IT (Tecnologías de la información)	Abrirse puertas a empresas del sector público con un producto innovador	Necesidad de los empleadores y empleados de facilitar el trabajo de históricos de documentos	La mayoría de las empresas públicas cuentan con infraestructura obsoleta y anticuada	Poseer gran variedad de sistemas que puedan satisfacer las necesidades del cliente	La baja creación de competencia por las barreras de entrada	TOTAL	
	Estancarse con un solo producto y no ofrecer diversidad a los clientes	5	5	3	5	3	1	22
	Falta un sistema que administre y gestión gastos	5	5	5	5	5	1	26
	Falta de alianzas estratégicas y convenio con los competidores que benefician	5	5	3	5	5	5	28

La infraestructura posiblemente con el aumento laboral de proyectos nuevos para la empresa no sea la adecuada	5	5	5	5	5	1	26
No posee un área bien estructura de marketing	5	5	5	5	5	1	26
No posee una cadena de valor	5	5	1	5	5	1	22
No tiene personal preparado para realizar un Sistema de Digitalización	5	5	1	5	5	1	22
TOTAL	35	35	23	35	33	11	172

Tabla 3.9 Matriz de Resumen de Debilidades contra Amenazas

Una vez realizada estas matrices se plantean las siguientes estrategias y sobre todo se desea lanzar un producto estrella que permita cumplir los mismos.

3.3.4 Estrategias

3.3.4.1 Estrategias competitivas Actuales

- **Trabajar con las mejores tecnologías de última generación**

Para la implementación del sistema de digitalización masiva (SADO), la empresa “Soluciones Inteligentes” ha manejado tecnología de última generación lo cual ha permitido cubrir las necesidades tecnológicas que el mercado demanda.

- **Ofrecer ofertas y/o servicios agregados que permita a Soluciones Inteligentes captar la mayor cantidad de clientes que puedan confiar su información.**

Los objetivos claves de la empresa "Soluciones Inteligentes" están enfocadas en la penetración a nuevos mercados cuya necesidad sea la digitalización de documentos de manera masiva, dando lugar, que su principal nicho de mercado sea las empresas publicas y empresa que manejen volumen de documentación física.

- **Establecer un vínculo de confianza entre el cliente y la empresa que permita un acercamiento cada vez mayor a más clientes.**

Las capacidades claves de la empresa "Soluciones Inteligentes" son las relacionadas a su estrategia de enfoque en los segmentos de mercado de menor a mayor poder adquisitivo a través de sistemas, que cumplan las expectativas del cliente de bajo precio y buena calidad. Se han distinguido dos capacidades claves, la Penetración a nuevos mercados como son las empresas públicas y el acercamiento de su producto en la de digitalización de documentos masivos.

- **Mayor difusión y masificación de nuestros servicios**

Soluciones Inteligentes restaurará el área de marketing que impulsará la venta de diversos sistemas estrellas desarrollados por la empresa y a su vez promoverá a la mismas, cumpliendo así el objetivo de captar más clientes.

3.3.4.2 Estrategias Funcionales

3.3.4.2.1 Estrategias Concretas

El denominador común de la estrategia propuesta por la empresa Soluciones Inteligentes es el enfoque en los segmentos de mercado con menor o mayor poder adquisitivo.

En estos segmentos la elección fundamental es satisfacer las necesidades del cliente por lo que es primordial que se conserve la imagen de la empresa, de sus productos y la estructura empresarial que hace posible ofrecer precios justos y sistemas de calidad.

De manera que el componente de la estrategia es la reducción del riesgo de dependencia de un tipo de producto, lo cual se busca diferenciar de la competencia con un sistema de digitalización masiva y así lograr la diversificación en productos la empresa y acaparar segmentos mercados no explotados y gozar de fuerte posicionamiento.

3.4 Formulación y Análisis

Una vez realizadas las observaciones de campo y las entrevistas, se ha logrado identificar las necesidades de los usuarios, lo cual permite describir las siguientes características del sistema:

- Eficaz, comprendida por todos
- Confiable, satisface las necesidades de los clientes.
- Segura, por la garantía de los documentos.

Estas características son tomadas como requerimientos no funcionales.

Por lo tanto, poseerá una calidad tanto en la elaboración y como en su utilización.

3.5 Especificación de Requerimientos

La presente especificación de requerimientos pertenece al desarrollo del Caso de Estudio " Sistema Automatizado de Digitalización de Documentos (SADO) " como Tesis para la obtención del título de Ingeniería en Sistemas e Informática y está desarrollada siguiendo las directrices de la metodología OOHDM y del Lenguaje Unificado de Modelado (UML).

Soluciones Inteligentes entregó los requerimientos del sistema en base a su identificación de las necesidades para el desarrollo de este proyecto.

Se procedió con la validación de requerimientos entregados por soluciones inteligentes, dichos requerimientos satisfacían en su mayoría el desarrollo del software, adicionalmente se agregaron otros requerimientos funcionales que fueron encontrados durante el análisis de la aplicación.

3.5.1 Introducción

3.5.1.1 Propósito

El propósito del presente apartado es definir los requerimientos que debe tener el Sistema Automatizado de Digitalización de Documentos (SADO) a partir de ahora.

Con la especificación de requerimientos se formalizará las funcionalidades de la aplicación junto al cliente.

3.5.2 Identificación de Roles y Tareas

Roles

➤ **Administrador**

Es el usuario que tiene todos los permisos y privilegios para gestionar cualquier área del sistema y posee la capacidad de auditar los movimientos del sistema.

➤ **Editor**

Es el usuario con ciertos privilegios que dependerán de las necesidades y funciones del personal para el manejo de la administración de documentos, consulta de históricos y reportes.

➤ Usuario

Es el usuario que únicamente podrá consultar la existencia de algún documento, por medio de la opción de históricos o consulta específica de los documentos puede solicitar impresiones de los mismos o actualizaciones de un documento, pero no tiene acceso a los demás servicios del mismo, como se indica en la Figura 3.1.

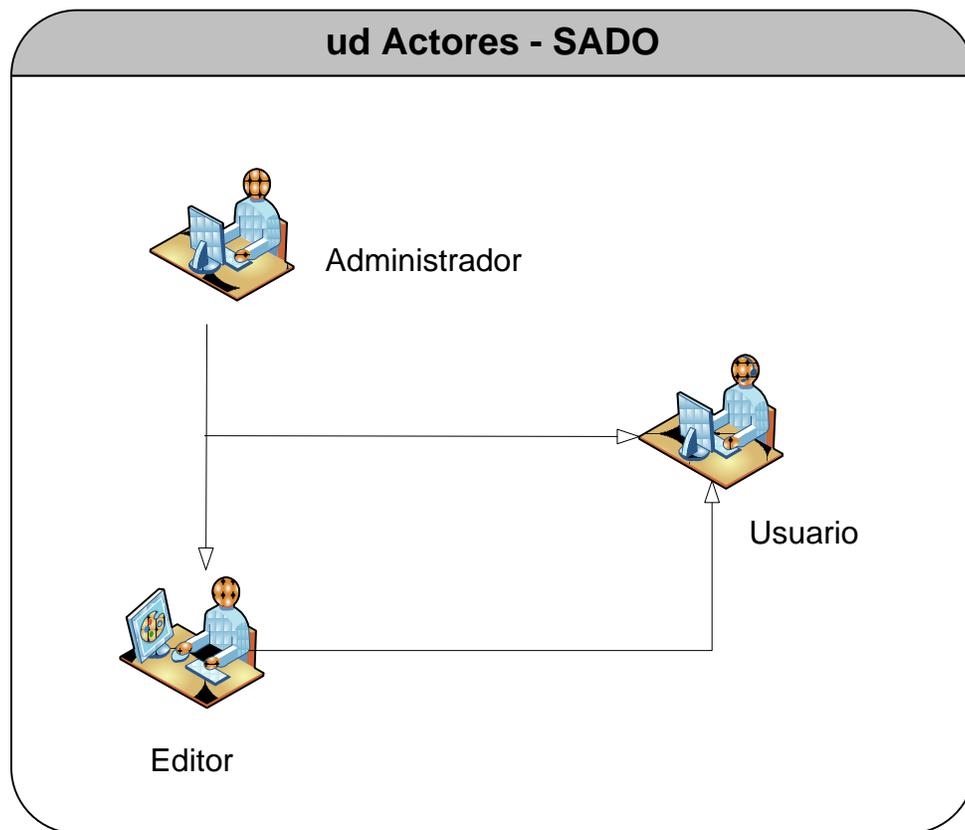


Figura 3.1 Actores - SADO

Tareas

➤ Usuario

1. Utiliza el Servicio de Históricos
2. Utiliza Servicio de Consulta de Documentos
3. Solicita Prestamos e Impresiones

➤ Editor

1. Utilizar Servicio de Administración de Documentos
 - Aprobación de Documentos Nuevos
 - Actualizaciones
 - Aprobación de Impresiones y Prestamos
2. Utilizar Servicio de Históricos
3. Utilizar Servicio de Consulta
4. Utilizar Servicio de Reportes

➤ Administrador

1. Gestionar Usuarios
2. Gestionar Seguridades
3. Gestionar Reportes
4. Gestionar Servicio de Administración de Documentos
 - a. Aprobación de Documentos Nuevos
 - b. Actualizaciones
 - c. Aprobación de Impresiones y Prestamos

3.5.3 Especificación de Escenarios

Rol Administrador

- **Gestionar Usuarios:** Se podrá importar y actualizar los usuarios y determinar privilegios o permisos para la administración y navegación al usar los servicios del sistema.

Importante: Este proceso se lo debe revisar mensualmente o cuando existan cambios dentro de la organización que ameriten la revisión de los usuarios asignados para el uso del Software.

- **Gestionar Seguridades:** Asignación de Grupos y Usuarios sincronizados con LDAP a los niveles de acceso al sistema
- **Gestionar Aprobación de Documentos:** Se organizará de manera eficaz los perfiles por cada modulo, creación, edición de los módulos existentes y categorías para cada tipo de documento y sobre toda especificación de los directorios donde se localizarán los documentos digitalizados.
- **Gestionar Histórico de Documentos:** Se podrá realizar reasignaciones para la aprobación de documentos, eliminaciones (Lógicas) o desactivación de su visibilidad, aprobaciones de Documentos que se presentará a cargo de una persona responsable de la categoría a la que pertenece el documento y además se permite tener un control de versionamiento de cada documento.
- **Gestionar Reportes:** Se ofrecerá una serie de reportes donde abarcará la información básica de los documentos más solicitados, editados o visitados

para así ofrecer a la empresa una información más veraz sobre el manejo de cada archivo que ha sido digitalizado.

Rol Editor

- **Utilizar Servicio de Administración de Documentos:** El Editor podrá revisar la categoría de documentos asignada correspondiente a su Rol a su vez podrá cambiar de directorio al documento o reasignar a otra categoría.
- **Utilizar Servicio de Históricos:** El Editor podrá realizar el versionamiento de un documento, enviar al administración la aprobación de un documento que haya sufrido cambios o solicitar la eliminación de el mismo.
- **Utilizar Servicio Consultas:** El Editor podrá realizar una búsqueda avanzada y visualización de dicho documento, y también y realizar las operaciones que su rol le permite realizar.
- **Utilizar Servicio Reportes:** El usuario podrá visualizar los reportes que se encuentran en el sistema y así poseer un conocimiento general del acceso y manejo de los documentos digitalizados.

Rol Usuario

- **Utiliza Servicio de Históricos:** El Usuario podrá visualizar los documentos que se encuentran como históricos, y a su vez solicitar al administrador la entrega o acceso al documento de su interés.

- **Utilizar Servicio Consulta de Documentos:** El usuario podrá realizar una búsqueda avanzada de cierto documento y realizar las operaciones que su rol le permite realizar.

- **Utilizar Servicio de Notificaciones:** El usuario creará solicitudes sobre diversos temas como prestamos, impresiones de documentos.

3.5.4 Especificación de casos de uso por Actor

ACTOR: ADMINISTRADOR

1. CU-ADM-SADO-01: Gestionar Usuarios
2. CU-ADM- SADO -02: Gestionar Seguridades
3. CU-ADM- SADO -03: Gestionar Administración de Documentos
4. CU-ADM- SADO -04: Gestionar Histórico de Documentos
5. CU-ADM- SADO -05: Gestionar Reportes
6. CU-ADM- SADO -06: Gestionar Servicio Notificaciones

A continuación El Diagrama de Caso de Uso en La Figura 3.2.

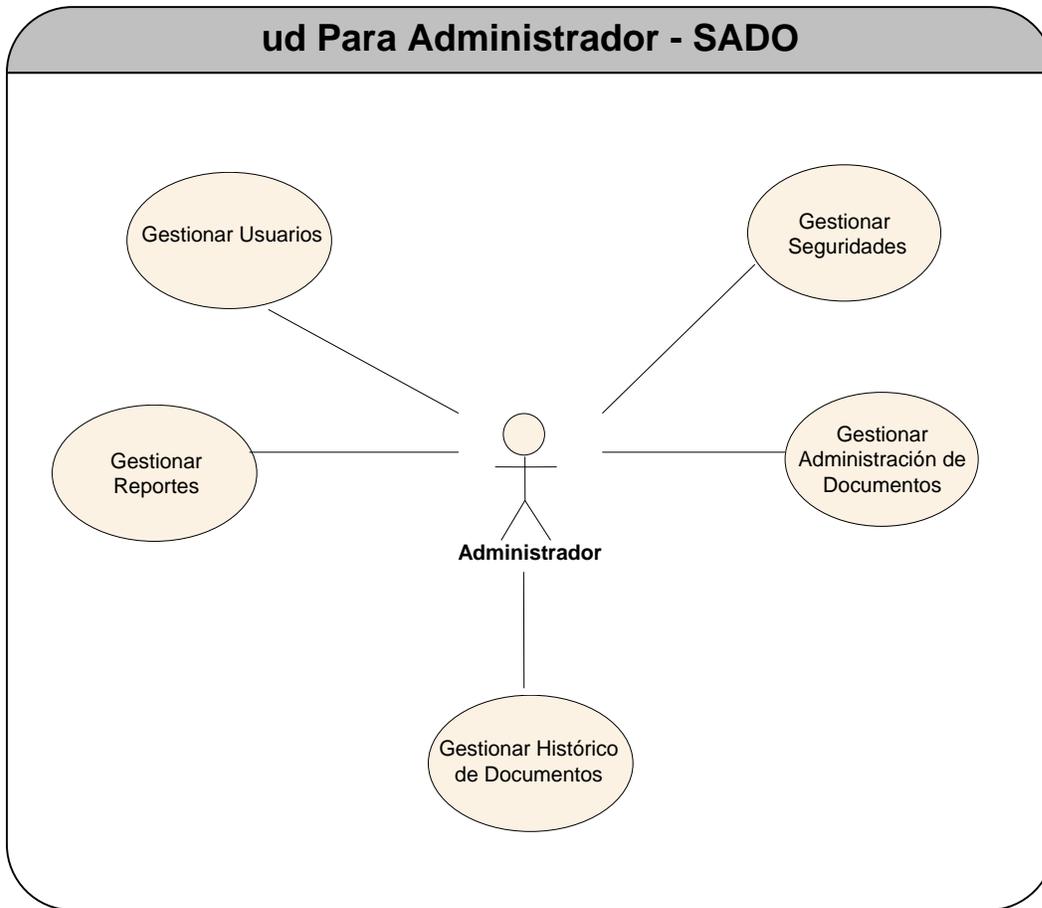


Figura 3.2 Casos de Uso para Administrador SADO

CASOS DE USO DEL ADMINISTRADOR SADO

CU-ADM-SADO-01: Gestionar Usuarios	
Resumen:	Proceso en el cual se administran los usuarios que interactúan con el sistema.
Prioridad:	Esencial
Actores Directos:	Administrador
Escenarios	
Tipo de Escenario	Descripción
Principal	1. El sistema lista los usuarios existentes en la base de datos, dando la opción de organizarlos por grupos: <ul style="list-style-type: none"> - Public Front-end (usuarios que acceden la aplicación) - Public Back-end (usuarios que administran la aplicación)

	<p>El aplicativo SADO se ha dividido a los usuarios en:</p> <ul style="list-style-type: none"> - Super Administrator (Public Back-end) → Rol Administrador - Registrado ó Autor (Public Front-end) → Rol Editor - Usuario (Public Front-end) → Rol Usuario <p>2. El Administrador escoge la acción que desea realizar: Crear, Modificar o Eliminar un usuario.</p>	
Secundario	Crear	<ol style="list-style-type: none"> 1. El sistema muestra los controles necesarios para el ingreso del nuevo registro, adicionalmente muestra controles para asignar el rol que tendrá el nuevo usuario. 2. El usuario ingresa los datos del nuevo registro. 3. El sistema valida los datos ingresados. 4. El sistema presenta un mensaje confirmando el ingreso de datos exitoso.
Secundario	Modificar	<ol style="list-style-type: none"> 1. El sistema muestra los controles necesarios para la modificación del registro. 2. El usuario ingresa o cambia los datos del registro. 3. El sistema valida los datos ingresados. 4. El sistema presenta un mensaje confirmando la actualización de datos exitosa.
Secundario	Eliminar	<ol style="list-style-type: none"> 1. El sistema muestra los controles necesarios para la eliminación del registro seleccionado. 2. El usuario elimina el registro indicado. 3. El sistema presenta un mensaje confirmando la eliminación de datos exitoso.
Pre-condiciones		
<ol style="list-style-type: none"> 1. El usuario debe ingresar al sistema autenticado como Administrador. <p>Notas:</p> <ul style="list-style-type: none"> - Por motivos de seguridad, todos los miembros de la aplicación SADO que requieran tener acceso a los servicios, deberán enviar sus datos al Administrador por e-mail, carta u oficio para ser habilitados. - El Rol Miembro podrá desempeñar 2 tipos de acciones diferentes, dependiendo las funciones del personal. <ul style="list-style-type: none"> - Como Editor podrá digitalizar documentos y pedir autorizaciones y tener acceso a los servicios - Como Usuario podrá visualizar los documentos que se encuentren como públicos. 		

Tabla 3.10 Gestionar Usuarios

CU-ADM-SADO-02: Gestionar Seguridades		
Resumen:	Proceso en el cual se gestiona la validación vía LDAP.	
Prioridad:	Esencial	
Actores Directos:	Administrador	
Escenarios		
Tipo de Escenario	Descripción	
Principal	1.Importar los usuarios y sus perfiles desde el active directory 2.Validar el perfil de usuario y acceso que posee el mismo dentro del sistema	
Secundario	Configurar	1. Se configura el acceso al LDAP. 2. Se importa los usuarios y permisos.
Pre-condiciones		
1. El usuario debe tener instalado la aplicación de administración de LDAP.		

Tabla 3.11 Gestionar Seguridades

CU-ADM-SADO-03: Gestionar Administración de Documentos		
Resumen:	Proceso en el cual se gestiona la administración de documentos que interactúa con cada uno de los roles de cada usuario.	
Prioridad:	Esencial	
Actores Directos:	Administrador	
Escenarios		
Tipo de Escenario	Descripción	
Principal	1. El sistema permite la creación de categorías para los documentos 2. El sistema permite controlar versiones del documento 3. El Administrador podrá aprobar el acceso e impresión de un documento. 4. El Administrador podrá: crear, publicar y eliminar un documento.	
Secundario	Crear Categorías	1. El sistema muestra los controles necesarios para administración de categorías

		<ol style="list-style-type: none"> 2. El Administrador crea una nueva categoría para el acceso del documento dependiendo de los perfiles. 3. El sistema presenta un mensaje confirmando el ingreso de datos exitoso.
Secundario	Crear Directorio	<ol style="list-style-type: none"> 1. El sistema muestra los controles necesarios para el ingreso del nuevo directorio. 2. El Administrador crea un nuevo directorio en la categoría seleccionada. 3. El sistema valida los datos ingresados. 4. El sistema presenta un mensaje confirmando el ingreso de datos exitoso.
Secundario	Modificar Directorio	<ol style="list-style-type: none"> 1. El sistema muestra los controles necesarios para la modificación del directorio 2. El Administrador ingresa o cambia los datos del directorio. 3. El sistema valida los datos ingresados. 4. El sistema presenta un mensaje confirmando la actualización de datos exitosa.
Secundario	Eliminar directorio	<ol style="list-style-type: none"> 1. El sistema muestra los controles necesarios para la eliminación del directorio seleccionado. 2. El Administrador elimina el registro indicado. 3. El sistema presenta un mensaje confirmando la eliminación de datos exitoso.
Secundario	Aprobación de documentos	<ol style="list-style-type: none"> 1. El sistema muestra los controles necesarios para la aprobación o eliminación de documentos. 2. El Administrador realiza las acciones pertinentes (Aprobar la visualización e impresión de un documento) 3. El sistema presenta un mensaje confirmando las acciones realizadas.
Pre-condiciones		
1. El usuario debe ingresar al sistema autenticado como Administrador.		

Tabla 3.12 Gestionar Administración de Documentos

CU-ADM-SADO-04: Gestionar Histórico de Documentos		
Resumen:	Proceso por el cual se maneja un histórico de documentos	
Prioridad:	Esencial	
Actores Directos:	Administrador	
Escenarios		
Tipo de Escenario	Descripción	
Principal	1. El sistema muestra la opción de Históricos de Documentos 2. El Administrador escoge la reasignación , eliminación , actualización y versionamiento de los documentos 3. El sistema despliega un mensaje de confirmación o fallo de la instalación o desinstalación realizada.	
Secundario	Crear	1. El sistema muestra los controles necesarios para el ingreso de una versión de documentos 2. El sistema ingresa los datos del nueva versión de documento. 3. El sistema valida los datos ingresados. 4. El sistema presenta un mensaje confirmando el ingreso de datos exitoso.
Secundario	Modificar	1. El sistema muestra los controles necesarios para la modificación o actualización de la versión. 2. El usuario ingresa o cambia los datos del la versión del documento. 3. El sistema valida los datos ingresados. 4. El sistema presenta un mensaje confirmando la actualización de datos exitosa.
Secundario	Eliminar	1. El sistema muestra los controles necesarios para la eliminación del versión seleccionado. 2. El usuario elimina la versión de el documento indicado. 3. El sistema presenta un mensaje confirmando la eliminación de datos exitoso.
Pre-condiciones		
1. El usuario debe ingresar al sistema autenticado como Administrador.		

Tabla 3.13 Gestionar Histórico de Documentos

CU-ADM-SADO-05: Gestionar Reportes		
Resumen:	Proceso en el cual se gestiona el servicio de reportes generales que son requeridas por los gerentes y usuarios.	
Prioridad:	Esencial	
Actores Directos:	Administrador	
Escenarios		
Tipo de Escenario	Descripción	
Principal	1. El sistema muestra la opción de instalar y configurar el servicio de descarga de archivos. 2. El Administrador podrá realizar las siguientes acciones: Crear y Eliminar.	
Secundario	Crear	1. El sistema muestra los controles necesarios la creación de un reporte específico 2. El Administrador realiza la parametrización del reporte. 3. El Administrador proporciona la descripción del reporte y categoriza el acceso al mismo.
Secundario	Eliminar	1. El sistema muestra los controles necesarios para la eliminación de un reporte específico 2. El Administrador busca, selecciona y elimina el reporte. 3. El sistema presenta un mensaje de confirmación para que el usuario acepte eliminar el reporte
Pre-condiciones		
1. El usuario debe ingresar al sistema autenticado como Administrador.		

Tabla 3.14 Gestionar Reportes

ACTOR: EDITOR

1. CU-EDI-SADO -01: Utilizar Servicio de Históricos
2. CU-EDI-SADO -02: Utilizar Servicio de Consulta

A continuación El Diagrama de Caso de Uso en La Figura 3.3.

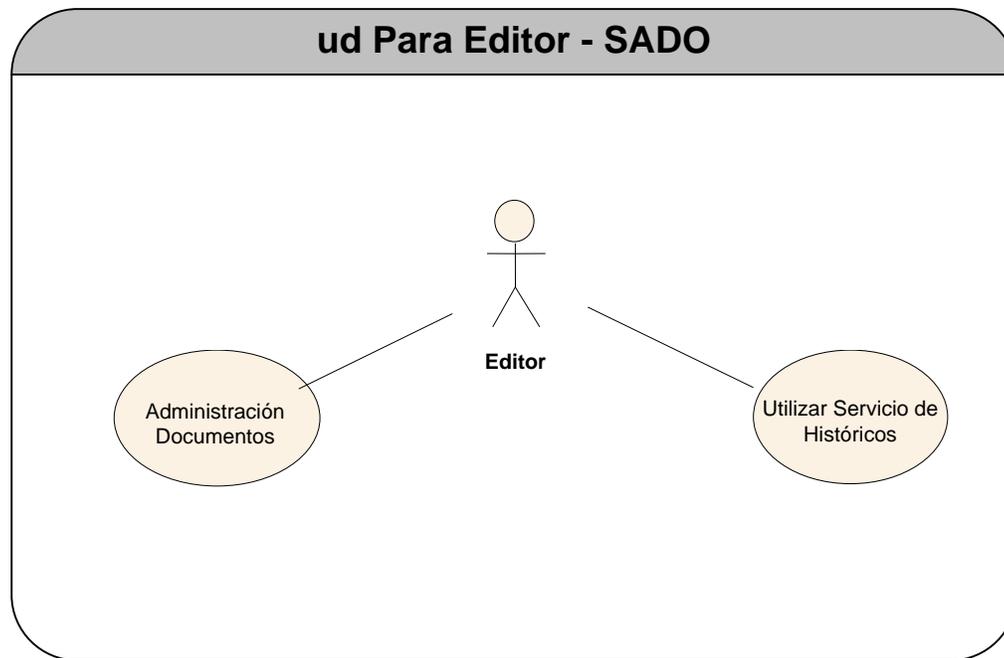


Figura 3.3 Casos de Uso para Editor SADO

CU-EDI-SADO-01: Utilizar Servicio de Históricos	
Resumen:	Proceso en el cual el usuario utiliza el servicio para obtener una versión de documento deseado
Prioridad:	Esencial
Actores Directos:	Miembro
Escenarios	
Tipo de Escenario	Descripción
Principal	1. El Sistema muestra la históricos de documentos 2. El editor escoge la categoría de documentos que desea observar estos se mostraran dependiendo de los privilegios establecidos si la categoría que se necesita observar no se encuentra disponible el usuario. 3. El editor puede realizar el servicio de petición de libros
Pre-condiciones	
1. El editor debe ingresar al sistema con autenticación.	

Tabla 3.15 Utilizar Servicio de Históricos

CU-EDI-SADO-02: Administración de Documentos	
Resumen:	Proceso en el cual el editor accede a realizar la aprobación documento deseado.
Prioridad:	Esencial
Actores Directos:	Editor
Escenarios	
Tipo de Escenario	Descripción
Principal	1. El sistema muestra la opción de administración de documentos. 2. El editor realiza aprobación de documento deseado dependiendo de la lista de documentos pendientes.
Pre-condiciones	
1. El usuario debe ingresar al sistema autenticado como Editor	

Tabla 3.16 Utilizar Servicio Administración de Documentos

ACTOR: USUARIO

1. CU-USU-SADO-01: Utiliza el Servicio de Históricos
2. CU-USU-SADO-02: Utiliza Servicio de Consulta de Documentos

A continuación El Diagrama de Caso de Uso en La Figura 3.4.

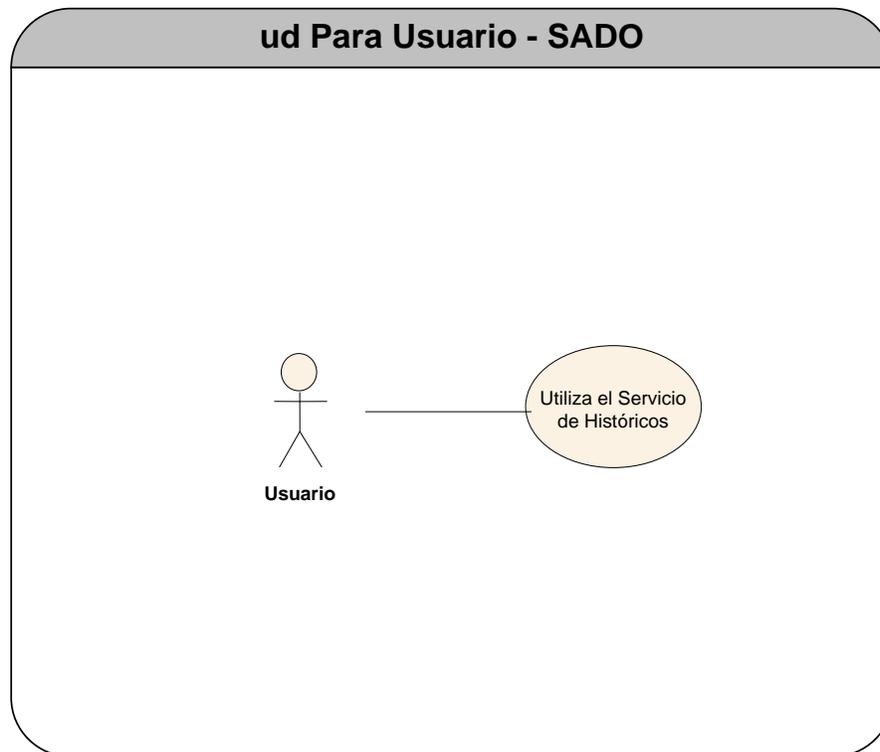


Figura 3.4 Casos de Uso para Usuario SADO

CU-USU-SADO-01: Utiliza el Servicio de Históricos		
Resumen:	Proceso en el cual el usuario tiene la posibilidad utilizar la opción de Históricos	
Prioridad:	Esencial	
Actores Directos:	Usuario	
Escenarios		
Tipo de Escenario	Descripción	
Principal	1. El usuario ingresa al sistema SADO. 2. El portal muestra su página de inicio con sus menús.	
Secundario	Ingreso	1. El sistema muestra un tipo de menú lateral. 2. El usuario puede ingresar a las opciones configuradas por el perfil que pertenecen.
Pre-condiciones		
1. El usuario debe ingresar al sistema sin ninguna autenticación.		

Tabla 3.17 Utiliza el Servicio de Históricos

3.6 Requerimientos No Funcionales

- El aplicativo debe poseer un esquema de perfiles de usuario con roles definidos, que garantice diferentes niveles de acceso a los servicios y a la información, de tal manera que un usuario no validado no tendrá acceso a todos los datos u operaciones disponibles.
- El aplicativo debe poseer un diseño de almacenamiento de datos que permita que el tiempo de generación de respuesta tanto para las consultas de información y paseos virtuales como para el uso de servicios sea optimizado en la mayor medida posible.
- Se necesita una unidad organizacional de archivo que maneje el documento actual este rol puede realizar el Administrador y Editor.

3.7 Diagramas de Secuencia

Se encuentran en el **Anexo A** del presente trabajo y representan el diseño de navegación del sistema SADO.

3.8 Diseño Conceptual

3.8.1 Diseño de Base de Datos

El siguiente diagrama de base de datos llamada SADOBD, posee relaciones lógicas en la Figura 3.5.

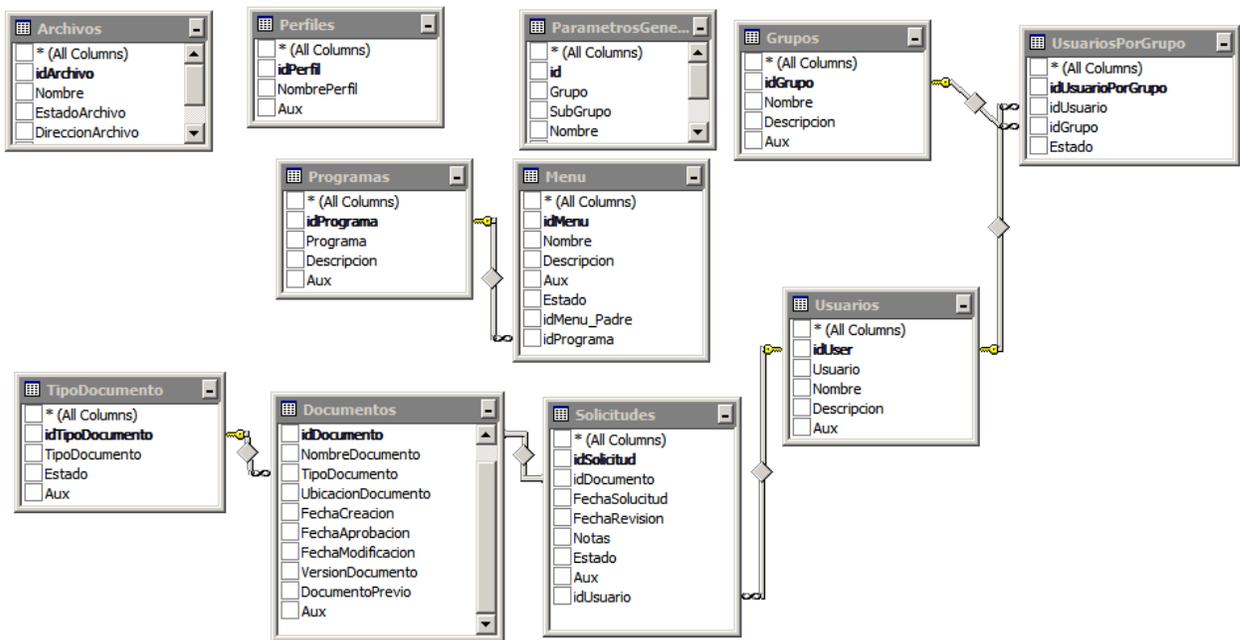


Figura 3.5 Diagrama de Base de Datos SADOBD

3.8.2 Diagramas de Clase

Archivo



Figura 3.6 Diseño Conceptual – Componente de Archivo

Perfiles

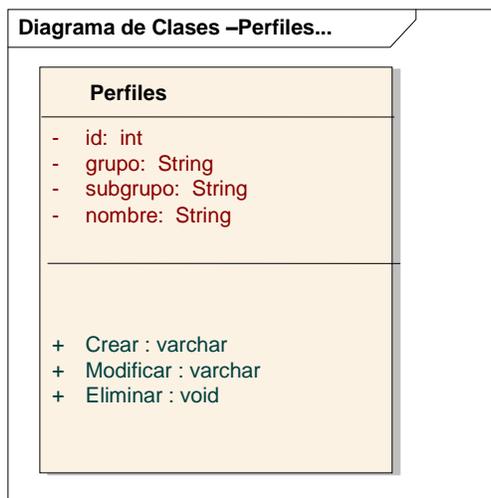


Figura 3.7 Diseño Conceptual – Componente de Perfiles

Parámetros Generales

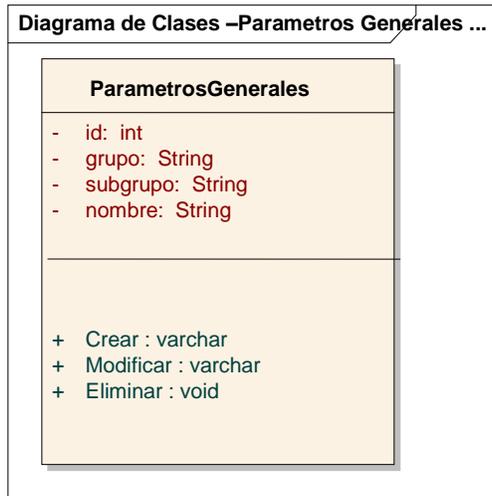


Figura 3.8 Diseño Conceptual – Componente

Grupos

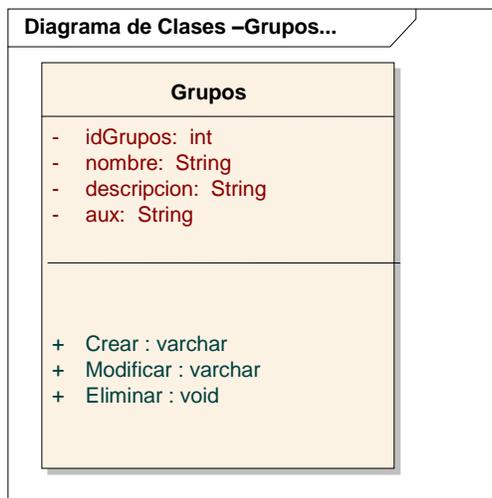


Figura 3.9 Diseño Conceptual – Grupos

Usuario Por Grupo

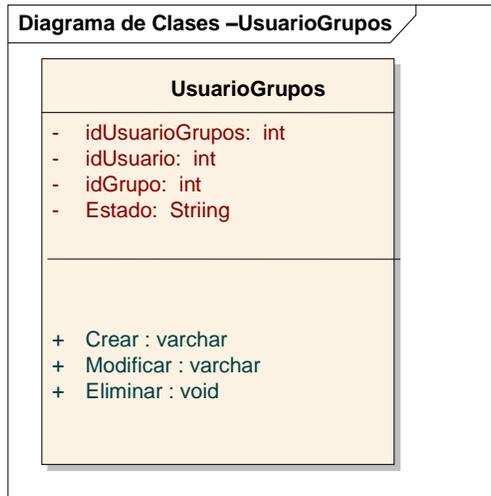


Figura 3.10 Diseño Conceptual – Usuario por Grupos

Programas

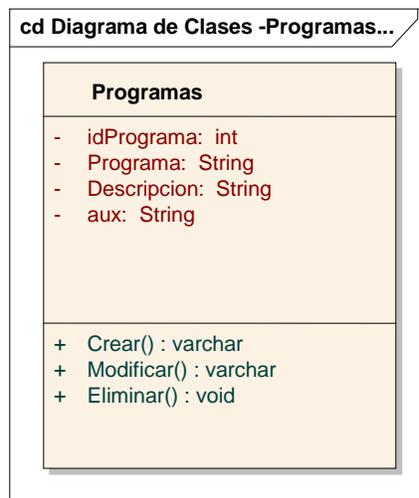


Figura 3.11 Diseño Conceptual – Programas

Menú

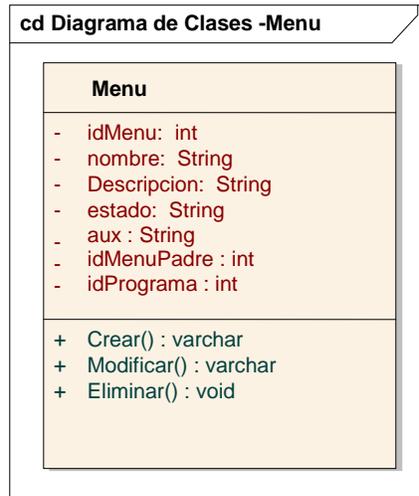


Figura 3.12 Diseño Conceptual – Menú

Tipo de Documento

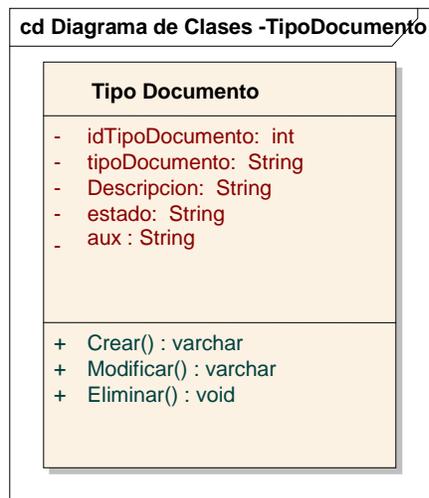


Figura 3.13 Diseño Conceptual – Tipo Documento

Documentos

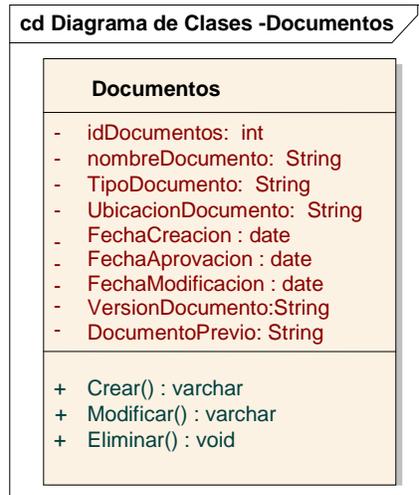


Figura 3.14 Diseño Conceptual –Documentos

Solicitudes

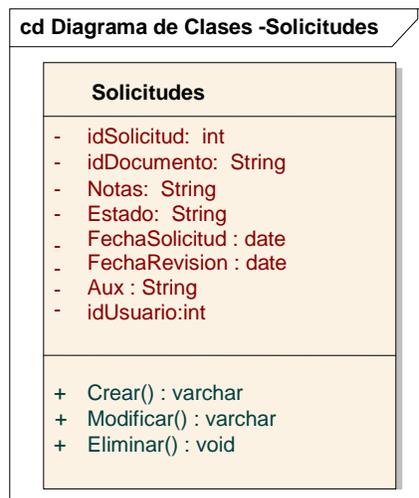


Figura 3.15 Diseño Conceptual – Solicitudes

PlugIn

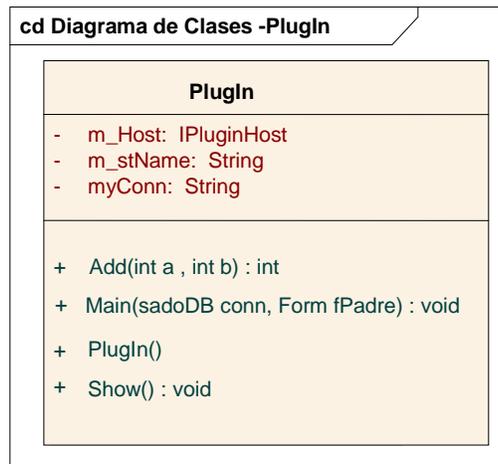


Figura 3.16 Diseño Conceptual – PlugIn

FrmMain

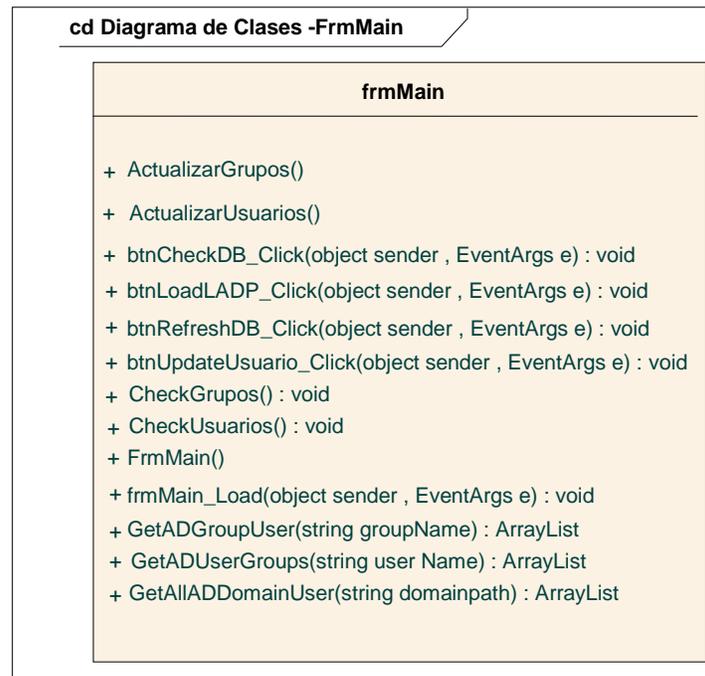


Figura 3.17 Diseño Conceptual –FrmMain

SadoDB

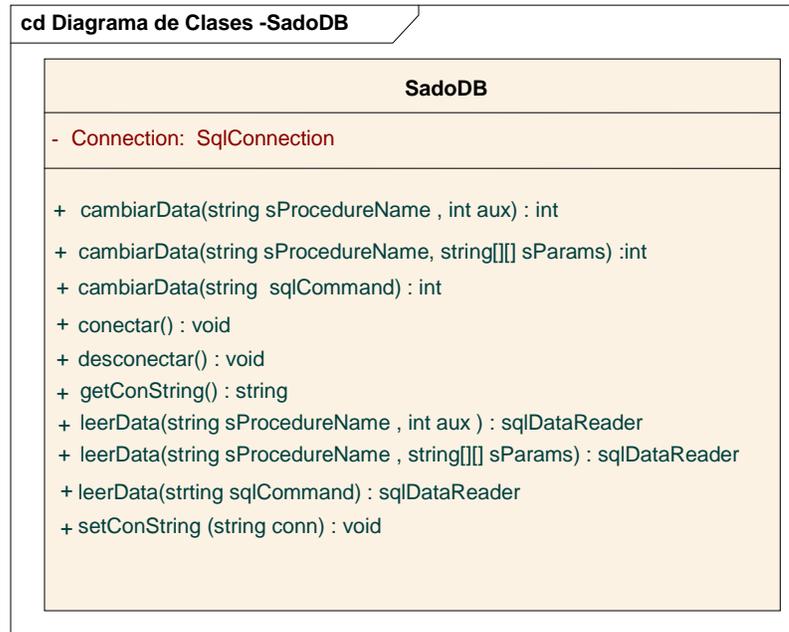


Figura 3.18 Diseño Conceptual – SadoDB

IPlugin

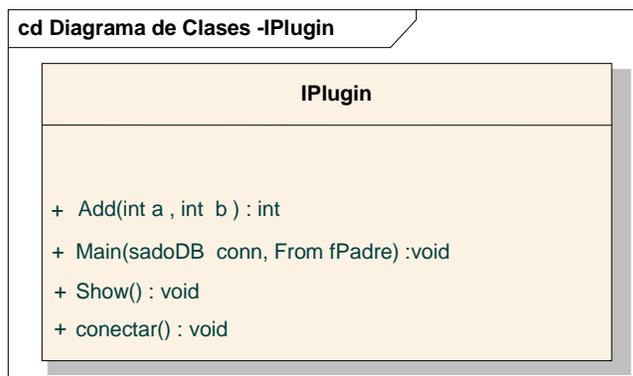


Figura 3.19 Diseño Conceptual – IPlugin

IPluginHost

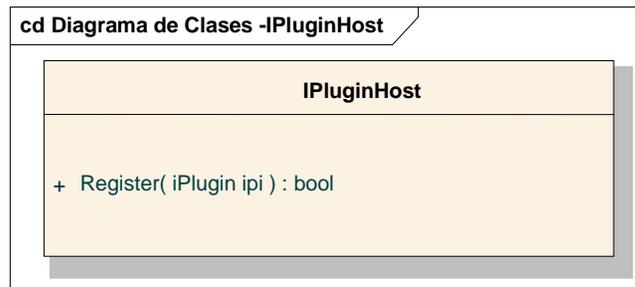


Figura 3.20 Diseño Conceptual – IPluginHost

Vigilante

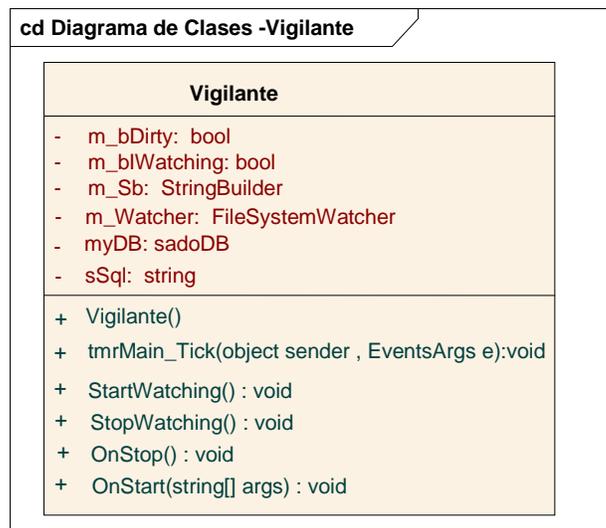


Figura 3.21 Diseño Conceptual – IPlugin

3.9 Diseño Navegacional

Está basado en IWeb, por tanto contiene una mezcla adecuada de estética, contenido y tecnología. Posee los siguientes objetos y contextos navegacionales:

3.9.1 Esquema de Contextos Navegacionales

Posee la siguiente estructura compuesta¹⁶ basada en el concepto de diseño arquitectónico de la IWeb:

¹⁶ **Estructura Compuesta:** La arquitectura global de la WebApp es jerárquica, pero también puede mostrar características lineales y otras en red

Menú Principal

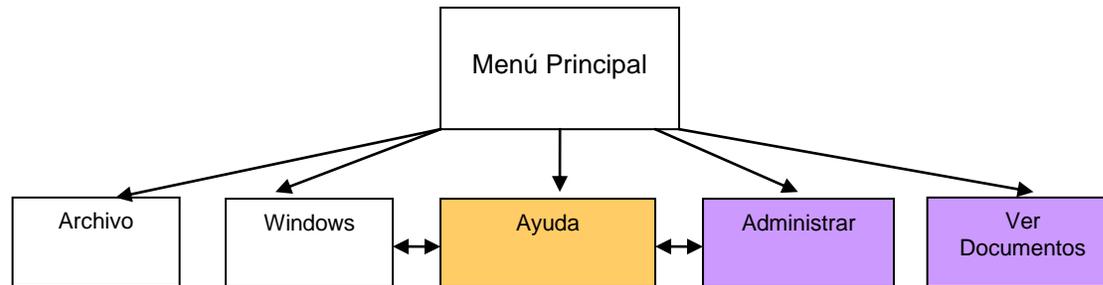


Figura 3.22 Esquema de Contexto - Menú Principal

División del menú principal:

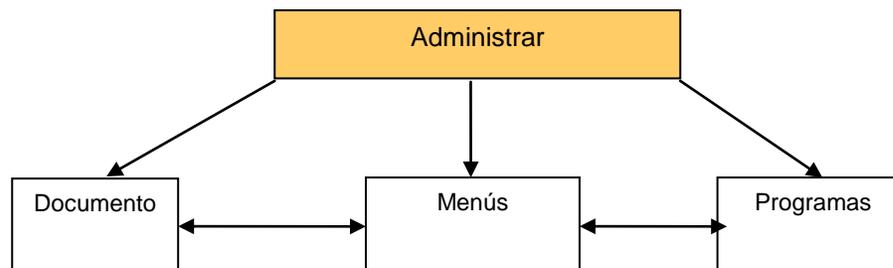


Figura 3.23 Esquema de Contexto – Menú Principal –Administración

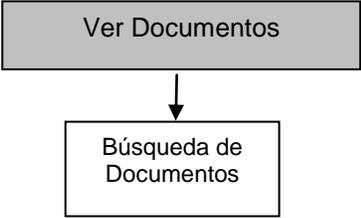


Figura 3.24 Esquema de Contexto – Menú Principal – Consultas

Menú Horizontal

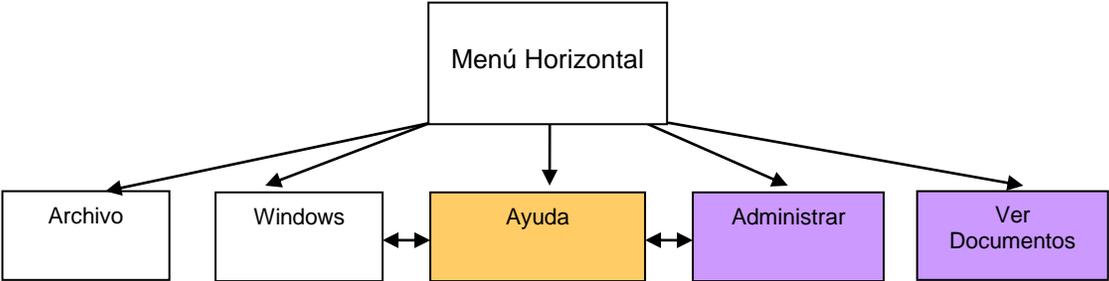


Figura 3.25 Esquema de Contexto – Menu Horizontal

Para el caso de usuarios registrados en el sistema existirá el siguiente menú de Usuario:

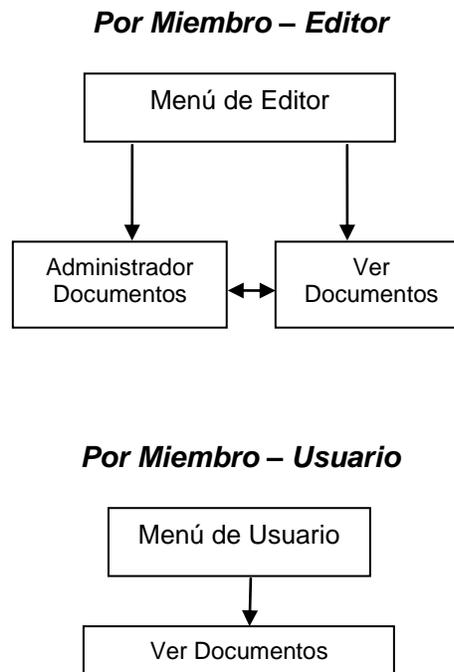


Figura 3.26 Esquema de Contexto – Enlaces Por Actor

3.10 Diseño de Interfaz Abstracta

Las interfaces están basadas tanto en una arquitectura de contenido (forma en la que los objetos se estructuran para su presentación y navegación) como en una arquitectura de WebApp (forma en la que la aplicación se estructura para gestionar la interacción del usuario).

Se detallan a continuación las interfaces del usuario, editor, administrador; ya que poseen interacción directa con el portal.

3.10.1 Vista de Datos Abstractos

➤ **Nodo Página**

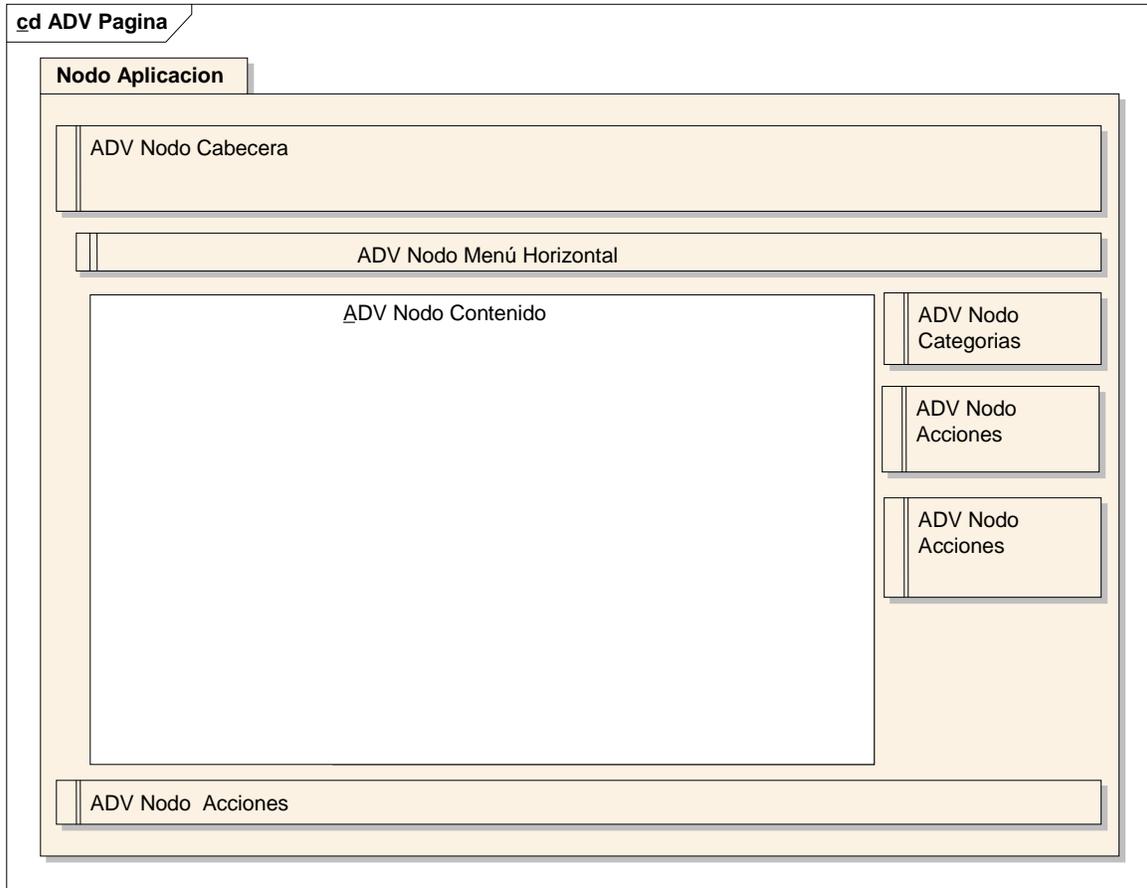


Figura 3.27 Vista Abstracta del Nodo Aplicaciones

➤ **Nodo Cabecera**

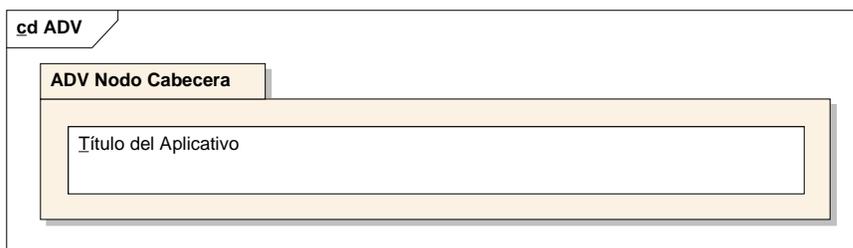


Figura 3.28 Vista Abstracta del Nodo Cabecera

➤ **Nodo Menú Horizontal para Administrador**



Figura 3.29 Vista Abstracta del Nodo Menú Horizontal Administrador

➤ **Nodo Menú del Editor**



Figura 3.30 Vista Abstracta del Nodo Menú del Editor

➤ **Nodo Menú del Usuario**

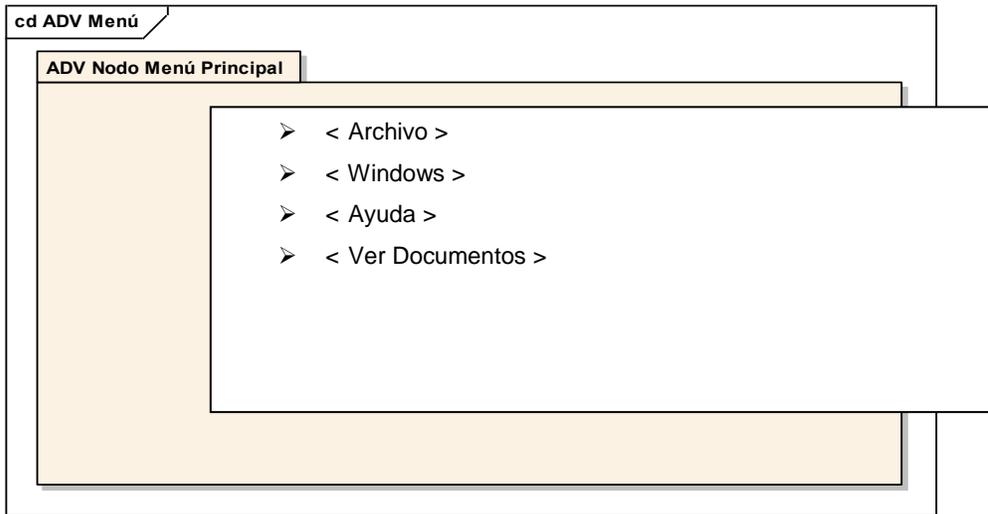


Figura 3.31 Vista Abstracta del Nodo Menú del Usuario

➤ **Nodo Contenido**

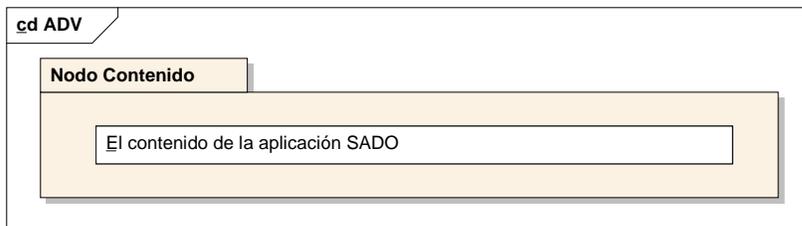


Figura 3.32 Vista Abstracta del Nodo Contenido

3.11 Diagramas de Estado

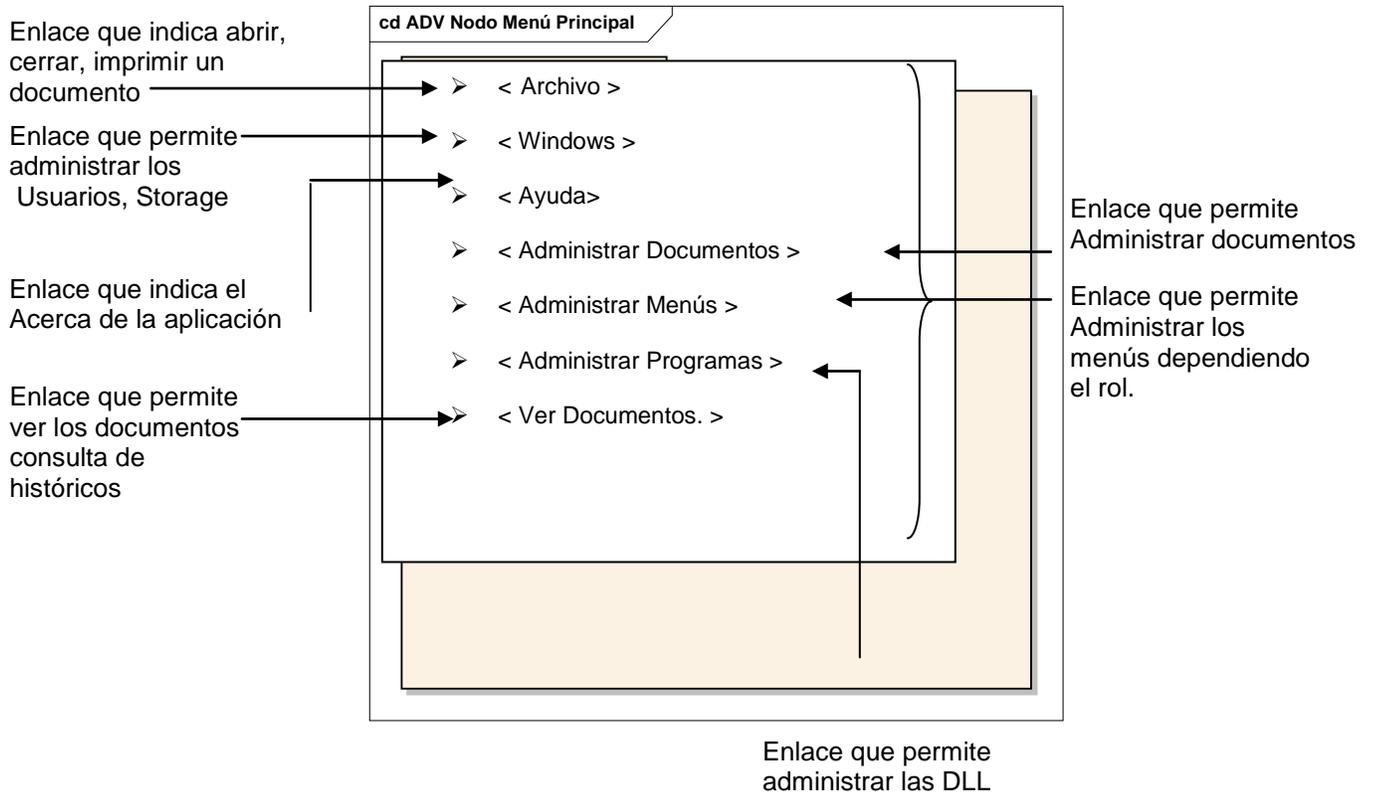


Figura 3.33 Diagrama de Estado – SADO

Todos los diagramas de Estado de los otros menús poseen la misma lógica que el Menú Principal y cada etiqueta de los enlaces posee un nombre relacionado con el detalle de la opción ó con el servicio a ser desplegado.

3.12 Diseño Estético

3.12.1 Características de la Plantilla

- Espacio en blanco: 10%
- Contenido: 90%
- Organización de los elementos:

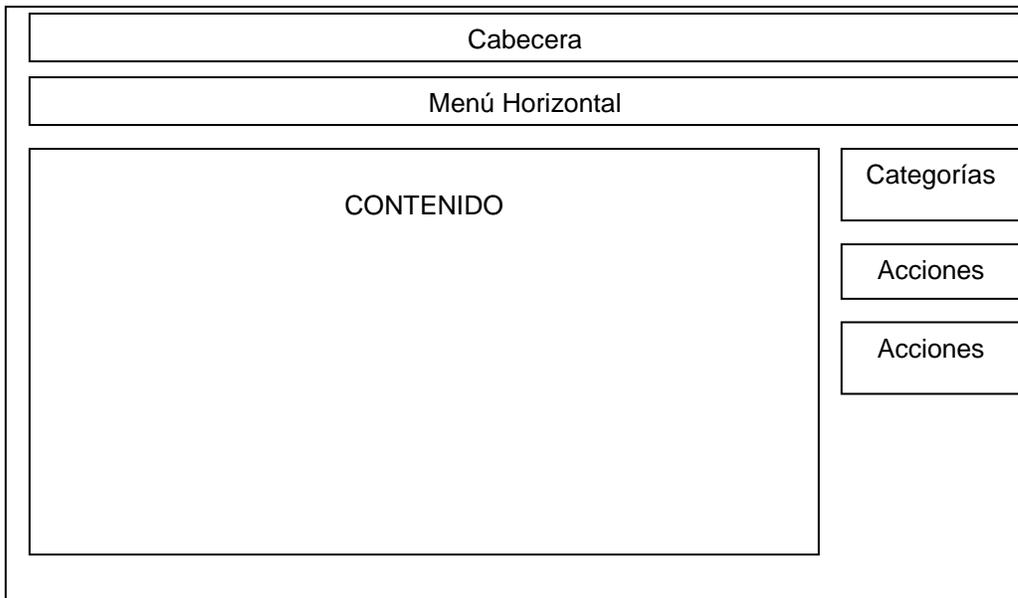


Figura 3.34 Diseño Estético – Organización de los Elementos

3.13 Consideraciones de Diseño Gráfico

Cabecera, Contenido

- Tipo de letra: Arial
- Tamaño promedio: 12px

Colores Globales del aplicativo: Colores básicos de Windows

3.14 Diagramas de Despliegue

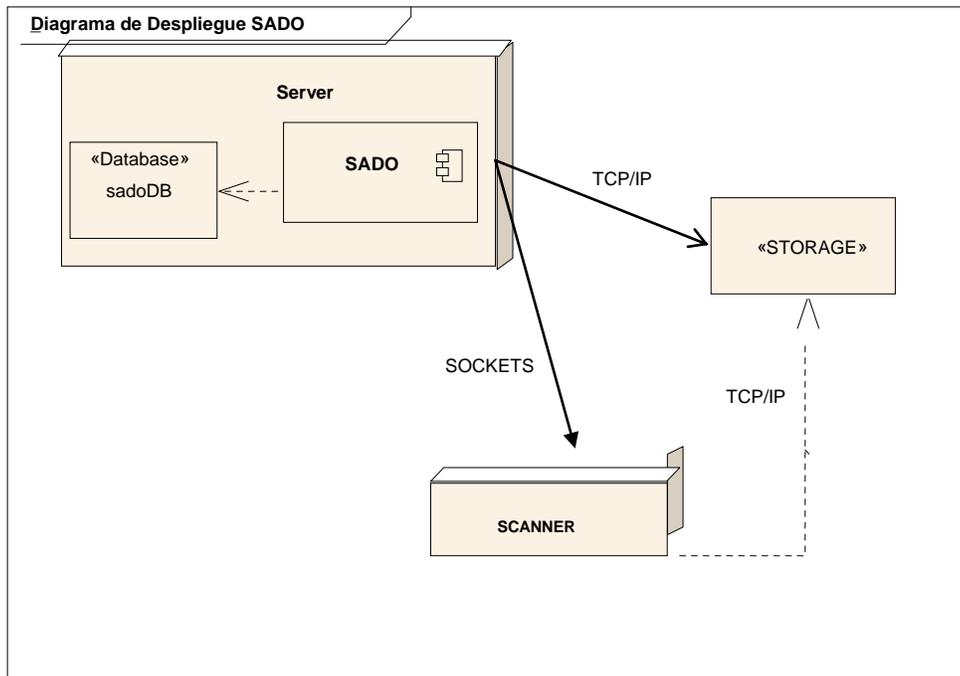


Figura 3.35 Diagrama de despliegue

3.15 Arquitectura

Es un sistema de Información que posee la siguiente arquitectura Cliente/Servidor de 3 capas como se muestra en la Figura 3.36:

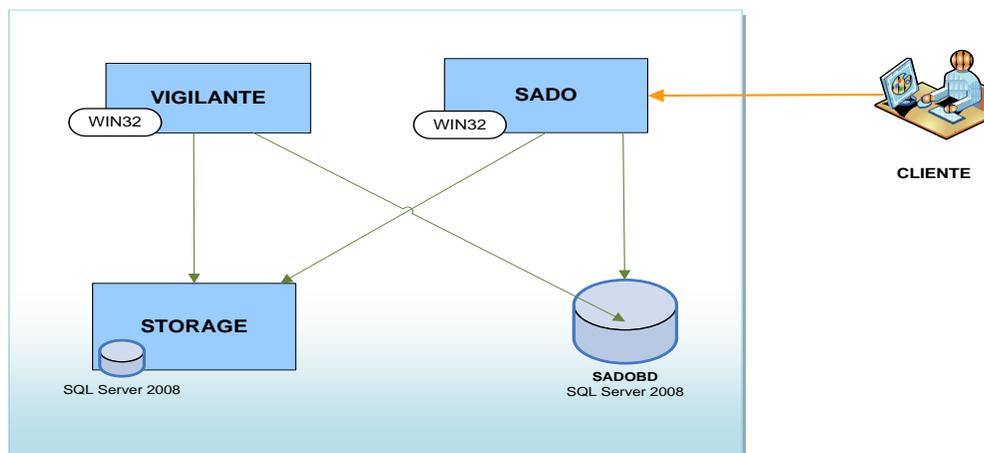


Figura 3.36 Diagrama de Arquitectura

CAPÍTULO IV

IMPLEMENTACIÓN SADO

4.1 Introducción

En este Capítulo se describirá paso a paso la instalación de diferentes componentes que son necesarios para la implementación del Sistema SADO.

Es recomendable seguir de forma ordenada, cada una de las instalaciones de las herramientas con el fin de tener un correcto funcionamiento de SADO.

4.2 Proceso de Instalación

4.2.1 Proceso de Instalación y Configuración de Aplicaciones

Para una correcta instalación de SADO debemos seguir los siguientes pasos:

- Instalación y Configuración de Adobe Capture
- Instalación y Configuración de los Roles de:
 - Active Directory
 - Print Server
 - File Server
- Instalación y Configuración de la Base de Datos
- Instalación y Configuración de SADO
 - LDAP Migrator
 - SADO (GUI)
 - SADO Vigilante

4.3 Implementación del Adobe Capture en el Servidor

Se debe aclarar que el producto Adobe Capture solo ayuda con el proceso de Escaneado de Documentos su modelo de flujo de trabajo permite definir bloques de trabajo para los diferentes tipos de documentos que el cliente necesite.

Al momento de la elaboración de este documento Adobe dio de baja el producto de su portafolio de soluciones pero el soporte para dicho producto estará vigente hasta finales del 2011, posteriormente sugiere que sus clientes migren al producto Adobe Lifecycle. Como se puede apreciar en el conjunto de imágenes a continuación:



Figura 4.1 Imagen de Bienvenida a la Instalación.

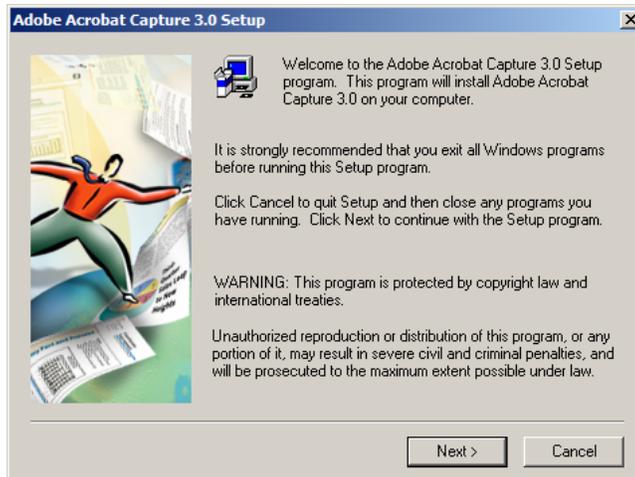


Figura 4.2 Imagen de Inicio de Instalación.

Se selecciona el Idioma de la Instalación.



Figura 4.3 Imagen de Selección de Idioma de Instalación.

Se debe Aceptar los términos de licencia de producto para completar la instalación como se puede apreciar en la Figura 4.4.

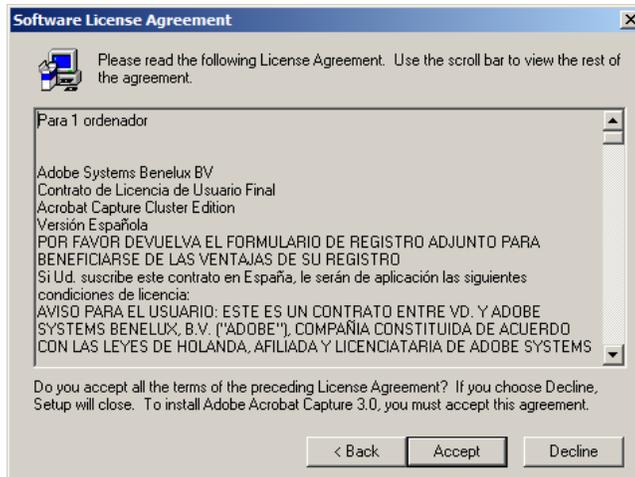


Figura 4.4 Imagen de Términos de Licencia.

Se procede a seleccionar la carpeta de instalación de la suite de captura como se puede apreciar en la Figura 4.5.



Figura 4.5 Imagen de Selección de Ubicación de la Instalación.

Se verifica la configuración de la Instalación, y se procede con la misma. Como se puede apreciar en la Figura 4.6.

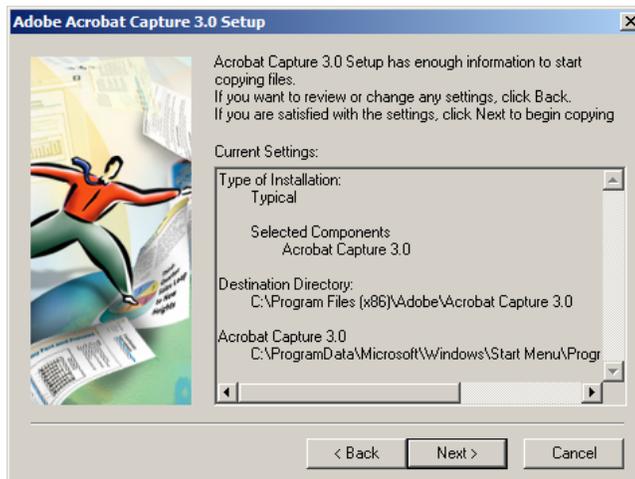


Figura 4.6 Imagen de confirmación de la Instalación.

Luego de completar la instalación , preguntara por reiniciar el servidor, lo cual es recomendable por que el instalador configura servicios de windows como el WIA, por lo que el reiniciar el servidor es prioritario para que el sistema comience a funcionar correctamente.

4.3.1 Configuración del Adobe Capture

Se procede a configurar el Adobe Capture por medio del suite. Al ser un software de servicios la interface posee opciones de definición, monitoreo y ejecución de trabajos. Como se puede apreciar en la Figura 4.7.

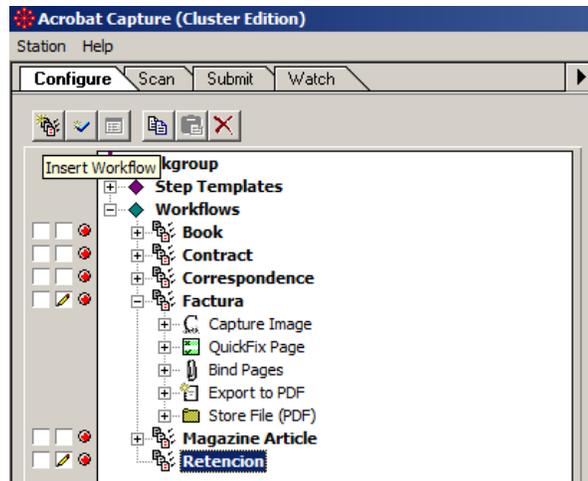


Figura 4.7 Imagen de Configuración de Adobe Capture.

Se procede a crear un nuevo Flujo de Trabajo, luego se configura para las necesidades que se plantearon, después de asignar un nombre a este trabajo. Como se puede apreciar en la Figura 4.8.

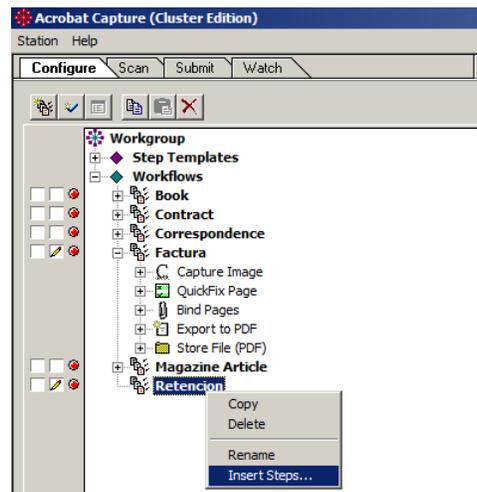


Figura 4.8 Imagen de creación de trabajo e ingreso de pasos a seguir.

En el Flujo de Trabajo que se va a seguir con nuestro documento, se debe tomar en cuenta la estructura de escanear, optimizar la imagen, reconocer el texto, grabar el PDF y exportar el PDF a la carpeta monitoreada de SADO Vigilante. Como se puede apreciar en la Figura 4.9 - 13.

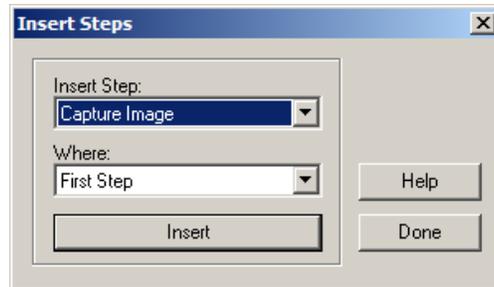


Figura 4.9 Imagen de Capturar de Imagen

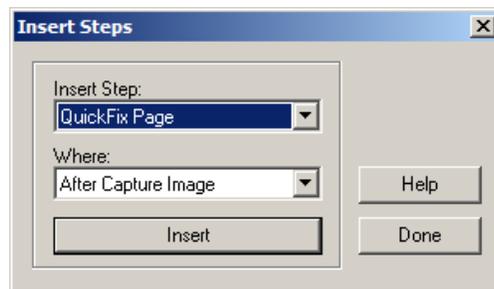


Figura 4.10 Imagen de Mejora de Imagen



Figura 4.11 Imagen de OCR

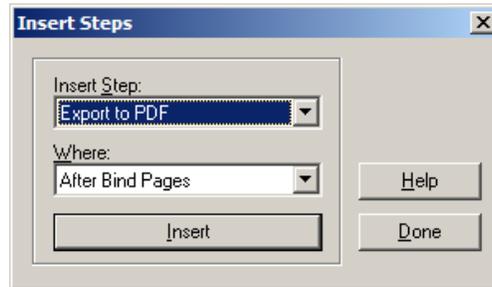


Figura 4.12 Imagen de Exportar a PDF

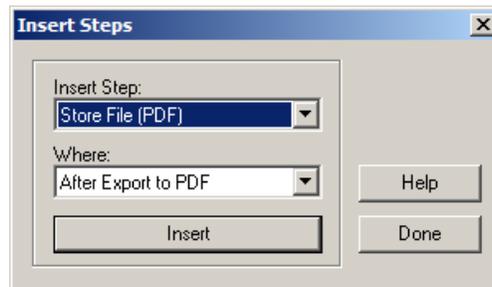


Figura 4.13 Imagen de Guardado en la Unidad de red.

Se debe especificar dónde será el almacenamiento de los documentos procesado.
Como se puede indicar en la Figura 4.14.

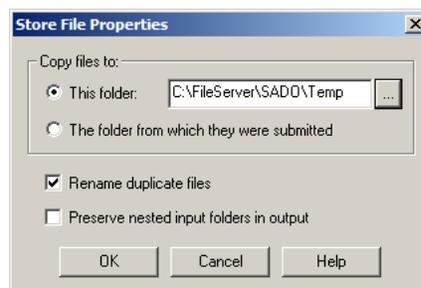


Figura 4.14 Imagen de Propiedades de Guardado en Disco

El programa necesita que el almacenamiento sea un recurso compartido para que sea accedido por la empresa, pero el programa Capture por sí solo no hace el trabajo completo, puesto que en la salida de los trabajos es donde SADO entra, en la organización de los documentos. Se debe revisar las diferentes configuraciones de los trabajos y acoplarlos a nuestras necesidades. Como se puede indicar en la Figura 4.15.

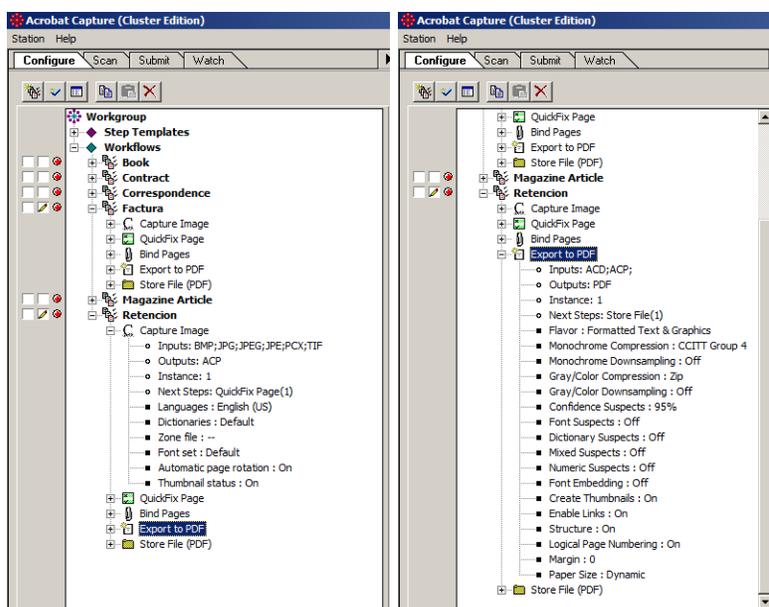


Figura 4.15 Imagen de Revision de las Propiedades de los Trabajos.

Una vez escogido el trabajo que vamos a realizar se procede al escaneado y posteriormente se puede ejecutar dicho trabajo para que SADO organice los documentos según la configuración planteada. Como se puede indicar en la Figura 4.16.

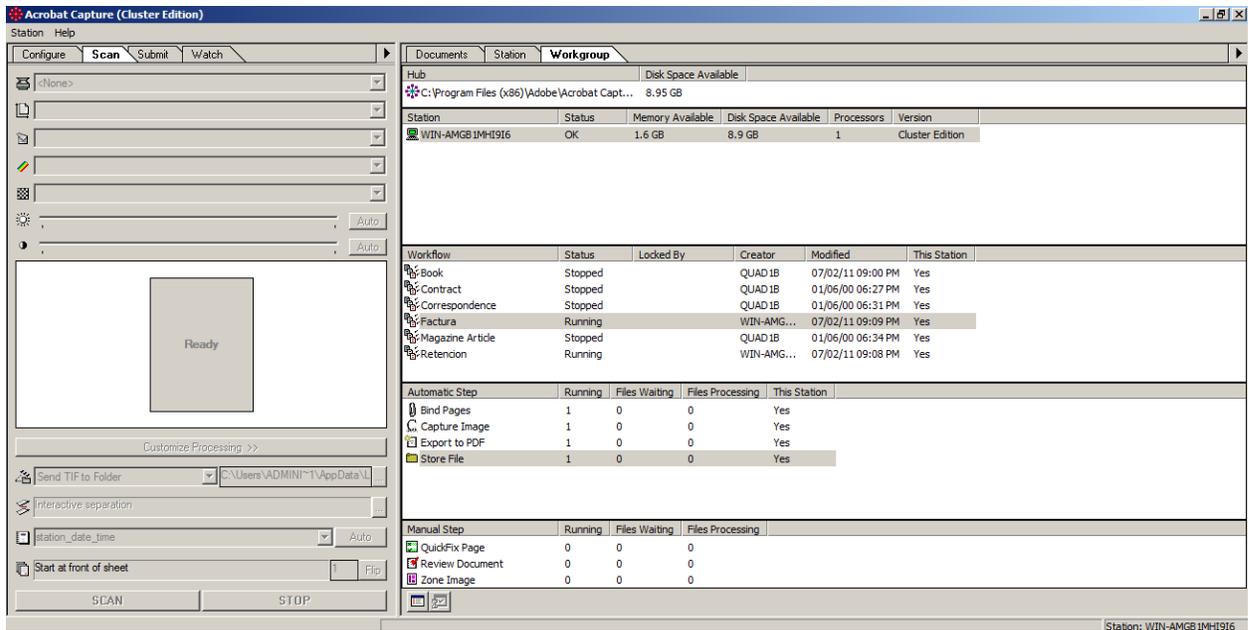


Figura 4.16 Imagen de WorkGroup de Acrobat Capture

4.4 Active Directory

4.4.1 Instalación de Active Directory

Para configurar este Controlador de Dominio vamos a proceder a instalar los servicios mínimos requeridos para la ejecución del mismo.

Usando el wizard de Windows server 2008 R2 para añadir roles a este servidor, se procede a seleccionar Active DirectoryDomainServices, como prerequisite solicita que se instale el .NET Framework 3.5.1. Como indica en la Figura 4.17.

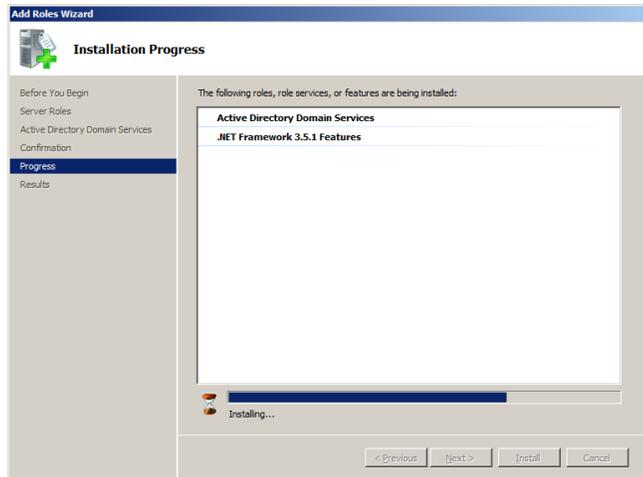


Figura 4.17 Imagen de estado de la instalación.

Una vez finalizada la instalación presenta el resumen de la instalación y el estado final de los roles instalados. Como indica en la Figura 4.18.

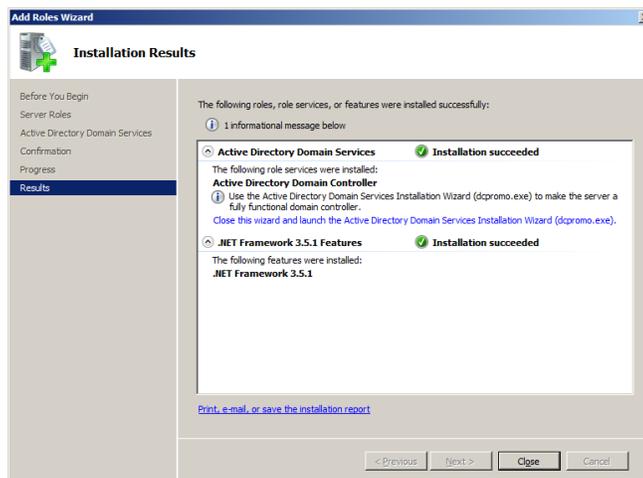


Figura 4.18 Imagen de Resultado de Instalación.

Al finalizar la instalación se debe ejecutar el Wizard de Windows Server 2008 R2 para promover dicho servidor a Controlador de Dominio, para lo cual se ejecuta “DCPROMO.EXE”

Una vez activada la herramienta de configuración se procede a instalar paso por paso el dominio de la empresa en este caso para soluciones inteligentes se usará el dominio “SOLUCINT.LOC”. Como indica en la Figura 4.19.

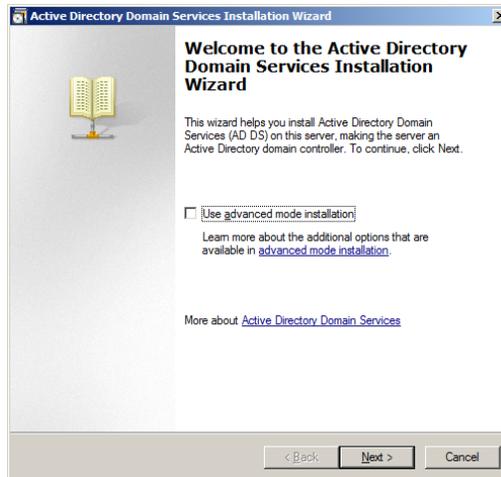


Figura 4.19 Imagen de Bienvenida al Wizard de Active Directory

La pantalla de bienvenida permite seleccionar el tipo de instalación, es recomendable mantener desactivado el modo avanzado por que vamos a crear un directorio nuevo, no unirnos a un dominio existente.

Se selecciona crear nuevo bosque de dominio y se procede a continuar, como se puede apreciar en la Figura 4.20.

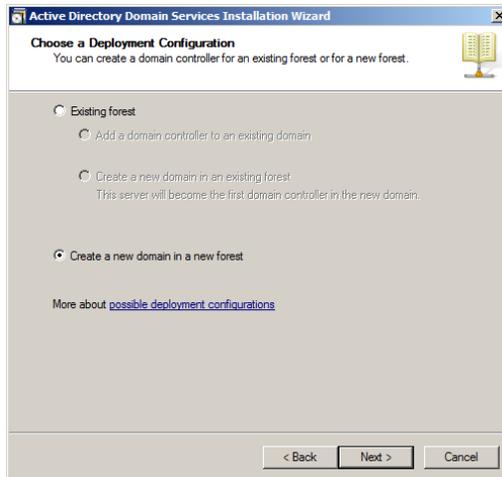


Figura 4.20 Imagen de creación de Nuevo Bosque de Dominio.

Se ingresa el dominio que se desea usar “solucint.loc” no utilizamos un dominio con la dirección web de la empresa debido a que al hacerlo nuestras búsquedas de dominio saldrían a internet volviendo el DNS lento en resolver direcciones locales. Como se puede apreciar en la Figura 4.21.

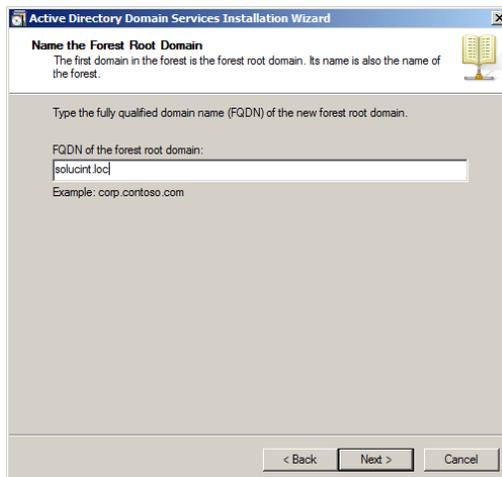


Figura 4.21 Imagen de nombre de dominio.

Al continuar con la instalación. el Wizard procederá con la respectiva validación del nuevo dominio. Como se puede apreciar en la Figura 4.22.

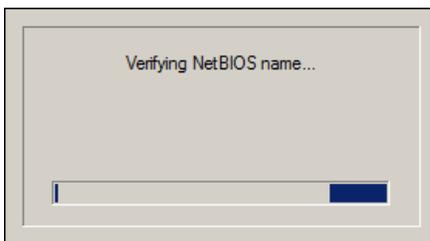


Figura 4.22 Imagen de Estado de la validación del nuevo dominio.

Una vez verificada la validez del dominio se procede a seleccionar el formato del dominio según nuestra infraestructura, se debe decidir qué tipo de servicios se quiere obtener, la versión de Active Directory de Windows Server 2008 R2 ofrece compatibilidad total con sus antecesores y adiciona características muy interesantes por lo cual es la que se escoge para nuestra instalación. Como se puede apreciar en la Figura 4.23.

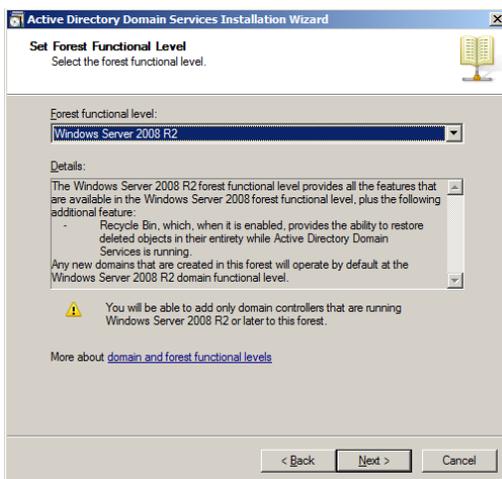


Figura 4.23 Imagen de Selección de Versión de Active Directory

Una vez seleccionada la versión de Active Directory, muestra los servicios adicionales que se debe instalar según el caso, para nuestra instalación adicionalmente se debe configurar el DNS para este dominio, siempre es recomendable que el controlador de dominio a su vez tenga el rol de DNS para evitar conflictos en la resolución de nombres dentro de nuestra organización, pero esto no significa que no podamos configurar el DNS en un servidor adicional pues es recomendable que estén juntos en el mismo servidor. Como se puede apreciar en la Figura 4.24.

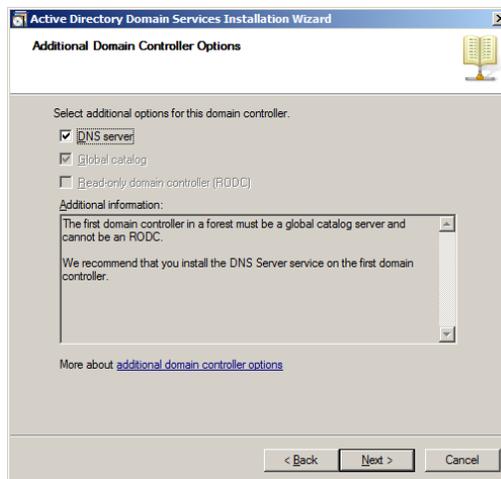


Figura 4.24 Imagen de recomendaciones adicionales de roles para el servidor.

De debe recordar que el controlador de dominio debe al menos en una de sus interfaces de red tener una IP estática, dependiendo de nuestra configuración de DHCP este paso puede omitirse. A continuación se puede apreciar en la Figura 4.25.

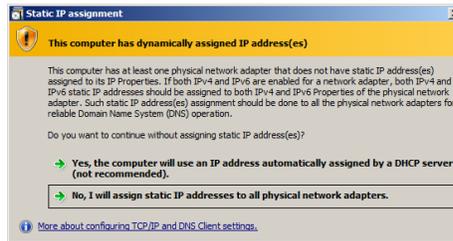


Figura 4.25 Imagen de Advertencia de IP Dinámica

Se define a dónde vamos a almacenar la DATABASE, el LOG y el SYSVOL de el dominio nuevo. A continuación se puede apreciar la pantalla de definición la Figura 4.26 .

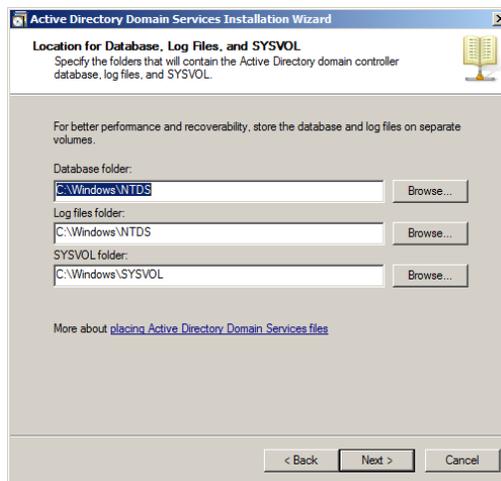


Figura 4.26 Imagen de Selección de Ubicación de Almacenamiento.

Se define un password administrativo para el administrador del Active Directory. Como se puede apreciar en la Figura 4.27.

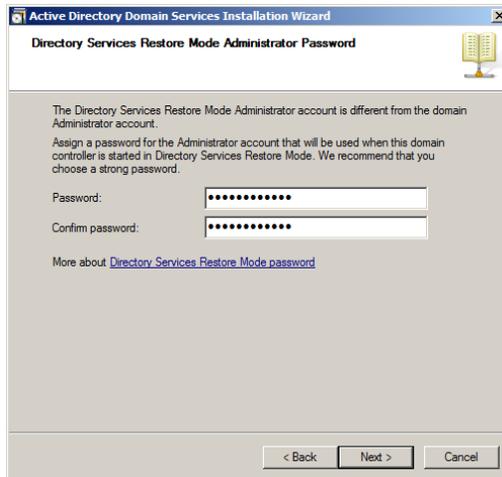


Figura 4.27 Imagen de Ingreso de Password de Administrador

Finalmente se puede revisar la configuración que se ha escogido para el nuevo dominio, si necesitara cambiar algo éste será el último punto para realizar cambios porque después de esto el Wizard procederá a la instalación del dominio con los parámetros escogidos. Como se puede apreciar en la Figura 4.28.

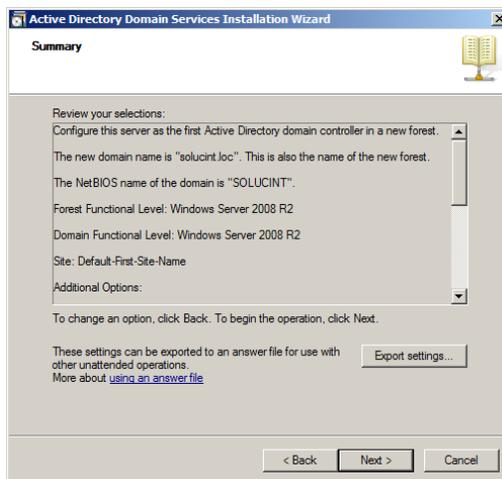


Figura 4.28 Imagen de Resumen de configuraciones

Se puede observar el proceso de configuración del dominio, este proceso dependiendo del servidor puede tomar hasta un máximo de 30 minutos. Como se puede apreciar en la Figura 4.29.

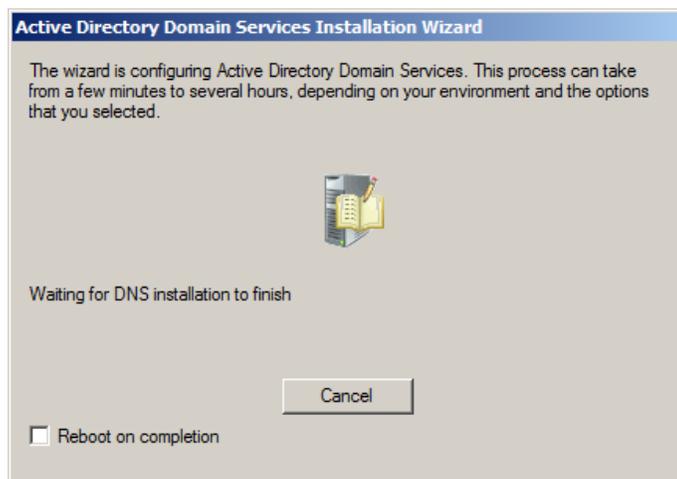


Figura 4.29 Imagen de estado de la configuración.

Una vez finalizada la instalación se presentara los resultados de la misma, de haber errores el Wizard procederá a realizar un rollback completo de la configuración para que después de corregir los errores poder realizar una configuración en limpio sin rastros de problemas de errores de instalación anteriores. Como se puede apreciar en la Figura 4.30.

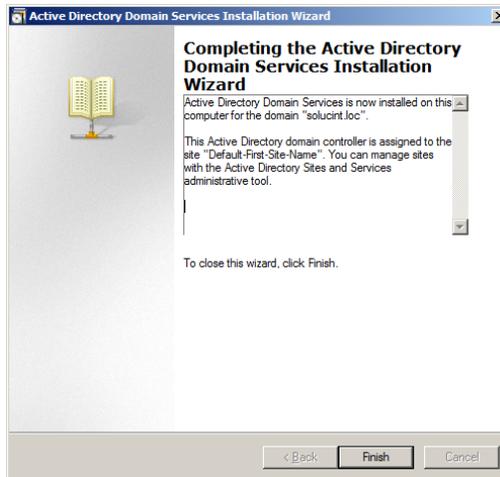


Figura 4.30 Imagen de Finalización de la instalación

Es recomendable reiniciar el servidor para que las configuraciones y los nuevos roles del servidor se activen y el perfil de administrador escogido, también se actualice al dominio nuevo. Como se puede apreciar en la Figura 4.31.

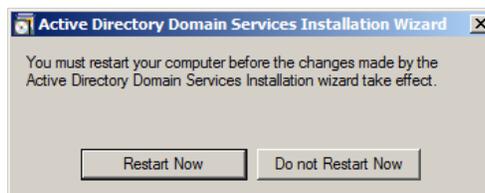


Figura 4.31 Imagen de Sugerencia de Reiniciar el Servidor.

4.4.2 Configuration Active Directory

Dentro del servidor de Active Directory se debe definir una estructura básica para mantener organizado el directorio, se procede desde la creación de 2 grupos Principales o Unidades Organizacionales una para Grupos y Otra para Usuarios

4.4.3 Creación de Grupos

Se crea una nueva Unidad Organizacional, para almacenar los Grupos de la Organización y otra para almacenar a los usuarios, lo recomendable es que dentro de dichos grupos, se cree una estructura de áreas igual a la de la empresa. Como se hace referencia a la Figura 4.32-34.

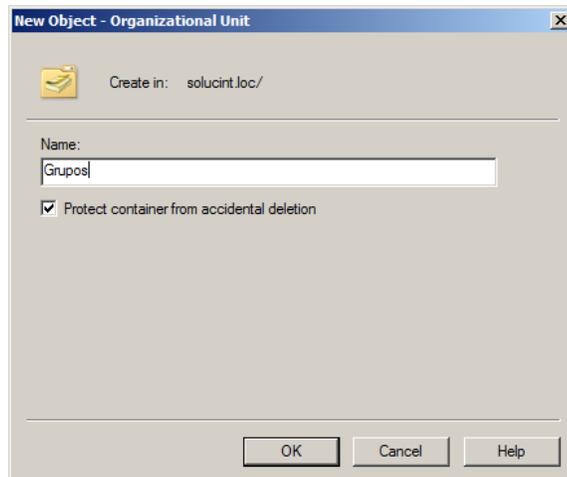


Figura 4.32 Imágenes de creación de las unidades Organizacionales Generales (Grupos).

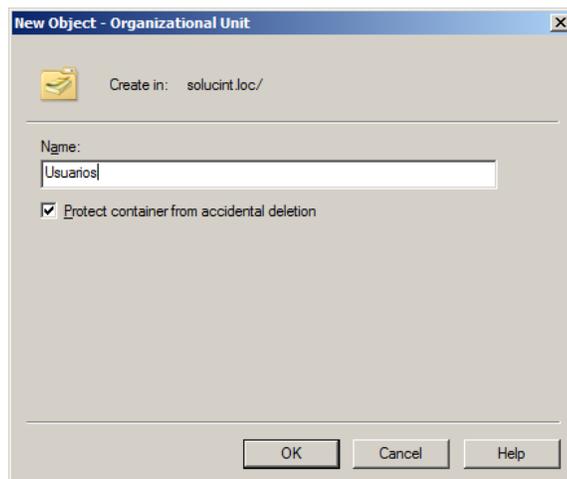


Figura 4.33 Imágenes de creación de las unidades Organizacionales Generales (Usuarios).

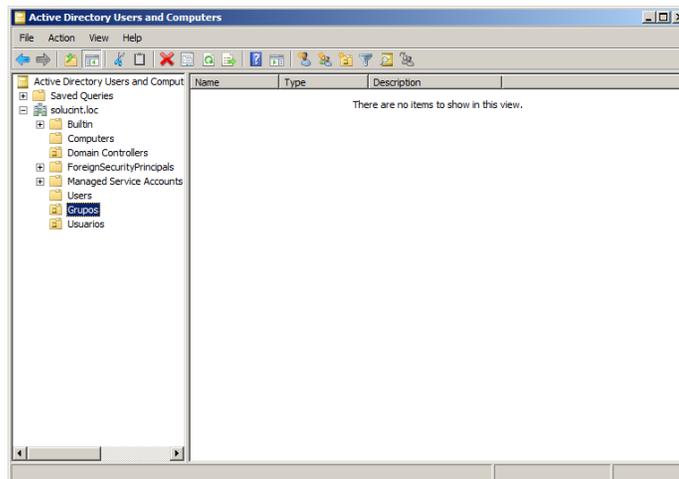


Figura 4.34 Imagen del Active Directory Users and Computers de nuestra organización.

Para una correcta administración del directorio escogido, es recomendable crear un Grupo Global o Grupo de Organización al cual todos los Grupos de Áreas pertenecerán posteriormente.

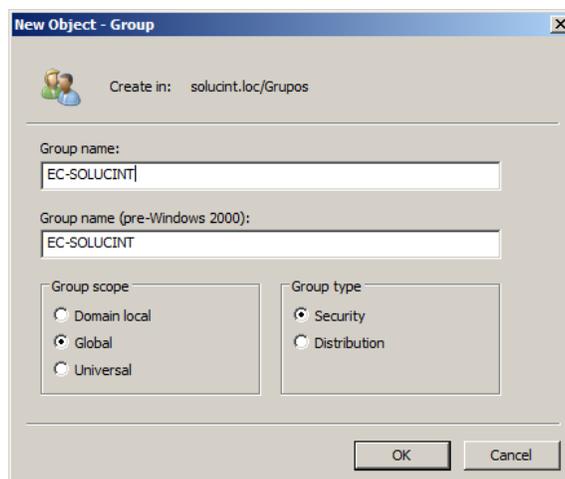


Figura 4.35 Imagen de Creación del Grupo Global.

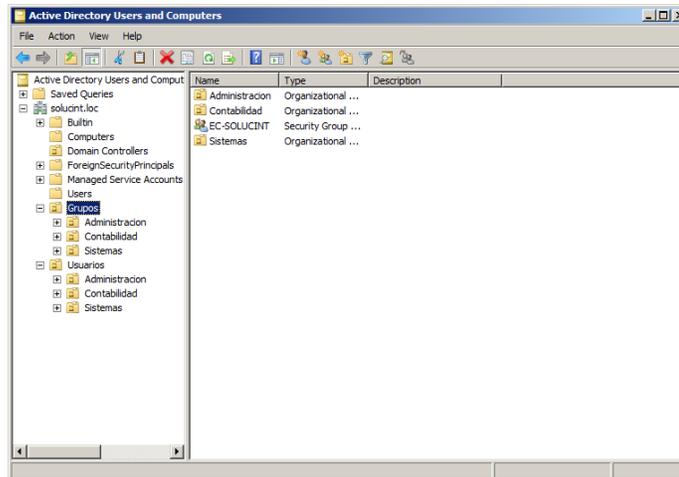


Figura 4.36 Imagen de la Organización actual dentro del Directorio.

4.4.4 Creación de Grupos de Usuarios

La Creación de Grupos de Usuario facilita la administración de permisos dentro de la organización a futuro, si bien en este momento no lo requerimos posteriormente para beneficio de la organización los creamos.

También es recomendable que la estructura que se utiliza para los nombres lleve un formato básico, el que se escogió para la Organización va de la siguiente manera:

- EC-AREA-Usuario-PERMISOS
- EC = País
- AREA = Área dentro de la Organización
- Usuario = Nombre de Usuario
- PERMISOS = Permisos dentro de la Organización

Como se puede observar en la figura 4.73

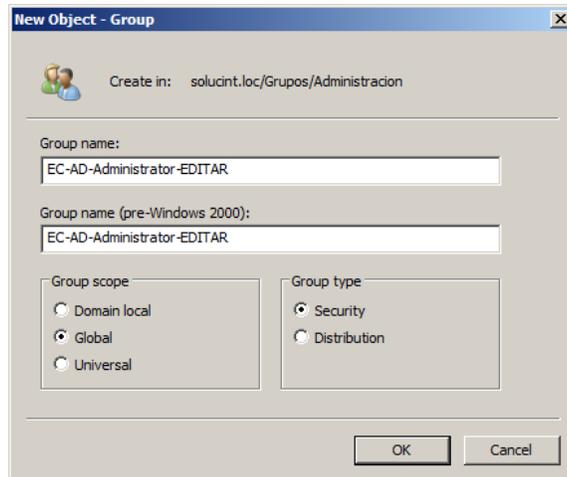


Figura 4.37 Imagen de Creación de Grupo para el Usuario.

Posteriormente se añade el usuario para este grupo, en este caso es el administrador de la red. Como se puede apreciar en la Figura 4.38.



Figura 4.38 Imagen de Búsqueda de Usuario.

Por último se verificó que el usuario quede dentro del grupo correcto. Como se puede apreciar en la Figura 4.39.

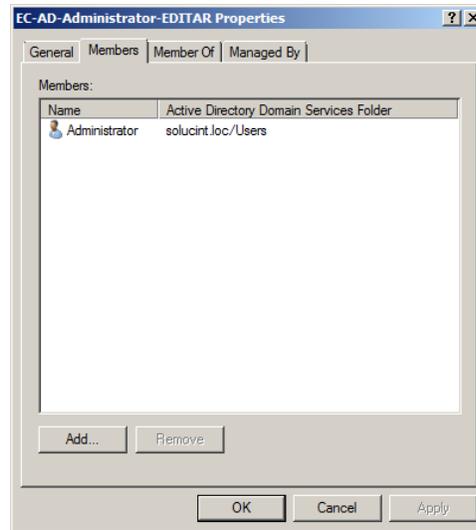


Figura 4.39 Imagen del Grupo para el Usuario.

4.4.5 Creación de Usuarios

Es importante que todos los miembros de la Organización cuenten con su propio usuario dentro del dominio, porque en temas de auditoría posteriormente se lleva un registro del usuario que trabaja sobre el sistema.

Al igual que con los Grupos de Usuarios se debe mantener un estándar que sirva para identificar a los usuarios, para la organización se escogió:

Nombre.Apellido

En las figuras anteriores se indica la creación de usuario, password en la Figura 4.40

- 42.

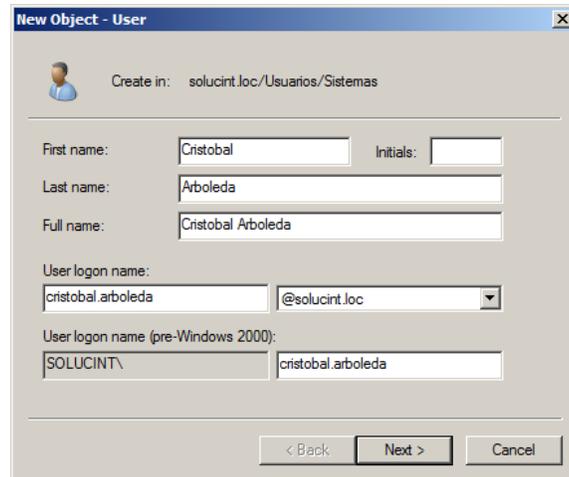


Figura 4.40 Imagen de Creación de Usuario.



Figura 4.41 Imagen de Asignación de Password.

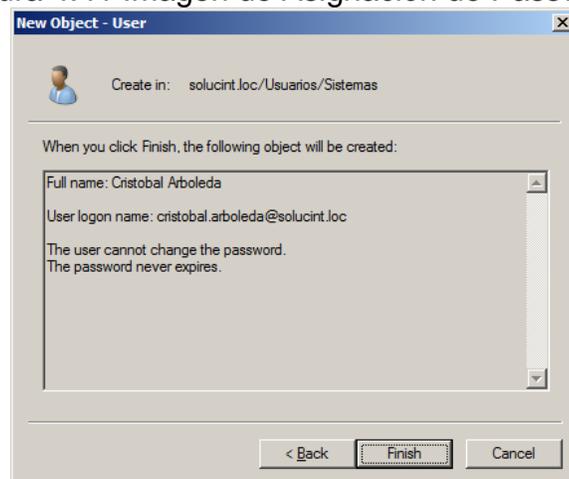


Figura 4.42 Imagen de Finalización de la creación de Usuario.

Posteriormente se debe Crear el Grupo del Usuario y asignarle a este el usuario creado como se puede observar en la Figura 4.43 - 45.

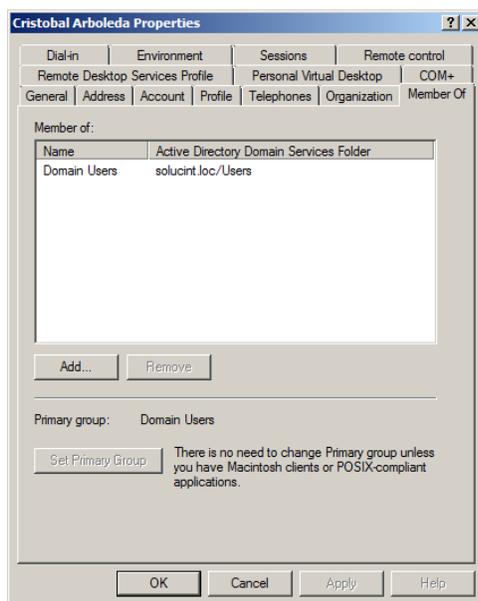


Figura 4.43 Imagen de Grupos a los que pertenece el Usuario.

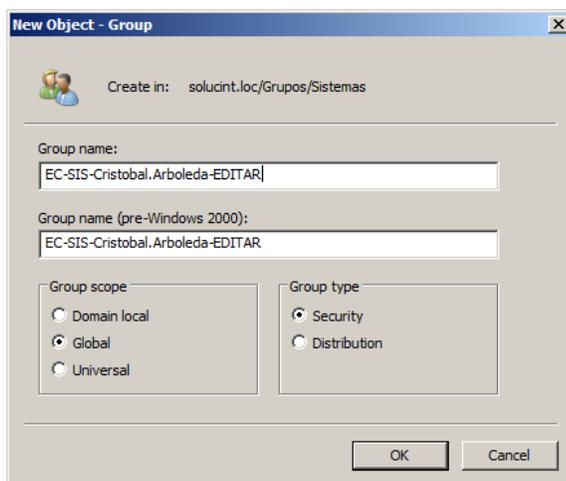


Figura 4.44 Imagen de creación del Grupo del Usuario.

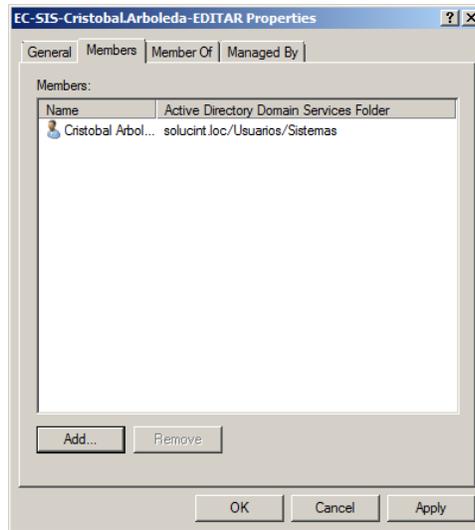


Figura 4.45 Imagen final del Grupo del Usuario con su respectivo usuario asignado.

No se debe olvidar que a este Grupo Global se debe asignarle todos los grupos de área, y a éstos los diferentes grupos de usuario, creando así una jerarquía de Grupos la cual debe estar basada en la jerarquía actual de la empresa. Como se puede apreciar en la Figura 4.46.

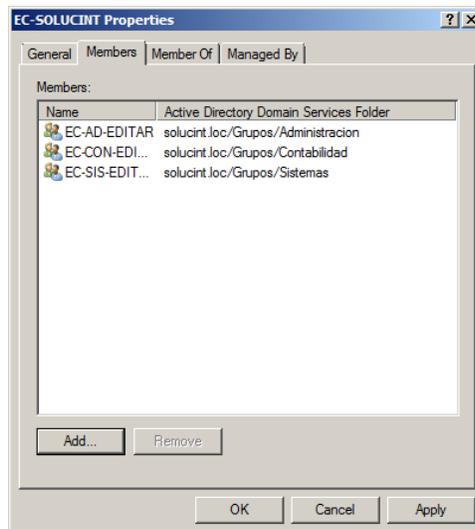


Figura 4.46 Imagen de Jerarquía desde el Grupo Global.

4.5 Print Server

4.5.1 Instalación y Configuración de Print Server

Para configurar este Servidor de Impresión vamos a utilizar el Wizard de Windows Server 2008 R2 añadiendo el nuevo Rol a el Servidor.

Usando el Wizard se procede a seleccionar Print and Document Services, éste es un prerequisite para el servicio que interesa que es File Server como se muestra en la Figura 4.47 a continuación.

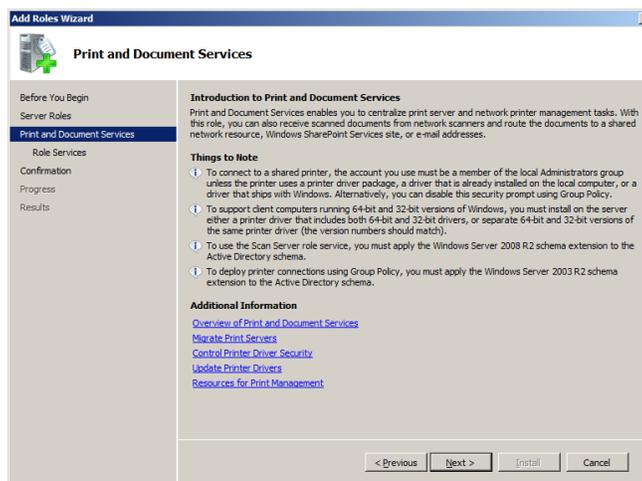


Figura 4.47 Imagen de Bienvenida a la Instalación del Nuevo Rol.

Solamente se debe escoger la opción de PrintServices debido a que por el momento solo se necesita la instalación para la posterior configuración del Rol de File Server.

Como se muestra en la Figura 4.48.

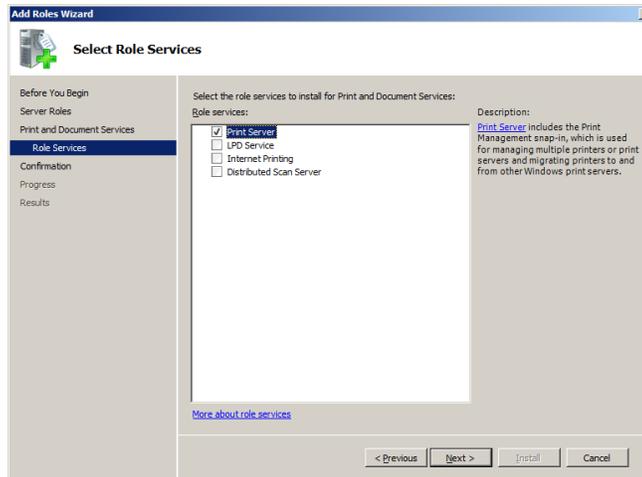


Figura 4.48 Imagen de Selección de Funciones del Rol.

Se confirma la Instalación del Nuevo Rol de Print and DocumentServices, esta función no tiene mayor configuración como se puede observar en la Figura 4.49 a continuación.

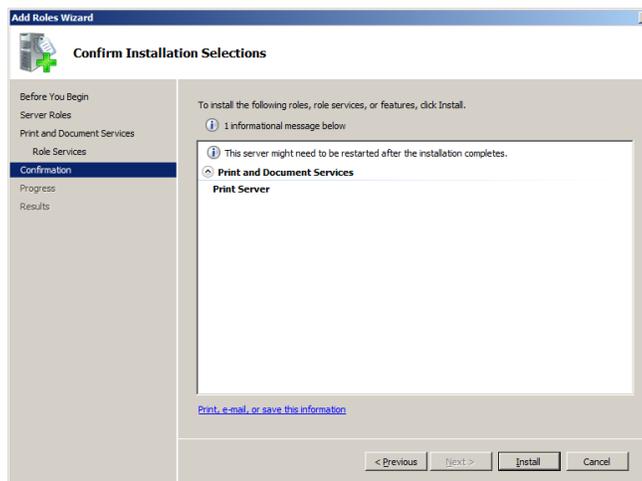


Figura 4.49 Imagen de Confirmación de la Instalación.

Se espera que finalice la instalación del nuevo rol, este rol del servidor no requiere un reinicio del sistema por lo cual al finalizar esta instalación se procede con la instalación del Rol de File Server como se puede indicar en la Figura 4.50.

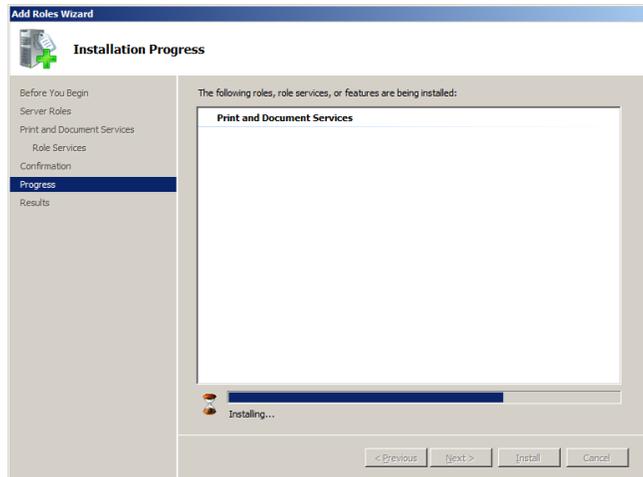


Figura 4.50 Imagen de Progreso de Instalación Print Server

4.6 File Server

4.6.1 Implementación del FileServer

Una vez instalado el Rol de Print and DocumentServices, aparece el Wizard de Windows Server 2008 R2 el Rol de File Service o File Server procese procede a seleccionarlo y a continuación con la instalación de dicho Rol como se aprecia en la Figura 4.51.

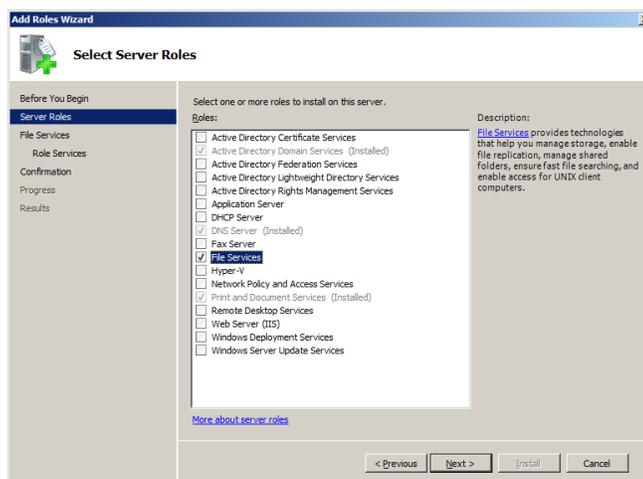


Figura 4.51 Imagen de selección de Nuevo Rol de File Service.

Se selecciona las Funciones que se necesita del Rol, para esta aplicación se requiere de:

- File Server,
- File Server Resource Manager,
- Windows SearchService

Como se indica en la siguiente figura 4.52.

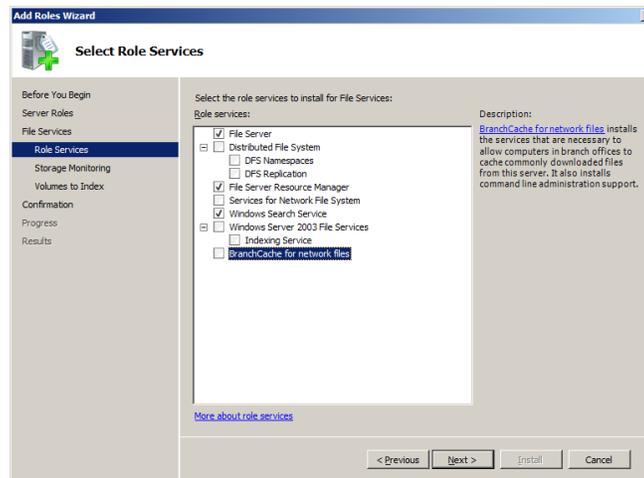


Figura 4.52 Imagen de Selección de Funciones del File Server.

Se procede con la Configuración de la Unidad donde se almacenara la información de este File Server, por defecto selecciona la unidad con mayor espacio disponible, pero se puede seleccionar nuestra unidad según las necesidades y la configuración de la empresa, es perfectamente válido utilizar almacenamientos de tipo NAS, SAN, RAID, etc.

Adicionalmente se puede configurar alertas sobre espacio y restricciones de tipos de archivos sobre el storage de nuestra red como se muestra en la Figura 4.53.

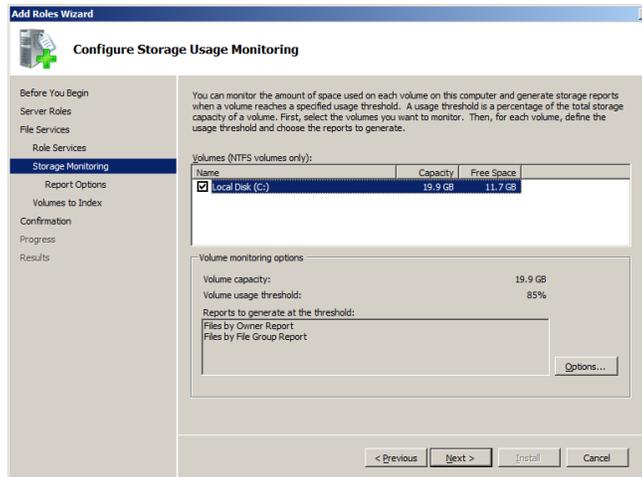


Figura 4.53 Imagen de Configuración de la Unidad de Storage del FileServer.

Se puede definir parámetros para recibir informes sobre el File Server y se define donde se desea recibir como se indica en la Figura 4.54.

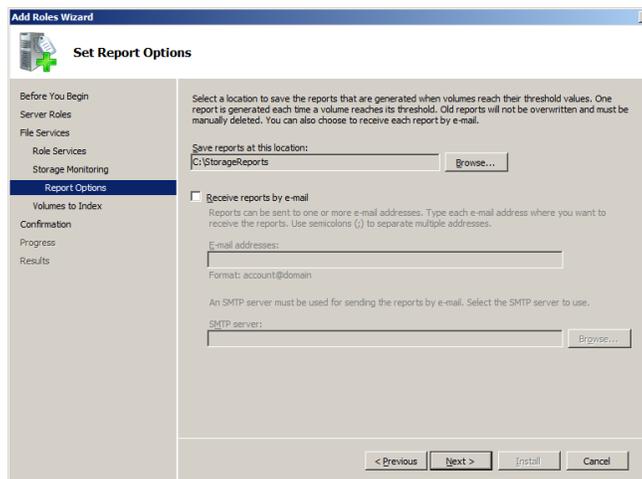


Figura 4.54 Imagen de Configuración de Reportes sobre el FileServer.

Al seleccionar el servicio de Windows Search se procede en este punto a configurar qué volúmenes de ese almacenamiento, se desea indexar para facilitar las búsquedas. Como se indica en la Figura 4.55.

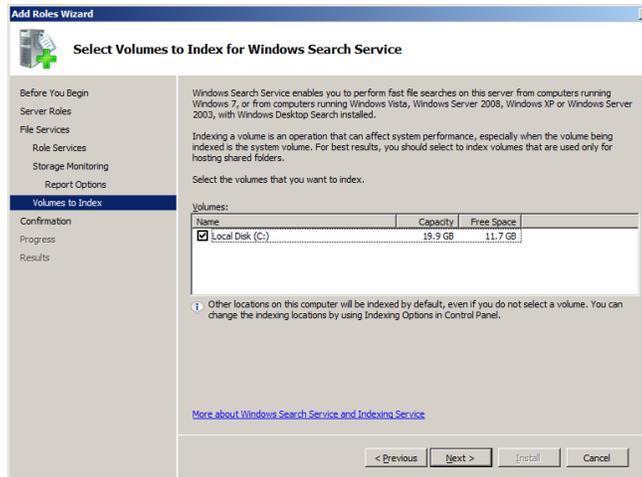


Figura 4.55 Imagen de Configuración de Indexación de Unidades.

Se revisa las Configuraciones y se comienza con la instalación del Rol en el servidor, el proceso dependiendo del espacio de almacenamiento y los archivos que se encuentren almacenados puede tomar un tiempo considerable, siempre es recomendable que se realice la instalación sobre una unidad vacía y posteriormente se copie la información deseada, como se puede apreciar en la Figura 4.56 a continuación.

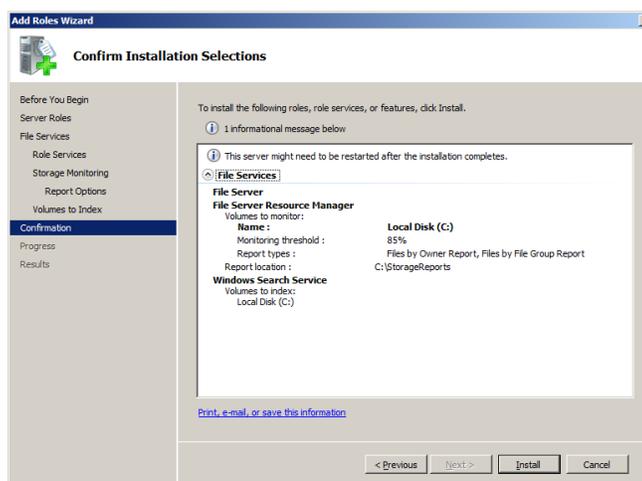


Figura 4.56 Imagen de Resumen de la Instalación del File Server.

Se procede con la finalización de la instalación, y es recomendable reiniciar el servidor pero no es obligatorio. Como se indica en el Resumen de la Instalación en la Figura 4.57.

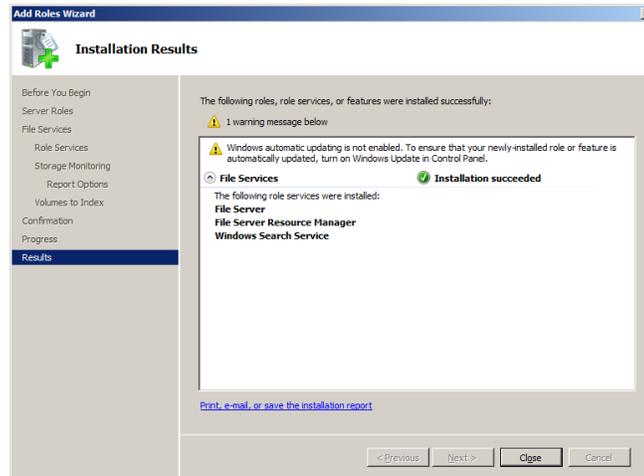


Figura 4.57 Imagen de Resumen de la Instalación.

4.6.2 Configuración de File Server

Se cree que la estructura básica del File Server, para lo cual se crea una carpeta que defina la ruta raíz dentro del almacenamiento, para esta configuración se necesita una carpeta que se llame FileServer. Como se indica en la Figura 4.58.

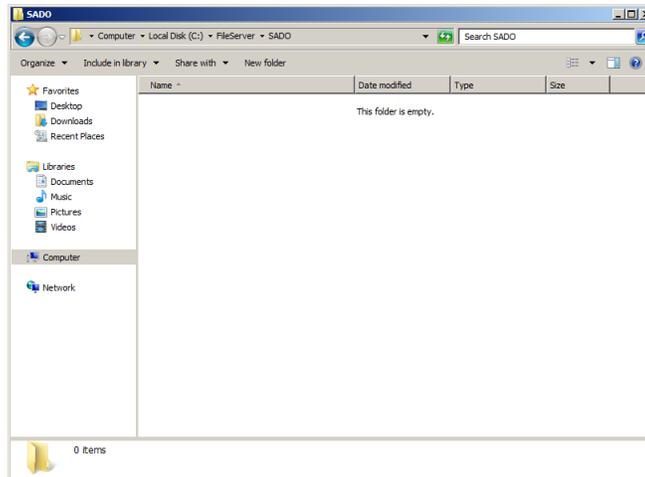


Figura 4.58 Carpeta dentro del almacenamiento.

Posteriormente se va a configurar dicha carpeta como Repositorio Compartido dentro de esta Red, para lo cual se procede con un compartimiento Avanzado como se aprecia en la Figura 4.59.



Figura 4.59 Opciones de Compartido de la Carpeta.

Se selecciona el nombre del recurso compartido y revisamos los permisos para el acceso del dominio. Como se muestra en la Figura 4.60-61.

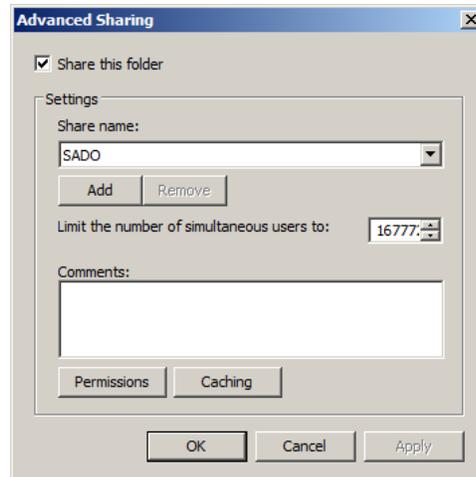


Figura 4.60 Configuración del recurso compartido.

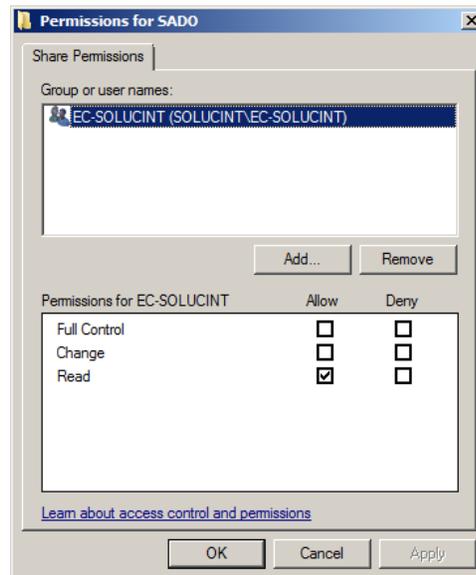


Figura 4.61 Permisos dentro del Dominio.

Ahora mediante el File Server Resource Manager, se define un Quote de almacenamiento, para que administre la cantidad de información, los tipos de archivos y notifique cuando existan anomalías dentro del almacenamiento como se indica a continuación en la Figura 4.62 - 64.

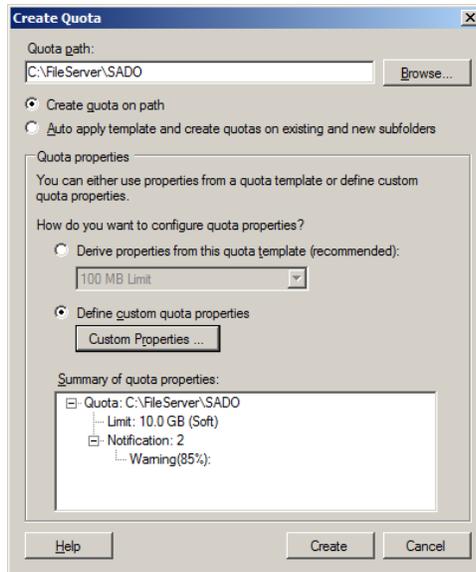


Figura 4.62 Ubicación del Recurso

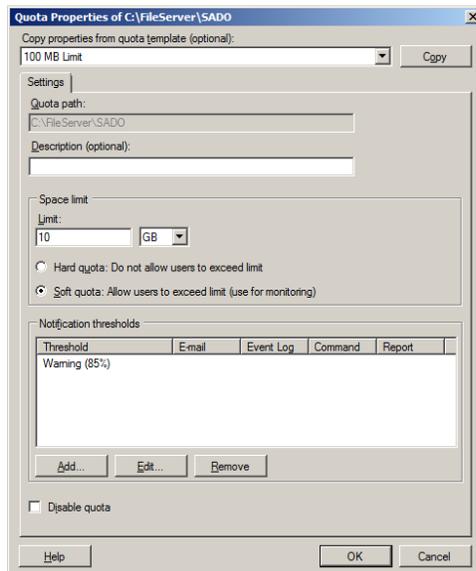


Figura 4.63 Parámetros de almacenamiento.

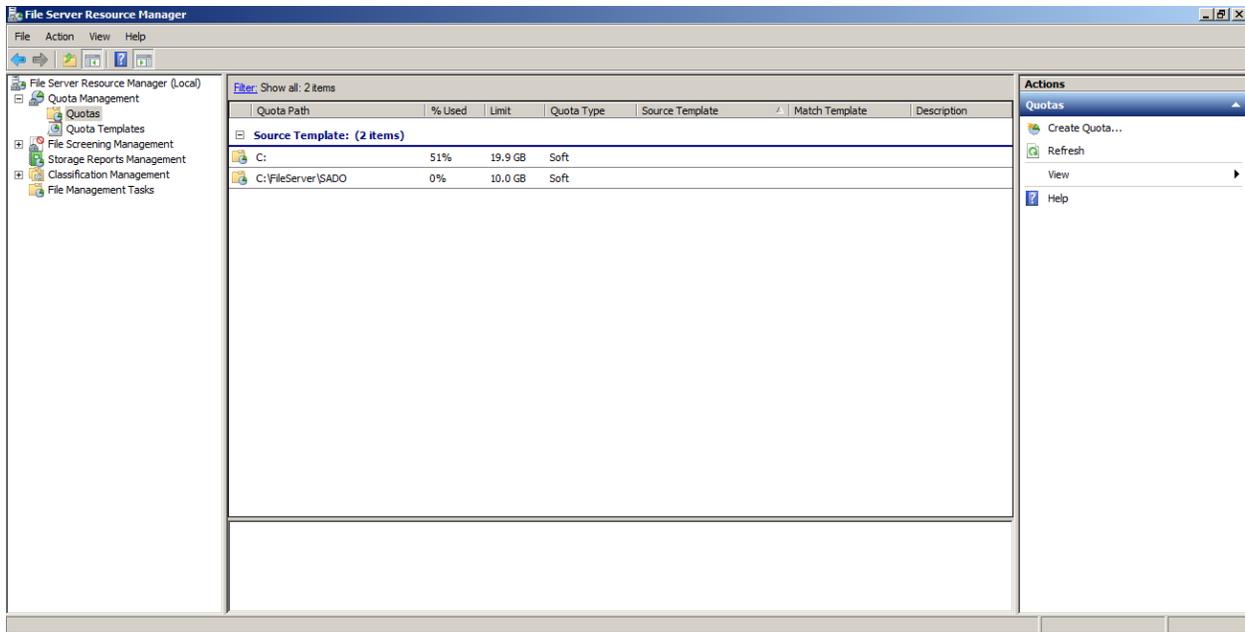


Figura 4.64 Imagen del File Server Resource Manager con el Recursos Monitoreados.

4.7 Base de Datos

4.7.1 Instalación de la Base de Datos

Se debe comenzar con la instalación seleccionando si vamos a instalar un servidor nuevo. A continuación se puede indicar en la Figura 4.65.



Figura 4.65 Imagen de Instalación Nueva

Una vez seleccionada la instalación se debe tomar en cuenta los chequeos de prerequisites, en este momento si se presenta algún error se debe solucionarlo caso contrario no se puede continuar con la instalación. Como se indica en la imagen Chequeo de Prerrequisitos en la Figura 4.66.

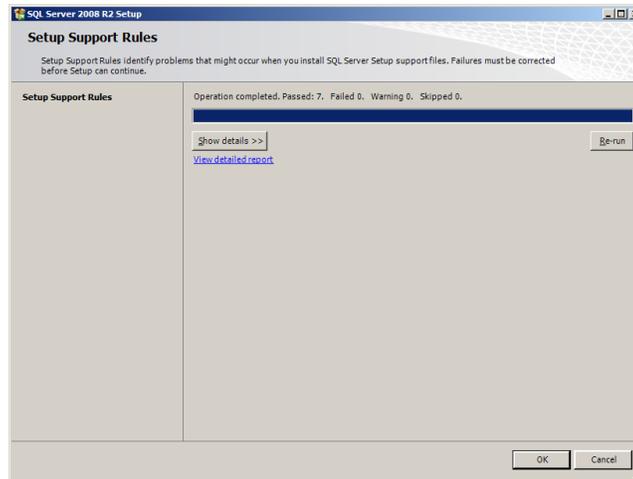


Figura 4.66 Imagen de Chequeo de Prerrequisitos.

Posteriormente se debe especificar el Key de Instalación como se indica en la Figura 4.67.

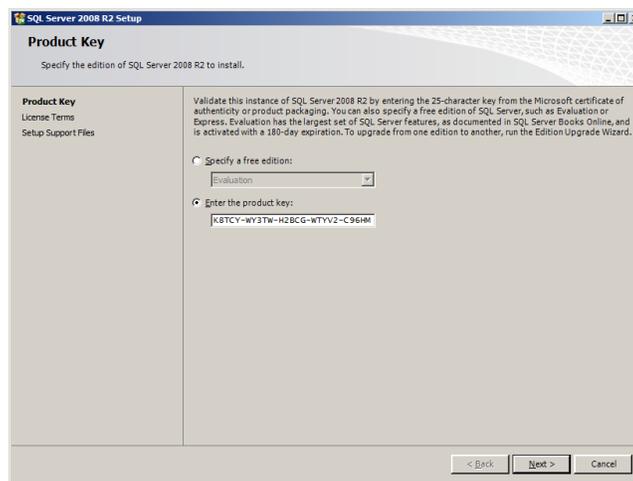


Figura 4.67 Imagen de Key de Instalación.

Se deben aceptar los términos de licencia para poder comenzar la instalación del Servidor de Base de Datos. Como se aprecia en la Figura 4.68.

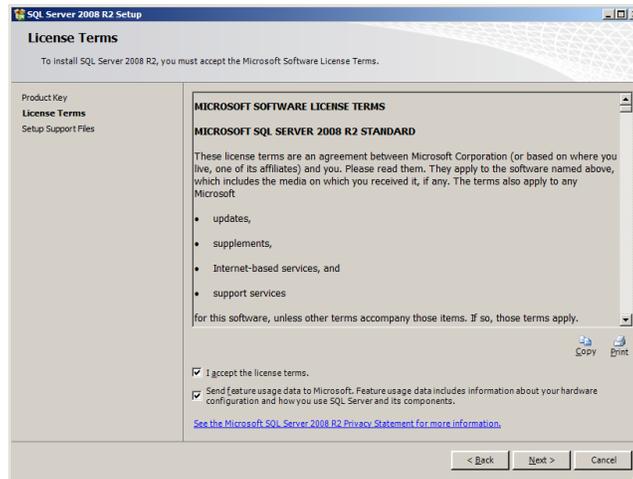


Figura 4.68 Imagen de Aceptación de Licencia.

Ahora la instalación indica que se debe instalar un prerequisite, que consiste en los archivos de ayuda de producto, posteriormente se podrá comenzar con la instalación del servidor. Como se aprecia en la Figura 4.69.

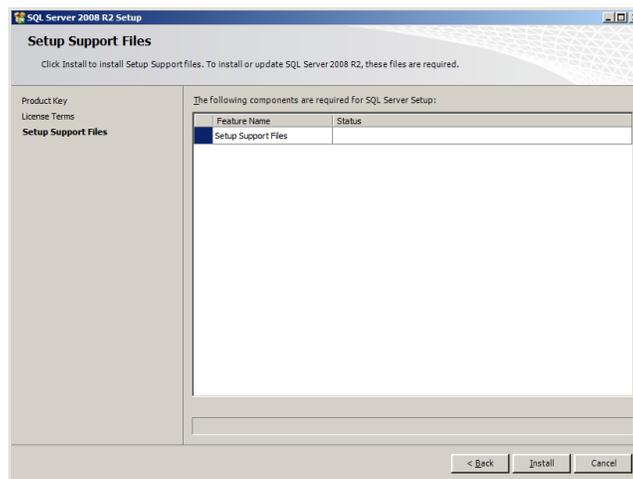


Figura 4.69 Imagen de Instalación de archivos de ayuda.

Ahora el instalador vuelve a correr un test del sistema para ejecutar la instalación de los servicios de Base de Datos. Como se aprecia en la Figura 4.70.

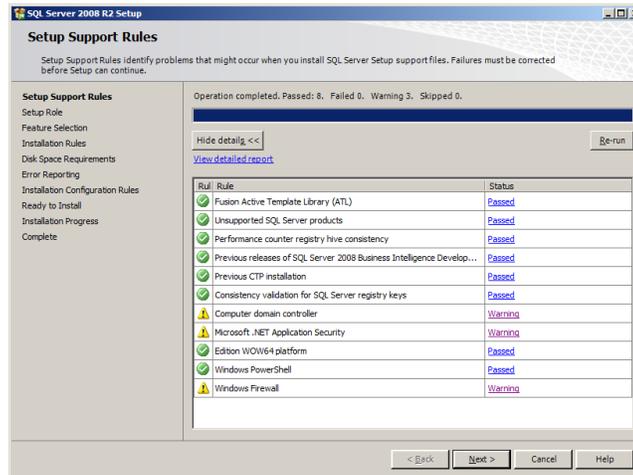


Figura 4.70 Imagen de chequeo de prerequisites de instalación.

Se selecciona el tipo de instalación según las necesidades, para esta aplicación vamos a optar por escoger los productos que vamos a instalar como se puede ver a continuación en la Figura 4.71.

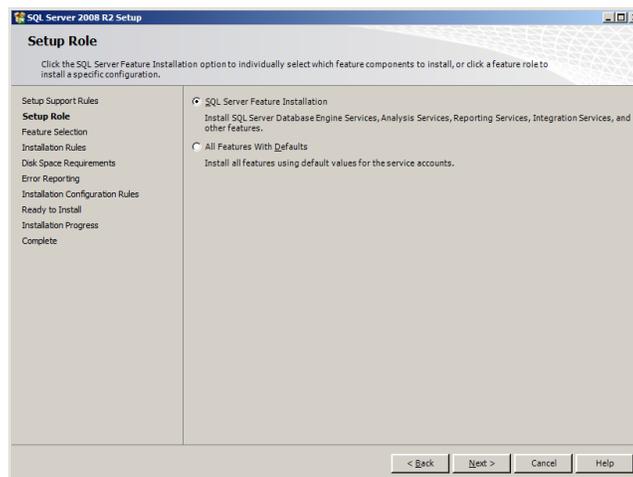


Figura 4.71 Imagen de selección del tipo de instalación.

Los productos que se necesita para este sistema son:

- Database Engine Services – Motor de la Base de Datos.
- Client Tools Connectivity – Componentes de Conexión Remota.
- Client Tools Backward Compatibility – Compatibilidad con Versiones anteriores de la Base de datos.
- SQL Server Books Online – Ayuda sobre SQL Server.
- Management Tools – Herramientas de Administración de la Base de datos.

Como se puede apreciar en la Figura 4.72.

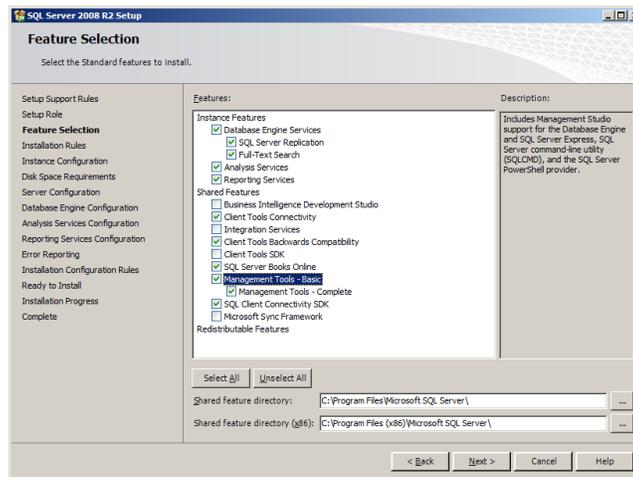


Figura 4.72 Imagen de Selección de Componentes a instalar.

Ahora se define el nombre de la instancia que se va a utilizar, al ser un servidor nuevo se puede dejarla por defecto, caso contrario se debe darle un nombre, también se selecciona donde se va a instalar el servidor en este caso la información por defecto, y se continua con la Instalación como se procede en la Figura 4.73.

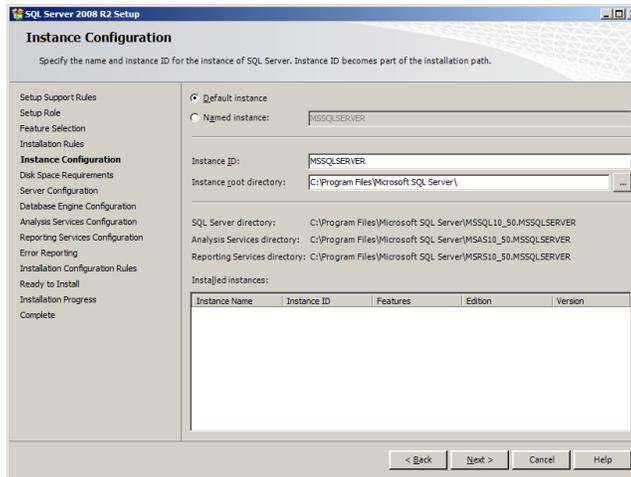


Figura 4.73 Imagen de la selección de la instancia y de la ubicación de la instalación.

Posteriormente se define las cuentas con las que funcionaran estos servicios es recomendable que se mantengan la misma cuenta por motivos de administración y de ser posible una cuenta creada por este proyecto como se indica a continuación en la Figura 4.74.

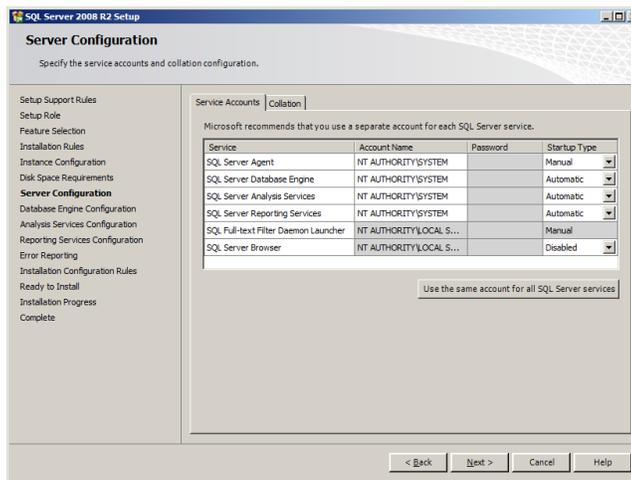


Figura 4.74 Imagen de Selección de Cuentas para Servicios.

Ahora se procede a definir el modelo de autenticación sobre este sistema, para el funcionamiento de SADO es de modo Windows, pero por motivos de solución de

problemas se elige la opción Mixta y se define un password para el administrador (sa) debido a que si en algún momento el controlador de dominio no está disponible, se puede conectarse al servidor de base de datos sin problemas.

También se procede a definir el usuario de dominio que funciona como administrador del servidor como se observa en la Figura 4.75.

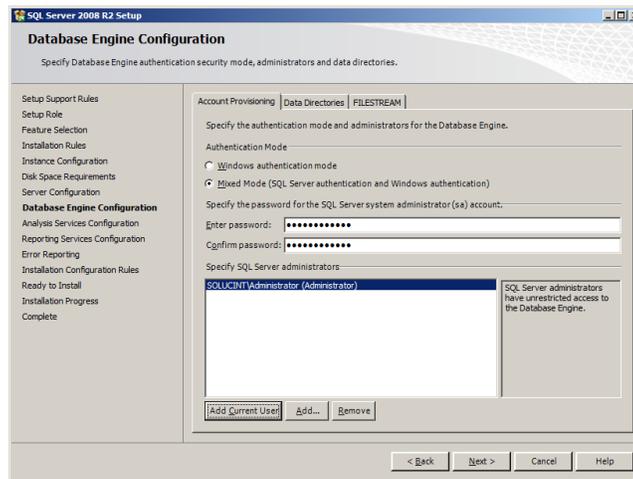


Figura 4.75 Imagen de selección de Tipo de autenticación y Administrador.

Por último se revisa la configuración seleccionada antes de realizar la instalación del servidor, si se requiere realizar algún cambio ésta es la última oportunidad de realizarlo, caso contrario posteriormente se puede revisar los componentes instalados mediante agregar o quitar características del servidor. A continuación la Figura 4.76 que muestra el resumen.

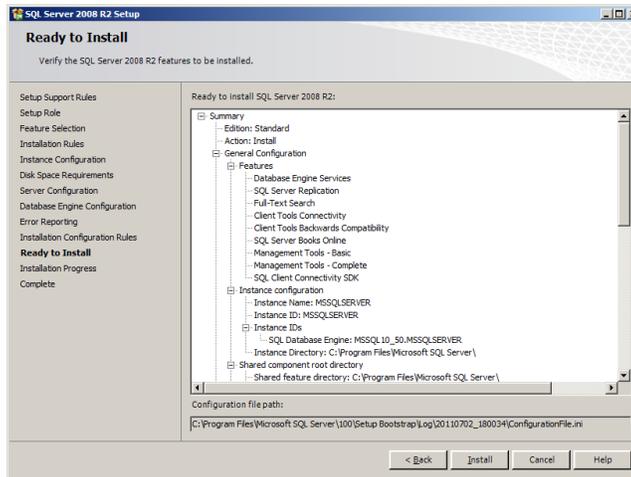


Figura 4.76 Imagen de Revisión de la Configuración de la Instalación.

Se procede a continuar con la instalación, y luego de que termine se mostrará un resumen de la misma y si esta fue exitosa o no. Como se observa a continuación en la Figura 4.77.

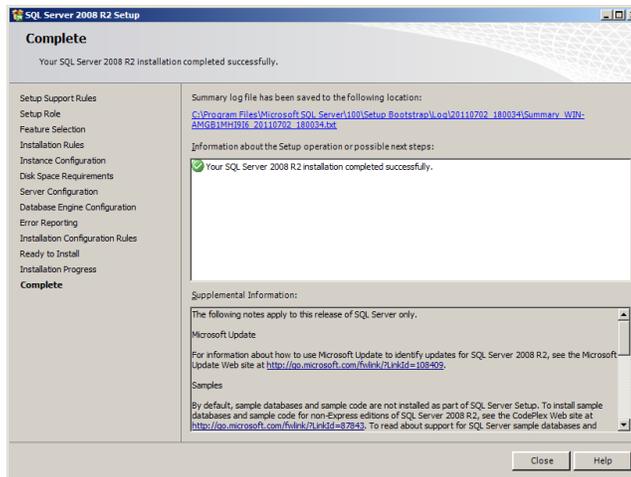


Figura 4.77 Imagen de Resumen de la Instalación.

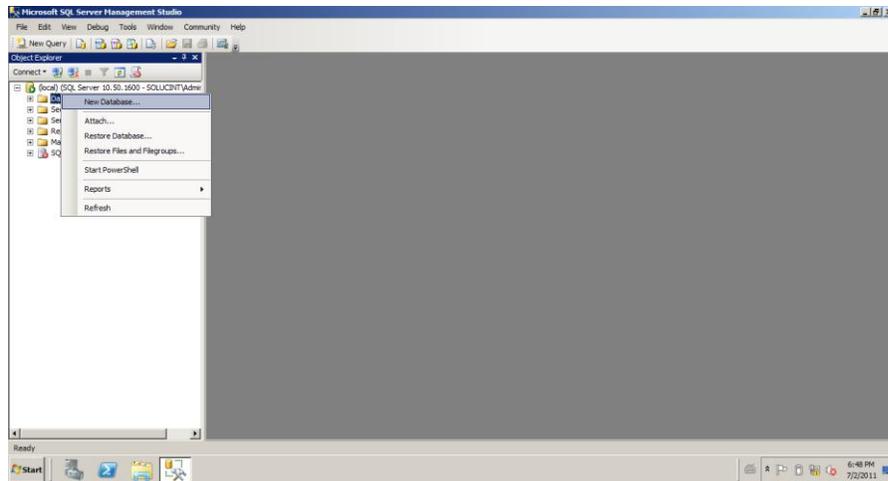


Figura 4.79 Imagen de creación de la base de datos.

Se selecciona la configuración inicial de la base, seguridad, grupos de archivos y opciones, qué se crea para poder subir el script con la estructura y programabilidad de la base de datos. Como se puede indicar en la Figura 4.80.

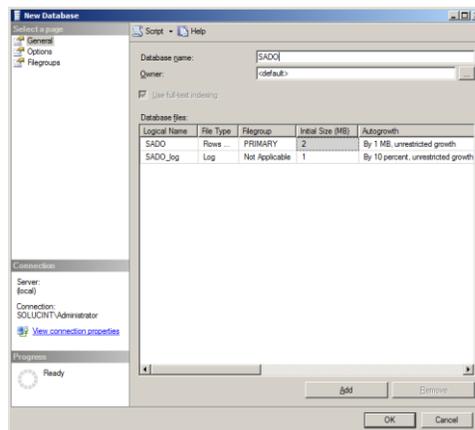


Figura 4.80 Imagen de Opciones de creación de la Base de datos.

Se carga el Script provisto con la estructura y programabilidad de la Base de datos para posteriormente ejecutarlo como se indica en la Figura 4.81.

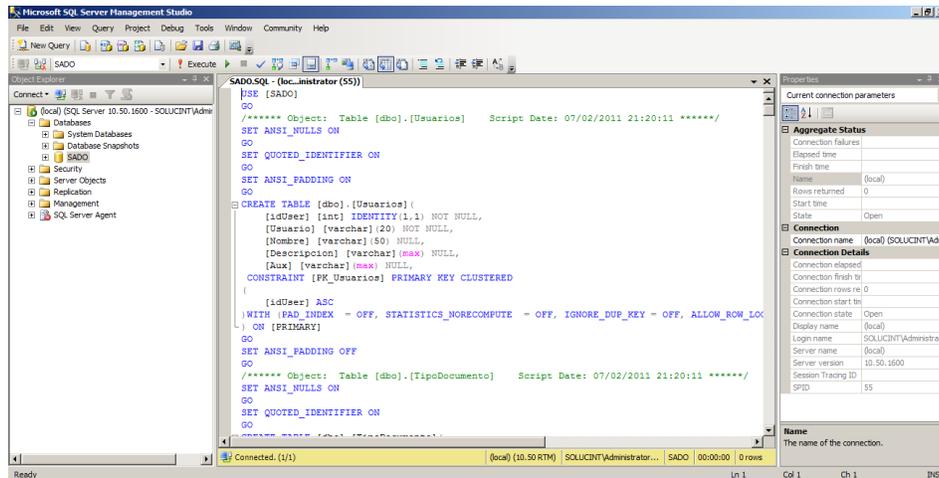


Figura 4.81 Imagen de la vista del Script de estructura y programabilidad de SADO.

Se ejecuta el Script y se debe observar que la ejecución sea limpia, en caso de tener inconvenientes o errores se revisa qué inconvenientes presentó la ejecución del script, en este punto se recomienda desactivar el firewall de Windows para que permita hacer un debug remoto caso contrario no se sabrá si el Script se ejecutó correctamente. Como se indica en la Figura 4.82.

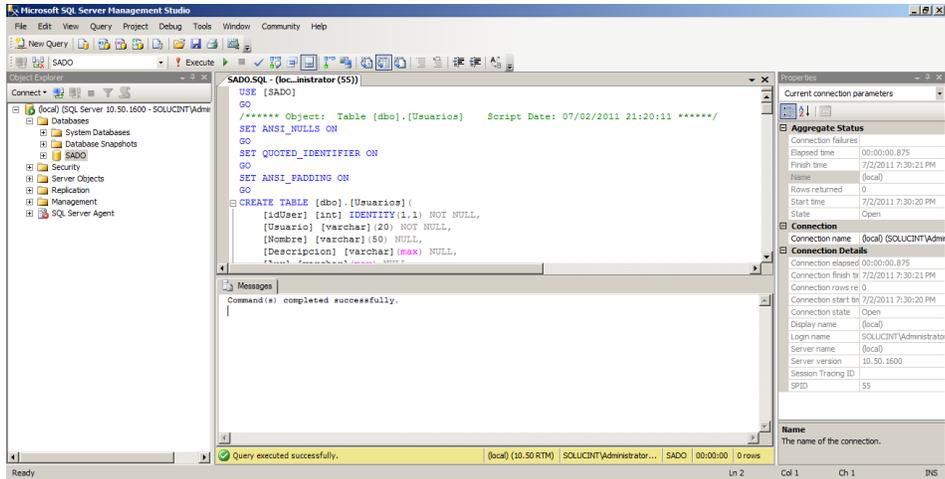


Figura 4.82 Imagen de Ejecución del Script.

Se revisa que la creación de Tablas, Procedimiento haya sido exitosa y se da por finalizada la instalación de la base de datos de SADO. Como se indica en la Figura 4.83.

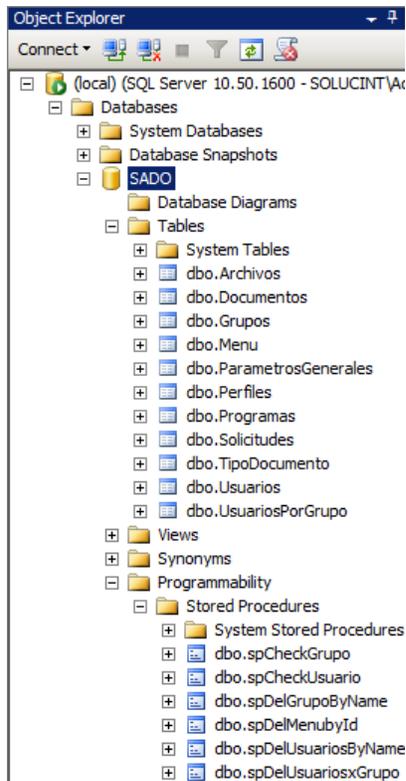


Figura 4.83 Imagen de Tablas de la Base de Datos

4.8 Implementación de SADO

4.8.1 SADO LDAP Migrator

El LDAP Migrator es una utilidad de esta suite que permite migrar los usuarios del dominio y definir quienes pueden tener acceso al programa, simplemente con un par de clic se tiene definido quienes de nuestra organización puede acceder al sistema, pero adicionalmente, a través de la seguridad que brinda las reglas establecidas dentro de este servicio de Active Directory, así se cubre el accesos en tres frentes, software, file server y dominio.

La ventana principal de SADO LDAP Migrator, es simple pero eficaz en lo que se necesita, por un lado se lee mediante LDAP el Directorio Activo, posteriormente se selecciona los usuarios que se desea que ingresen a este sistema, se puede ver sus grupos dentro del dominio y según esto definir quienes ingresan. Como se muestra a continuación en la Figura 4.84.

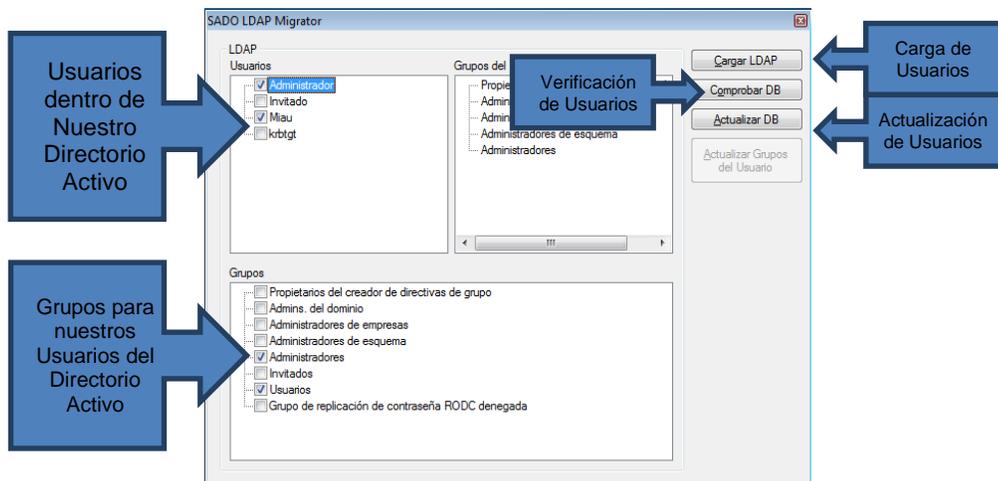


Figura 4.84 Imagen de vista principal del SADO LDAP Migrator

4.8.2 SADO

Al ser una aplicación que va a ser instalada donde los clientes, el logo inicial debe ser configurado según el cliente, por lo que en nuestra aplicación no se muestra nada adicional a la carga de los diferentes módulos (plugins) del sistema. Se muestra a continuación en la Figura 4.85.

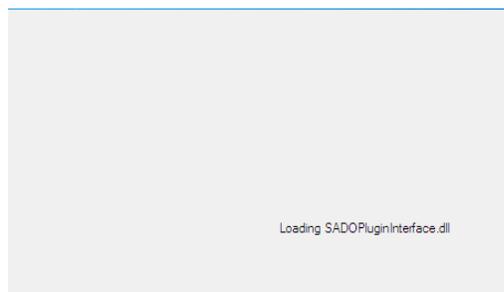


Figura 4.85 Imagen de Inicio del Sistema.

Se recuerda que el sistema está de la mano con el Active Directory, por eso no se tiene la necesidad de una ventana de inicio de sesión dentro del sistema en sí, en

cambio el sistema automáticamente detecta el usuario del cual se accede y valida el ingreso al sistema, así mismo el sistema valida paso por paso el ingreso a la información y posteriormente el acceso a los archivos en el FileServer. Se muestra a continuación en la Figura 4.86.



Figura 4.86 Imagen de la Vista Principal del Sistema.

El sistema con el que se observó la parte de diseño cuenta con 2 niveles de acceso básicos, el principal de tipo Administrador y otro de tipo Usuario pero esto no quiere decir que no se puede crear más perfiles, se pueden desarrollar según las necesidades del cliente.

La vista de administración cuenta con módulos como: la aprobación de documentos, la configuración del sistema y parámetros de la misma, la consulta de documentos y solicitudes de los mismos.

La vista de Cliente solamente cuenta con la consulta de documentos y solicitudes de los mismos.

Se enfocará desde la vista de administrador para ver las características del software, debido a que ciertos módulos son propietarios de Soluciones Inteligentes por lo que no se puede incluir dentro de este documento.

4.8.3 Administración de Documentos

Este módulo como se describió anteriormente en el capítulo III, permite que el administrador apruebe los archivos que se procesaron y estos sean ingresados al sistema como documentos para que su información sea utilizada por los usuarios del sistema.

Entre las Opciones del módulo se tiene como Ingresar los Documentos, Seleccionar el Tipo de Documento, y un borrado de los archivos pendientes del sistema.

Se muestra a continuación en la Figura 4.87.

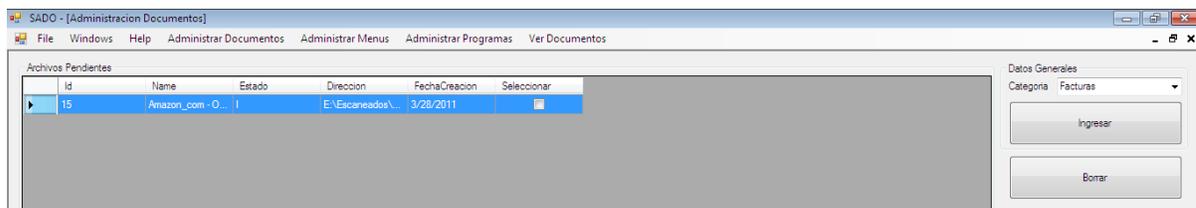


Figura 4.87 Imagen del módulo de Administración de Documentos.

El proceso es sencillo en la versión completa el administrador puede ver los documentos en línea, al momento de la edición de este documento el software es funcional en sus procesos primordiales como aprobarlo documentos uno por uno

pero también existe la posibilidad de aprobarlos en grupos/lotés así mismo su contra parte de borrado de archivos.

4.8.4 Consulta de Documentos

La Consulta de documentos es una vista que permite por una parte buscar documentos para aprovechar la información de los mismos y por otra parte un sub-módulo de solicitudes para poder utilizar dichos documentos, poder imprimirlos, o guardar copias locales de los mismos.

La consulta incluye un sub-módulo de búsqueda de documentos para proceder a crear las solicitudes que posteriormente les permita utilizar los documentos del sistema.

El módulo muestra por una parte los documentos a los cuales se tiene acceso, por otra las solicitudes pendientes de documentos a los cuales el cliente necesita acceso. Se muestra a continuación en la Figura 4.88 - 89.

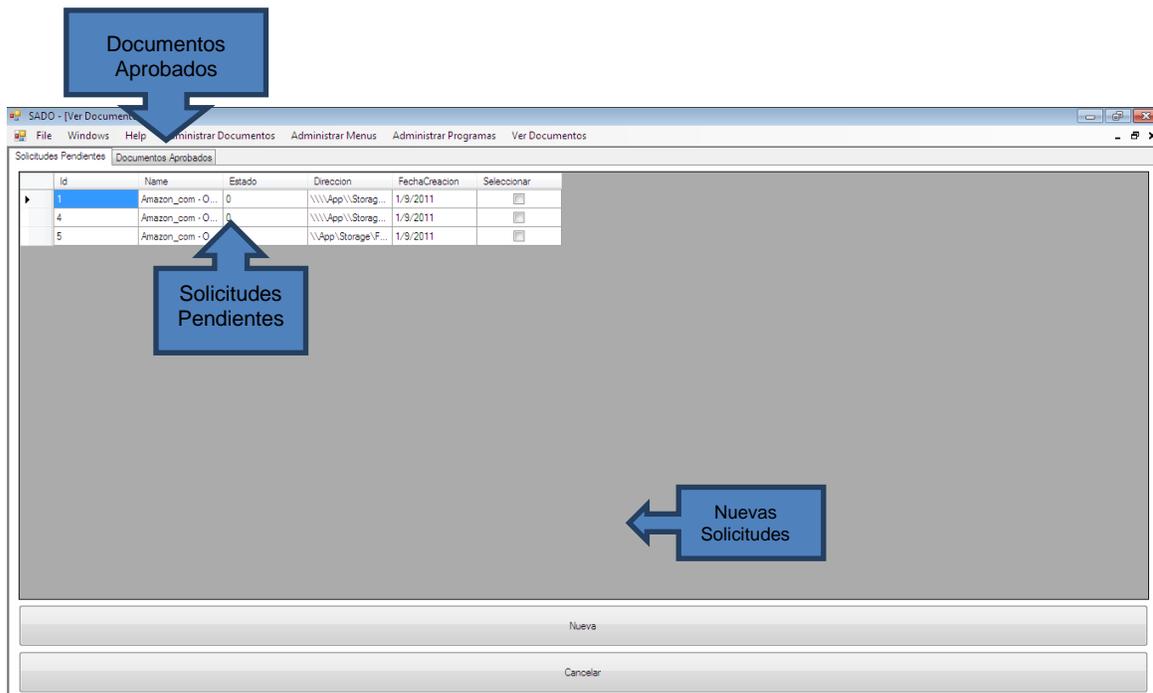


Figura 4.88 Imagen del Módulo de Consulta de documentos.

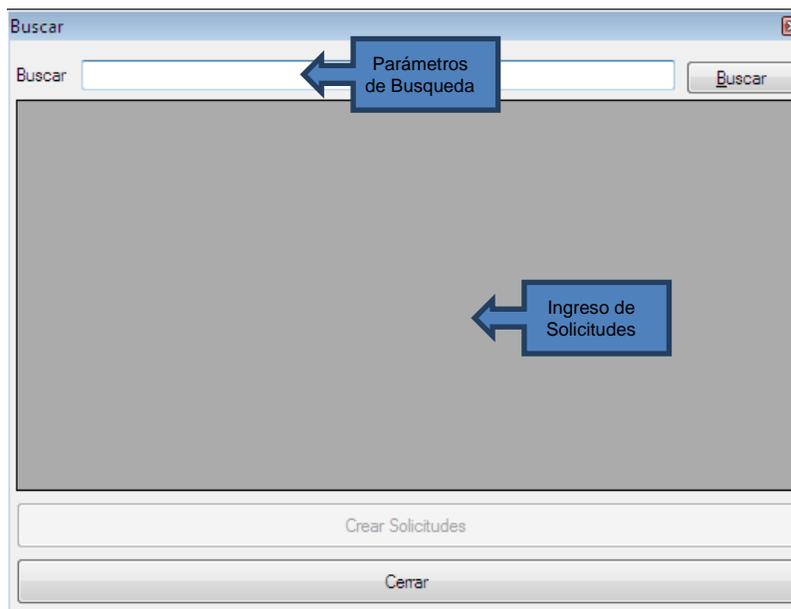


Figura 4.89 Imagen del sub-módulo de búsqueda de documentos.

4.8.5 SADO Vigilante

Este módulo es un servicio de Windows el cual monitorea este almacenamiento en la carpeta que se define donde se depositarán los archivos que posteriormente se aprobarán por el administrador para convertirse en documentos y subirlos al sistema.

Dicho modulo debe Funcionar con una cuenta de usuario que tenga acceso al sistema y al almacenamiento para lo cual es recomendable se cree una cuenta de servicio con acceso a dichos recursos dentro del dominio.

SADO no tiene una Interface de Configuración solamente sus parámetros son editables mediante el documento de configuración de la aplicación. Se muestra a continuación en la Figura 4.90.

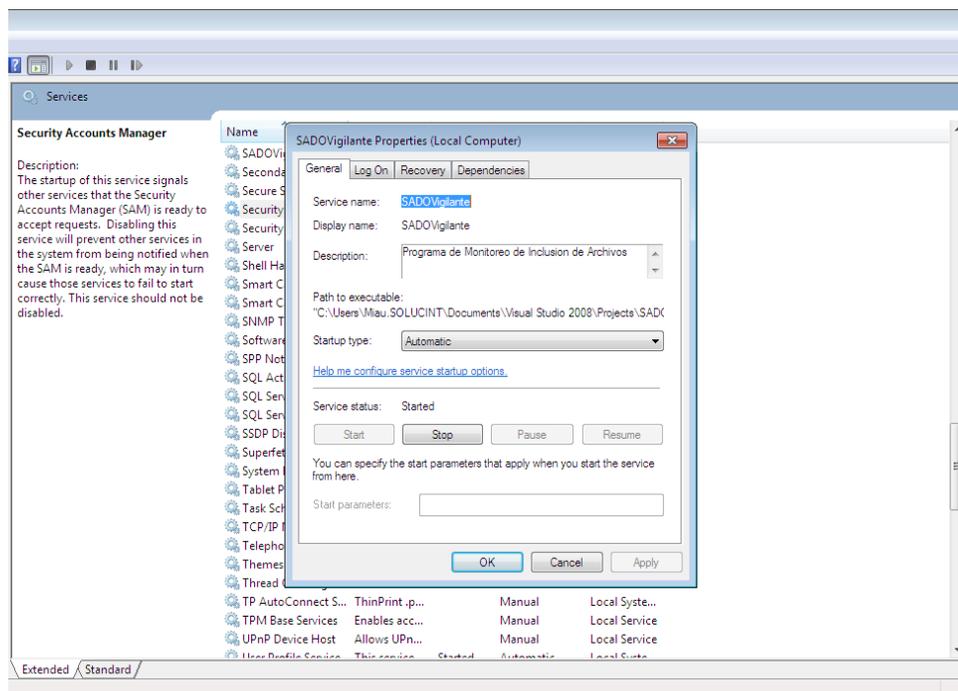


Figura 4.90 Imagen de Vista del Servicio SADO Vigilante.

El Proceso SADO a continuación una Figura 4.91 de pasos para utilizar el sistema SADO.

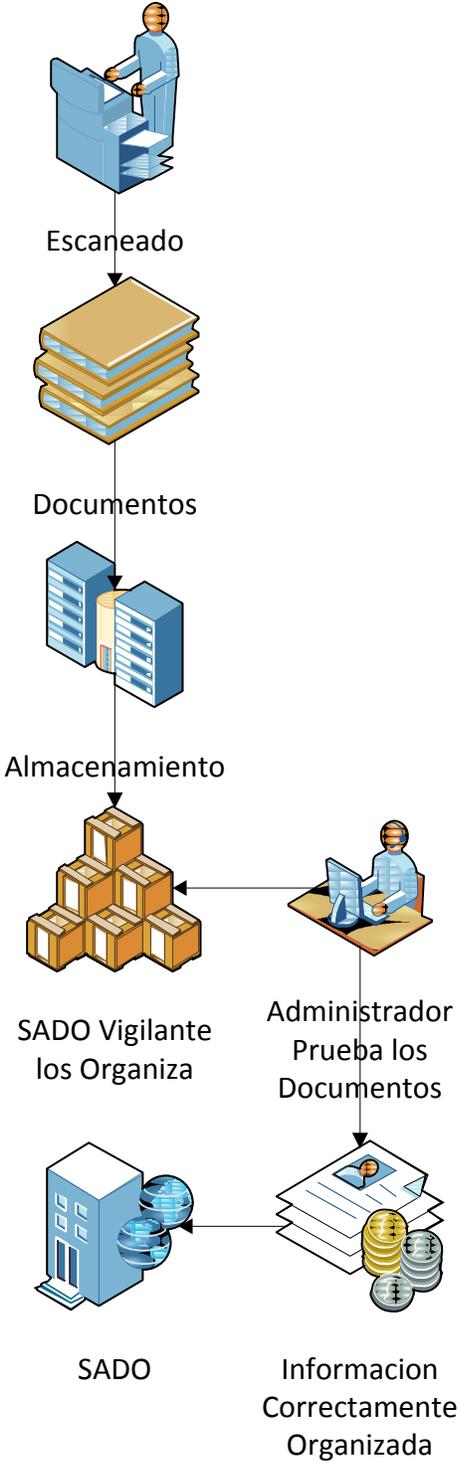


Figura 4.91 Proceso SADO

Las Solicitudes de Documentos a continuación una Figura 4.92 de pasos para utilizar el sistema SADO.

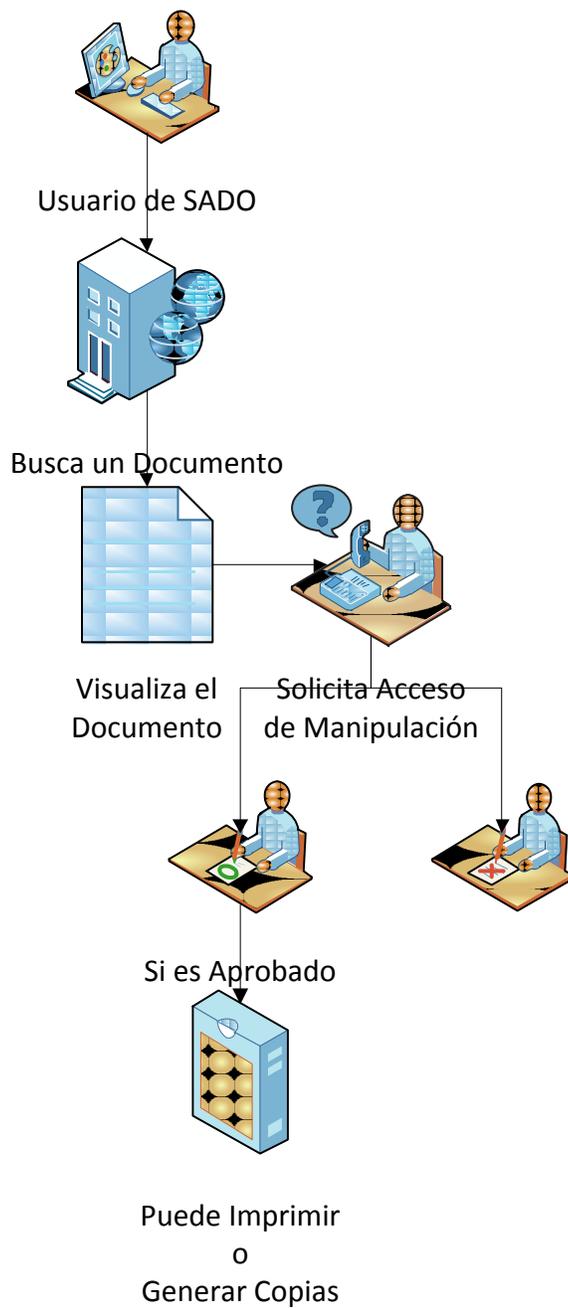


Figura 4.92 Proceso SADO

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- En base a los requerimientos presentados por parte de Soluciones Inteligentes se diseñó el plan de desarrollo de la aplicación SADO con el objetivo de cumplir las expectativas del cliente y sus futuras implementaciones.
- Con respecto a la parte investigativa de los autores se concluye que para alcanzar el objetivo propuesto, fue necesario adquirir nuevos conocimientos en cuanto a LDAP, Adobe Capture debido a que las aplicaciones en la vida profesional cubren una gama mucho más amplia de componentes, interfaces, lenguajes, artificios de desarrollo, que los aprendidos en la universidad.
- Para poder implementar el Módulo Vigilancia con Adobe Capture fue necesario tener el conocimiento de la Arquitectura de Adobe Capture con el fin de que SADO Vigilante sea un servicio complementario en la construcción de aplicación.

- Después de la evaluación de “Soluciones Inteligentes” a través del análisis FODA se concluyó que para alcanzar una de sus oportunidades el sistema SADO es el futuro aplicativo estrella.
- El diseño inicial de interfaces del estándar modular fue la forma más correcta de desarrollo aplicado. Su escalabilidad es única, sus librerías particulares serán aplicadas dentro de Soluciones Inteligentes como el nuevo Estándar de Desarrollo.
- Las pruebas de contenido, función, estructura, facilidad de uso, navegabilidad, y desempeño; ayudaron a detectar y corregir los errores antes de la puesta en producción del producto SADO para Soluciones Inteligentes.

5.2 Recomendaciones

- Realizar una instalación y configuración adecuada utilizando las explicaciones expresadas en este documento, en el cual se forma un precedente dichas configuraciones para el correcto funcionamiento de la aplicación.
- Soluciones Inteligentes debe plantear un esquema de implementación que incluya un periodo de pruebas lo suficientemente extenso para que sus clientes

puedan comprender el uso y se aprecien si existen problemas con la implementación/configuración de la misma.

➤ El administrador del Sistema de parte del cliente debe contar con conocimientos intermedios de administración de Servidores Windows, y así mismo debe contar con conocimientos básicos de Administración de Servidores SQL.

➤ Que la Universidad masifique o intensifique las células de desarrollo en las diferentes arquitecturas existentes, así mismo como que a los profesores se les motive a que en sus clases no se focalicen a una herramienta en especial y se enfoquen más a los conceptos de las asignaturas.

➤ Involucrar a los alumnos en el desarrollo de aplicativos complejos desde los primeros niveles para fomentar la investigación y la habilidad de crear arquitecturas propias.

BIBLIOGRAFÍA

Libros:

- PRESSMAN Roger, Ingeniería de Software Un Enfoque Práctico, Mc. Graw Hill, Madrid – España, 2002.

- LAMARCA María Jesús, Hipermedia/Multimedia. Universidad Complutense de Madrid – 2006.

- SAZ TRAMILLAS Jesús, PICAZO GARRIDO Piedad; Software Libre para Servicios de Información Digital, Pearson Prentice Hall

- GARCÍA RUBIO Ramón, QUIRÓS SUÁREZ Javier, GONZÁLEZ Santiago Martín, SANTOS GALLEGO Ramón, FERNÁNDEZ MORÁN Samuel; Diseño Gráfico de Contenidos para Internet, Pearson Prentice Hall

- NARAMORE Elizabeth, GERGER Jason, LE SCOUARNEC Yann, STOLZ Jeremy, K. Michael; Beginning PHP5, Apache, MySQL, Web Development; Glass Wrox

Web:

- Wikipedia, LDAP , Actualización 27 julio 2011;
<http://es.wikipedia.org/wiki/LDAP>[Último Acceso, 18 Julio 2010].

- Microsoft, 2010-2011, SQL Server;
<http://www.microsoft.com/sqlserver/en/us/default.aspx>[Último Acceso, 17 Junio 2010].

- Wikipedia ,Active Directory, Actualización 22 agosto 2011;
http://es.wikipedia.org/wiki/Active_directory[Último Acceso, 4 Mayo 2010].

- Wikipedia , OOHDM, Actualización 12 agosto 2010;
<http://es.wikipedia.org/wiki/OOHDM>[Último Acceso, 17 Enero 2010].

- Fabián Tang, 21 Diciembre del 2005, Referencias de Interfaces de Sistemas MDI; <http://www.codeproject.com/KB/bugs/MgdMDIClientActivation.aspx>
<http://www.codeproject.com/KB/cs/MDITabBrowsing.aspx> [Último Acceso, 5 Septiembre 2010].

- Microsoft, 2009-2010, Librería de Referencia de .Net;
<http://msdn.microsoft.com/es-ec/library/>[Último Acceso, 18 Junio 2010].

- Adobe Systems Incorporated, 14 Julio 2009 , Adobe Capture;
<http://www.adobe.com/products/acrcapture/>[Último Acceso, 17 Julio 2011].

- Gustavo Mendoza Lemus, 30 Abril 2007, Digitalización de Documentos
Posibilidad de Integrar Historia de Todos los Países
<http://archivologo.blogcindario.com/2007/04/00999-digitalizacion-de-documentos-posibilidad-de-integrar-historia-de-todos-los-paises.html> [Último Acceso, 22 Diciembre 2010].

- Ian Burley, 11 Enero 2011 2009 , Adobe Capture;
<http://www.pcpro.co.uk/reviews/graphics-design/2176/adobe-acrobat-capture-3-personal-editio.html> [Último Acceso, 17 Julio 2011].

GLOSARIO DE TÉRMINOS

Actor: Es un objeto que se encuentran fuera del sistema a modelar. Representan entes que tienen necesidad de intercambiar información con el sistema; pueden ser instanciados por usuarios, dispositivos u otros sistemas.

Base de Datos (*Database*): Conjunto de información organizada que contiene una colección de registros donde un sistema puede buscar, reescribir, clasificar, borrar, añadir información a la misma.

Caso de Uso: Descripción a detalle de las actividades y procesos necesarios para el desarrollo de un sistema o aplicación.

LDAP: Las siglas de Lightweight Directory Access Protocol (en español Protocolo Ligero de Acceso a Directorios) que hacen referencia a un protocolo a nivel de aplicación el cual permite el acceso a un servicio de directorio y distribuido para buscar diversa información en un entorno de red

OCR: El Reconocimiento Óptico de Caracteres. Identifican automáticamente símbolos o caracteres que pertenecen a un determinado alfabeto, a partir de una imagen para almacenarla en forma de datos con los que podremos interactuar mediante un programa de edición de texto o similar.

Kerberos: Es un protocolo de autenticación de redes de ordenador que permite a dos computadores en una red insegura demostrar su identidad mutuamente de manera segura.

ANEXOS

ANEXO A

A. Diagramas De Secuencia Por Actores

ACTOR: ADMINISTRADOR

1. CU-ADM-SADO-01: Gestionar Usuario

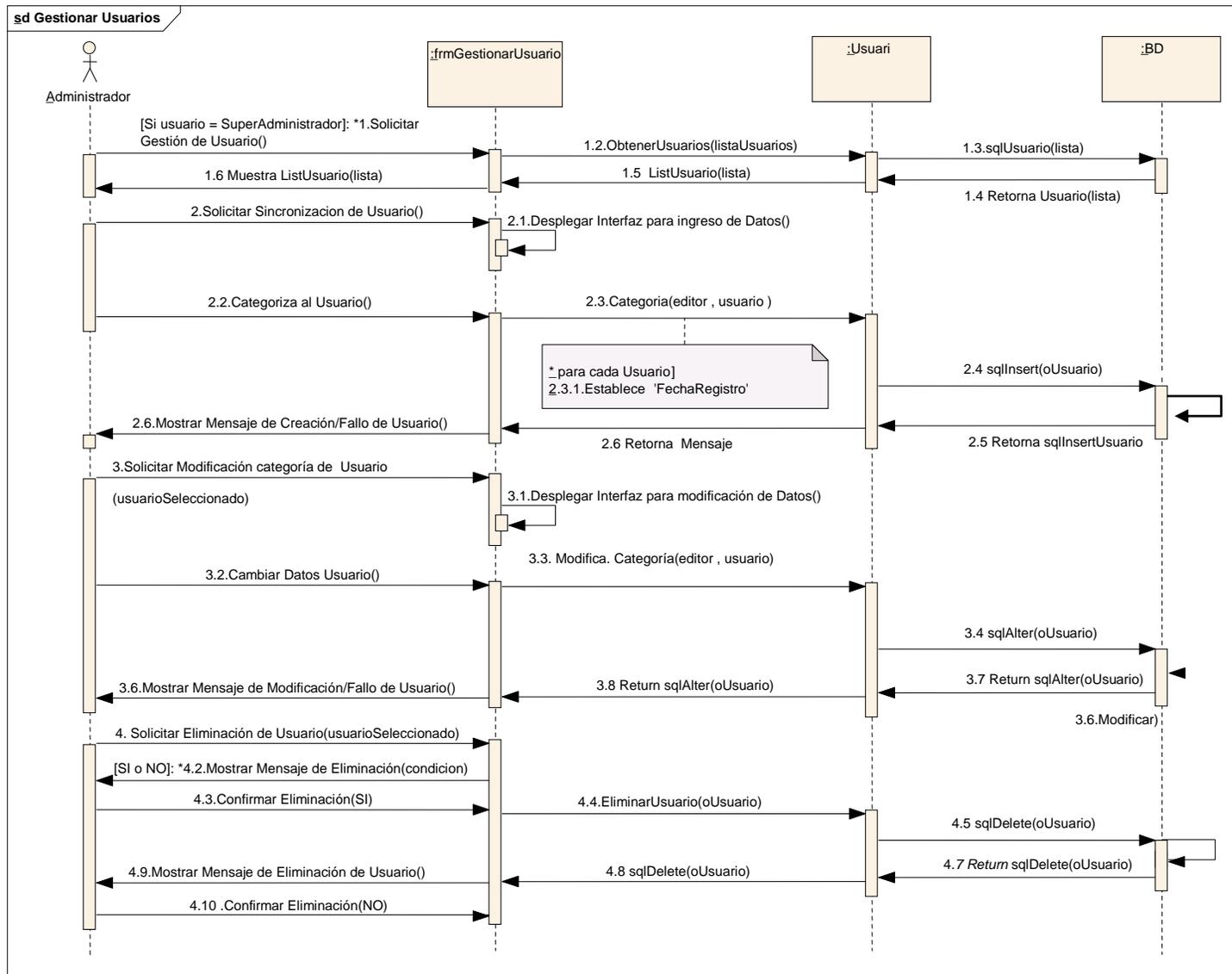


Figura A.1: Diagrama de Secuencia – Gestionar de Usuarios

2. CU-ADM- SADO -02: Gestionar Administración de Documentos

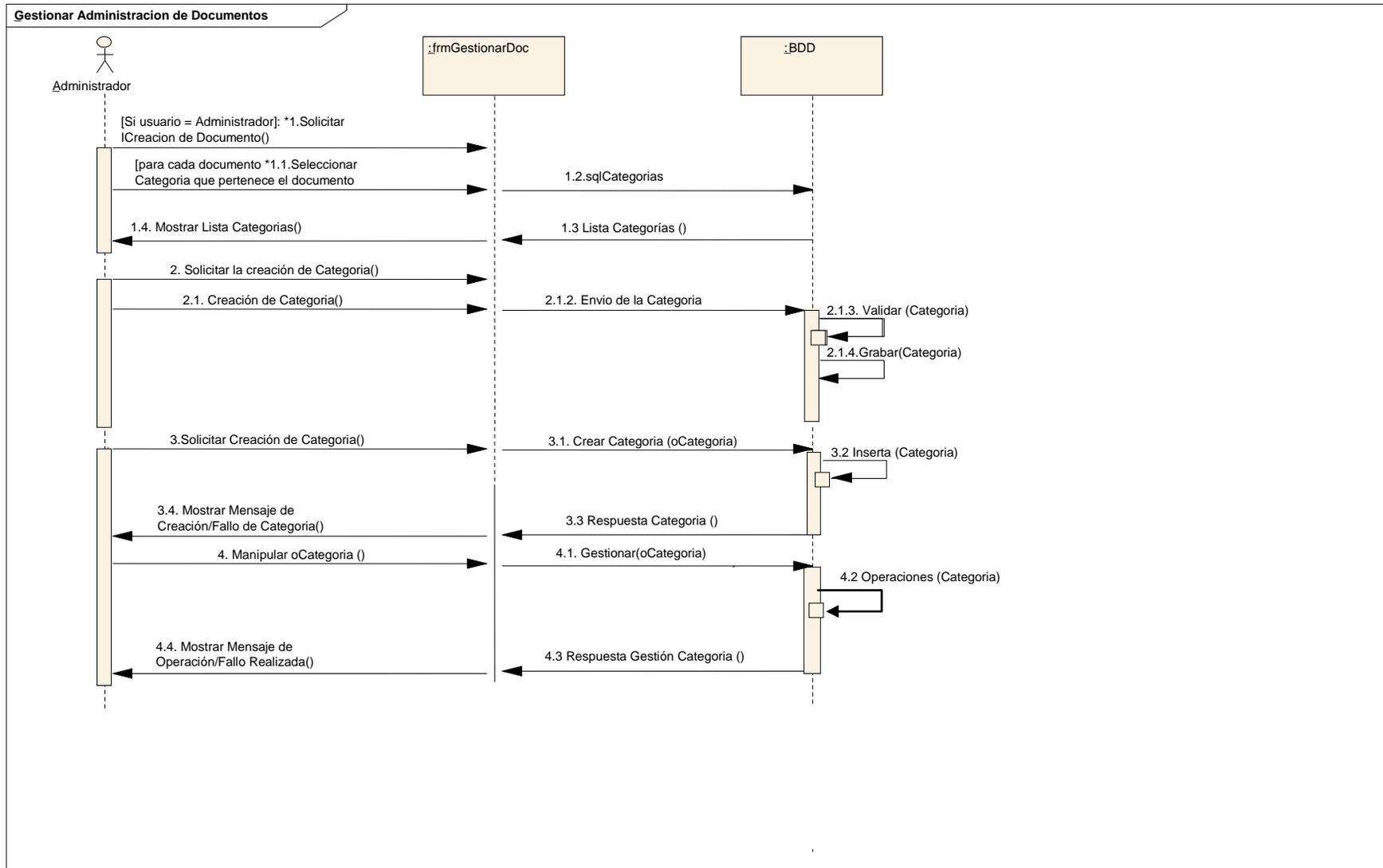


Figura A.3: Diagrama de Secuencia – Gestiona Administración de Documentos

3. CU-ADM- SADO -04: Gestionar Histórico de Documentos

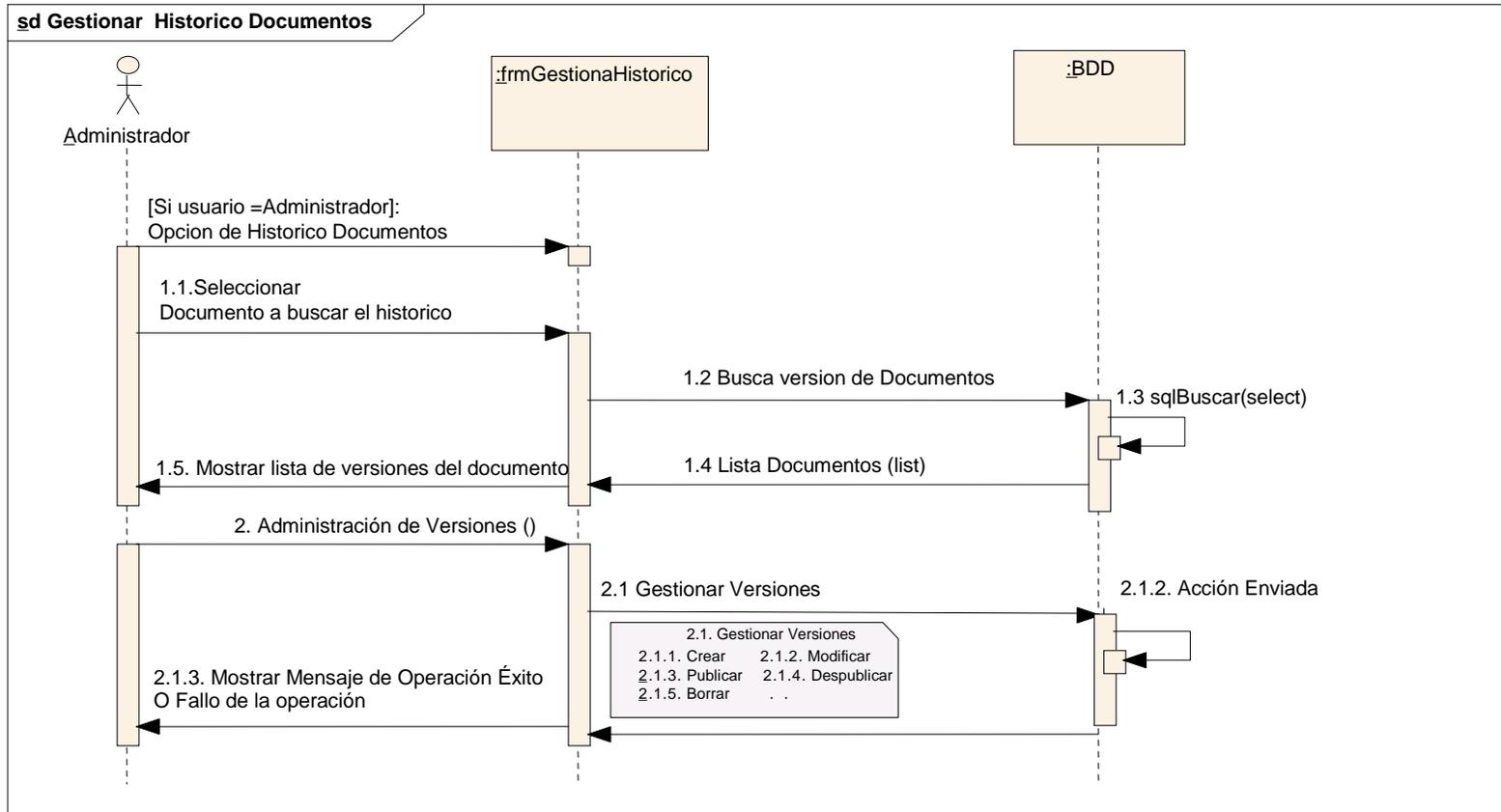


Figura A.4: Diagrama de Secuencia – Gestionar Histórico de Documentos

ACTOR: EDITOR

1. CU-EDI-SADO -01: Utilizar Servicio de Históricos

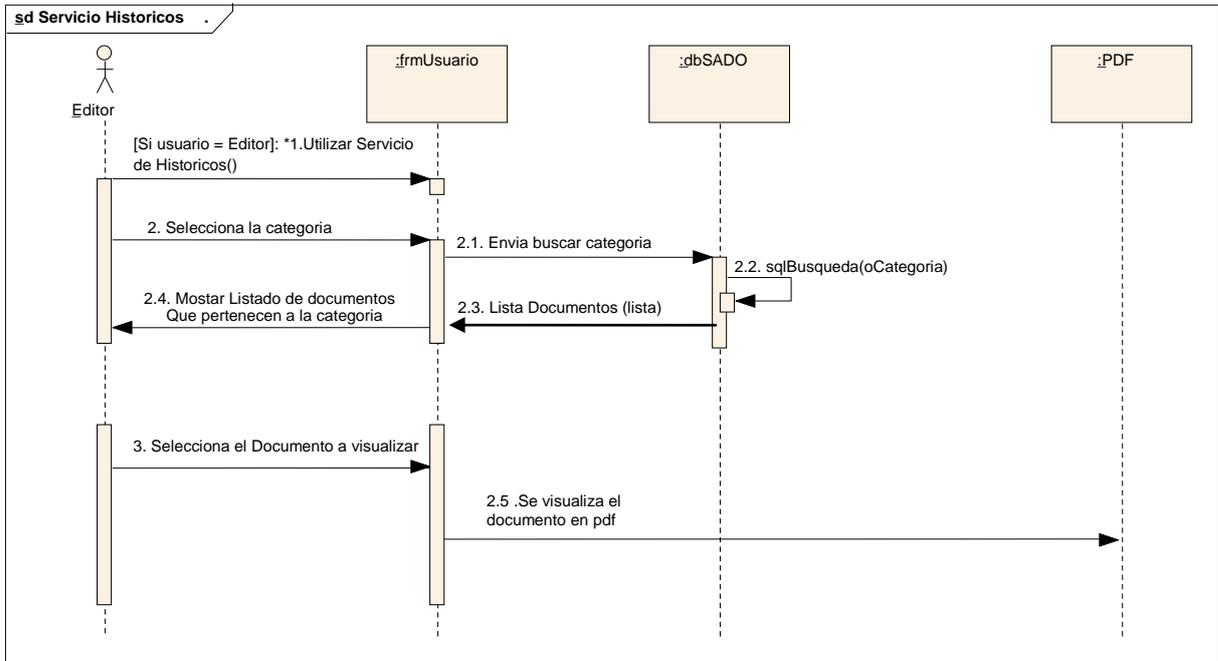


Figura A.11: Diagrama de Secuencia – Servicio Histórico de Documentos

2. SEC-EDI-SADO-02: Utilizar servicio de Consultas

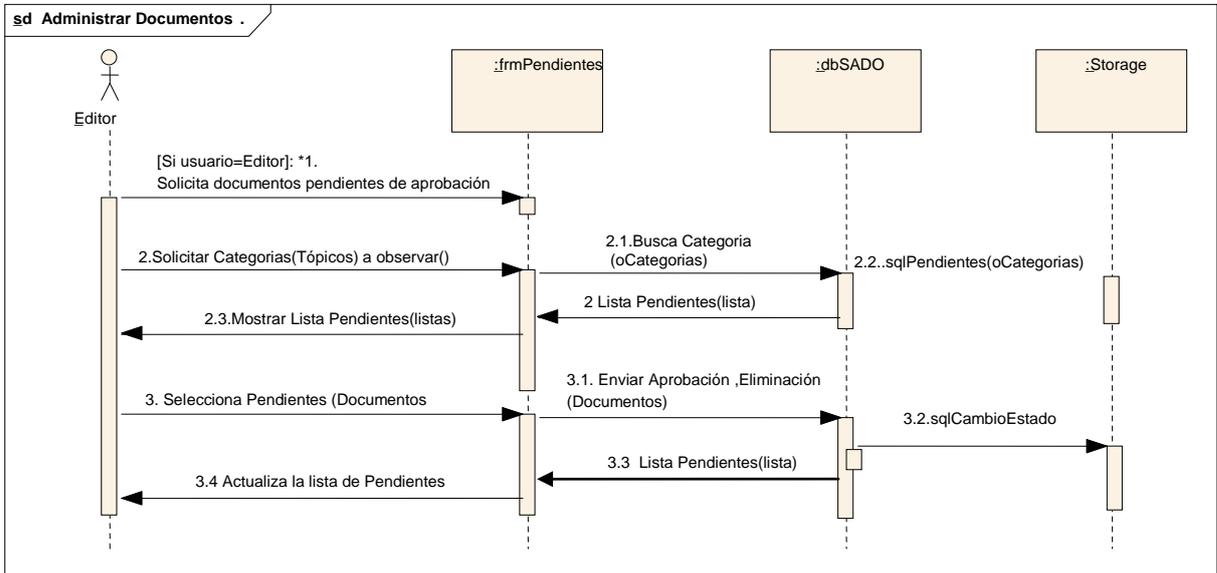


Figura A.12: Diagrama de Secuencia – Administrar Documentos

ACTOR: USUARIO

1. SEC-USU-SADO-01: Servicio de Histórico de Documentos

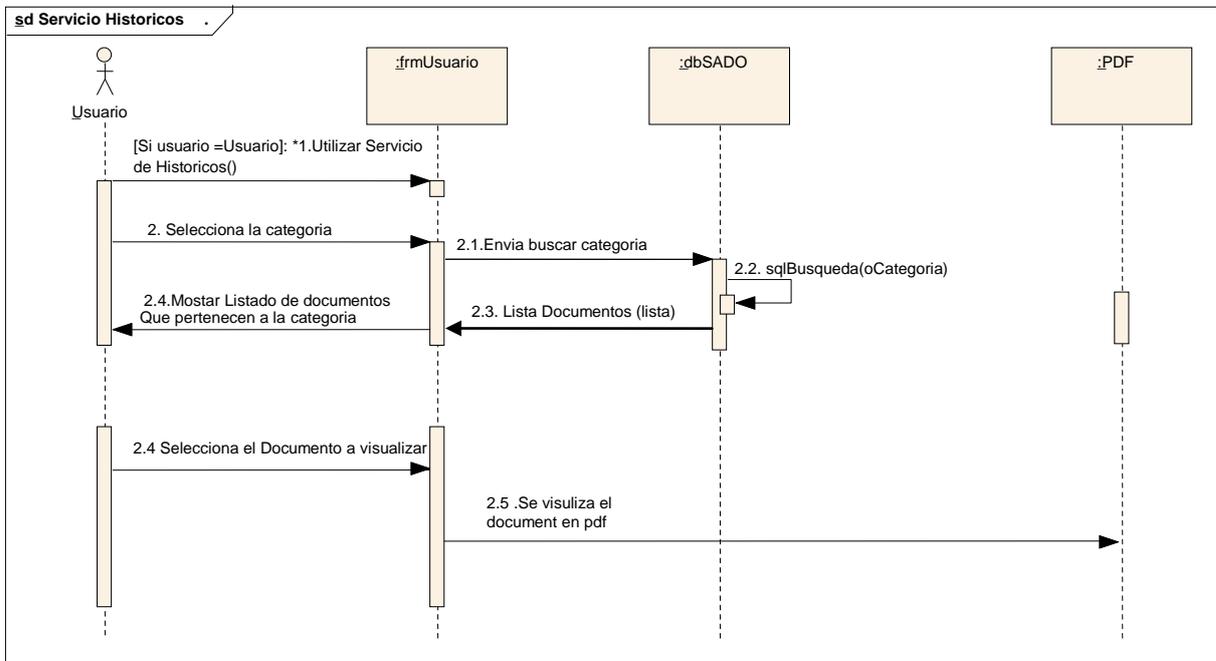


Figura A.12: Diagrama de Secuencia – Histórico de Documentos