

CONTENIDO

CONTENIDO	I
ÍNDICE DE FIGURAS	IV
ÍNDICE DE TABLAS	VII
RESUMEN	VIII
PRESENTACIÓN	XI
CAPÍTULO 1. SOLUCIONES PERSONALIZADAS DE MICROSOFT OFFICE 2003	1-1
1.1 MICROSOFT OFFICE 2003 Y LAS SOLUCIONES PERSONALIZADAS	1-1
1.1.1 INTRODUCCIÓN	1-1
1.1.2 DESARROLLO DE SOLUCIONES BASADAS EN LAS APLICACIONES DE OFFICE.....	1-2
1.1.3 IMPORTANCIA DEL DESARROLLO DE LAS SOLUCIONES PERSONALIZADAS DE OFFICE	1-2
1.1.4 NOVEDADES PARA LOS DESARROLLADORES EN EL OFFICE 2003	1-3
1.1.4.1 Manejo de Datos con XML	1-3
1.1.4.2 Consumo de Servicios Web XML.....	1-3
1.1.4.3 Desarrollo de Aplicaciones Inteligentes	1-3
1.2 DESARROLLO CON ALTERNATIVAS QUE UTILIZAN CÓDIGO NO ADMINISTRADO.....	1-4
1.2.1 COMPONENT OBJECT MODEL	1-4
1.2.2 EL MODELO DE OBJETOS DE OFFICE.....	1-5
1.2.3 GENERALIDADES DE VISUAL BASIC PARA APLICACIONES.....	1-6
1.2.3.1 El Desarrollo de Aplicaciones de Office Utilizando VBA	1-6
1.2.4 LOS COMPLEMENTOS EN LAS APLICACIONES DE OFFICE.....	1-7
1.2.4.1 Complementos de Aplicación Específica	1-7
1.2.4.2 Complementos COM.....	1-7
1.3 GENERALIDADES DE LA TECNOLOGÍA .NET Y EL CÓDIGO ADMINISTRADO.....	1-8
1.3.1 ELEMENTOS PRINCIPALES DEL FRAMEWORK .NET.....	1-8
1.3.1.1 Common Language Runtime.....	1-8
1.3.1.2 Clases de Programación Unificadas	1-8
1.3.2 EL CÓDIGO ADMINISTRADO EN LAS SOLUCIONES PERSONALIZADAS DE OFFICE	1-9
1.3.2.1 Interoperabilidad COM	1-9
1.3.2.2 Ausencia de los Tipos de Datos Variant.....	1-9
1.3.2.3 Eliminación de Elementos no Utilizados.....	1-9
1.3.2.4 La Compilación JIT.....	1-10
1.4 INTERACCIÓN ENTRE EL CÓDIGO ADMINISTRADO Y EL NO ADMINISTRADO.....	1-10
1.4.1 DIFERENCIAS ENTRE COM Y .NET	1-10
1.4.2 OBJETIVO DE LA INTEROPERABILIDAD	1-11
1.4.3 ENSAMBLADOS DE INTEROPERABILIDAD	1-12
1.4.3.1 Ensamblados de Interoperabilidad Primaria	1-13
1.4.3.2 Los PIAs de Office 2003	1-14
1.4.4 LOS CONTENEDORES EN TIEMPO DE EJECUCIÓN	1-15
1.4.4.1 Tipos de Contenedores	1-15
1.4.4.2 Relación Servidor – Contenedor – Cliente	1-16
1.4.4.3 Los Contenedores y las Interfaces	1-17
1.4.4.4 Funciones Principales de los Contenedores.....	1-18
1.4.4.5 Naturaleza de los Contenedores	1-19
CAPÍTULO 2. DESARROLLO DE FORMULARIOS, ETIQUETAS Y DOCUMENTOS INTELIGENTES PARA MICROSOFT OFFICE 2003 UTILIZANDO .NET Y XML	2-20
2.1 XML EN LAS APLICACIONES DE OFFICE	2-21
2.1.1 CARACTERÍSTICAS PRINCIPALES DE UN ARCHIVO XML.....	2-21
2.1.2 TIPOS DE ARCHIVOS XML	2-22

2.1.3	IMPORTANCIA DEL XML EN LAS APLICACIONES DE OFFICE	2-22
2.1.4	LOS ESQUEMAS XML EN EL OFFICE 2003	2-23
2.1.5	LOS FORMATOS DE ARCHIVO XML EN EL OFFICE 2003	2-23
2.1.6	LOS DESARROLLADORES, EL XML Y EL OFFICE 2003	2-24
2.2	DESARROLLO DE SOLUCIONES UTILIZANDO XML Y WORD 2003	2-25
2.2.1	TRABAJO CON ESQUEMAS XML	2-26
2.2.1.1	Biblioteca de Esquemas XML	2-26
2.2.1.2	Panel de Tareas - Estructura XML	2-28
2.2.2	XML Y EL MODELO DE OBJETOS DE WORD 2003	2-29
2.3	DESARROLLO DE SOLUCIONES UTILIZANDO INFOPATH 2003	2-30
2.3.1	CARACTERÍSTICAS PRINCIPALES DE INFOPATH	2-30
2.3.2	CARACTERÍSTICAS DE LOS FORMULARIOS DE INFOPATH	2-31
2.3.3	DISEÑO DE FORMULARIOS DE INFOPATH	2-32
2.3.4	INFOPATH Y EL .NET 2003	2-33
2.4	UTILIZACIÓN DE SERVICIOS WEB XML DENTRO DEL OFFICE 2003	2-35
2.4.1	PROTOCOLOS Y ESTÁNDARES ASOCIADOS CON LOS SERVICIOS WEB XML	2-35
2.5	VISIÓN GENERAL DE ETIQUETAS Y DOCUMENTOS INTELIGENTES	2-37
2.5.1	ETIQUETAS INTELIGENTES	2-37
2.5.1.1	Características Principales de la Tecnología de Etiquetas Inteligentes	2-38
2.5.2	DOCUMENTOS INTELIGENTES	2-39
2.5.2.1	Características Principales de la Tecnología de Documentos Inteligentes	2-40
2.5.2.2	Elementos característicos de los Documentos Inteligentes de Microsoft	2-41
2.6	DESARROLLO DE ETIQUETAS Y DOCUMENTOS INTELIGENTES	2-42
2.6.1	DESARROLLO DE ETIQUETAS INTELIGENTES	2-42
2.6.1.1	El Microsoft Office Smart Tag List para Etiquetas Inteligentes	2-43
2.6.1.2	Desarrollo de Etiquetas Inteligentes utilizando Código	2-43
2.6.2	DESARROLLO DE DOCUMENTOS INTELIGENTES	2-44
2.6.2.1	El Microsoft Office Smart Tag List para Documentos Inteligentes	2-45
2.6.2.2	Desarrollo de Documentos Inteligentes utilizando Código	2-46
2.7	HERRAMIENTAS PARA EL DESARROLLO	2-48
2.7.1	KIT DE HERRAMIENTAS DE INFOPATH 2003 PARA VISUAL STUDIO .NET	2-48
2.7.2	HERRAMIENTAS DE VISUAL STUDIO PARA MICROSOFT OFFICE 2003	2-50
2.7.3	VERTIGO MANAGED SMART DOCUMENTS WRAPPER	2-50
2.8	MECANISMOS DE SEGURIDAD Y PRIVACIDAD	2-52
2.8.1	SEGURIDAD DE OFFICE 2003 A NIVEL DE DOCUMENTO	2-53
2.8.2	SEGURIDAD DE LAS APLICACIONES PERSONALIZADAS DE OFFICE 2003 QUE UTILIZAN CÓDIGO ADMINISTRADO	2-54
2.8.3	APLICACIONES PERSONALIZADAS DE OFFICE 2003 QUE UTILIZAN EL VISUAL STUDIO TOOLS FOR OFFICE LOADER	2-56
2.8.4	SEGURIDAD EN LOS DOCUMENTOS DE OFFICE 2003 QUE UTILIZAN XML	2-58
CAPÍTULO 3. DISEÑO E IMPLEMENTACIÓN DE UNA SOLUCIÓN PERSONALIZADA DE MICROSOFT OFFICE 2003 CON .NET Y XML		3-60
3.1	ANÁLISIS DEL PROBLEMA, SITUACIÓN ACTUAL Y REQUERIMIENTOS	3-61
3.1.1	ANÁLISIS DEL PROBLEMA	3-61
3.1.2	SITUACIÓN ACTUAL	3-62
3.1.2.1	Proceso de Aprobación	3-63
3.1.2.2	Proceso Posterior a la Aprobación	3-65
3.1.3	REQUERIMIENTOS	3-66
3.1.3.1	Funcionalidad	3-67
3.1.3.2	Conectividad	3-72
3.1.3.3	Seguridad	3-73

3.1.3.4 Casos de Uso	3-75
3.2 DISEÑO	3-82
3.2.1 ARQUITECTURA	3-83
3.2.1.1 Elementos Principales e Interconexión Física	3-83
3.2.1.2 Interconexión Lógica.....	3-85
3.2.1.3 Esquema Detallado de la Arquitectura	3-87
3.2.2 FUENTES DE DATOS.....	3-90
3.2.2.1 Base de Datos	3-90
3.2.2.2 Archivos de Datos	3-113
3.2.3 ESQUEMAS XML DE LA SPO.....	3-118
3.2.3.1 Esquema XML para el Documento Inteligente	3-118
3.2.3.2 Esquema XML para el Formulario de InfoPath.....	3-124
3.2.4 CAPAS LÓGICAS.....	3-128
3.2.4.1 Interfaz de Usuario	3-129
3.2.4.2 Lógica de Negocio	3-129
3.2.4.3 Acceso a Datos y Servicios	3-130
3.3 IMPLEMENTACIÓN.....	3-131
3.3.1 IMPLEMENTACIÓN DE LAS CÁPAS LÓGICAS.....	3-131
3.3.1.1 Interfaz de Usuario	3-131
3.3.1.2 Lógica de Negocio	3-152
3.3.1.3 Acceso a Datos y Servicios	3-158
3.3.1.4 Consideraciones para la Escritura del Código Fuente	3-163
3.3.2 CONSIDERACIONES PARA EL DOCUMENTO INTELIGENTE.....	3-166
3.3.2.1 Archivo de Manifiesto XML	3-166
3.3.2.2 Directorio del Archivo de Manifiesto.....	3-170
3.3.2.3 Directorio de Instalación	3-171
3.3.2.4 Consideraciones de Seguridad.....	3-172
3.3.2.5 Instalación	3-176
3.3.3 CONSIDERACIONES PARA EL FORMULARIO DE INFOPATH.....	3-178
3.3.4 REQUERIMIENTOS DE SOFTWARE	3-182
3.4 PRUEBAS.....	3-183
3.4.1 PRUEBAS DE SISTEMA	3-184
3.4.1.1 Administración de los PPTs	3-184
3.4.1.2 Administración del Estado de los PPTs Y PTs.....	3-191
3.4.1.3 Elaboración de Cartas y Memorandos	3-194
3.4.1.4 Visualización de Reportes	3-199
3.4.2 PRUEBAS DE ACEPTACIÓN.....	3-201
 CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES.....	 4-206
 ANEXO A.....	 221
ANEXO B.....	228
ANEXO C.....	231
ANEXO D.....	233
ANEXO E.....	235
ANEXO F	239
ANEXO G	256
ANEXO H	282

ÍNDICE DE FIGURAS

ILUSTRACIÓN 1-1 CONTENEDORES EN TIEMPO DE EJECUCIÓN DE OBJETOS COM (RCW) O .NET (CCW).	1-16
ILUSTRACIÓN 1-2 REPRESENTACIÓN DE LA RELACIÓN UNÍVOCA ENTRE CONTENEDOR Y OBJETO.	1-17
ILUSTRACIÓN 1-3 INTERFACES COM Y EL RCW.	1-17
ILUSTRACIÓN 2-1 CUADRO DE DIÁLOGO – “BIBLIOTECA DE ESQUEMAS”	2-27
ILUSTRACIÓN 2-2 PANEL DE TAREAS – “ESTRUCTURA XML”	2-28
ILUSTRACIÓN 2-3 PANELES DE TAREAS PARA EL DISEÑO DE FORMULARIOS EN INFOPATH.	2-32
ILUSTRACIÓN 2-4 SECUENCIA DE PASOS QUE SIGUE UNA APLICACIÓN ANFITRIÓN CUANDO CARGA UNA DLL QUE DEFINE UN DOCUMENTO INTELIGENTE.	2-47
ILUSTRACIÓN 2-5 ARCHIVOS DEL PAQUETE DE EXPANSIÓN DE UNA SOLUCIÓN CREADA CON <i>VERTIGO MANAGED SMART DOCUMENTS WRAPPER</i> .	2-51
ILUSTRACIÓN 2-6 ARQUITECTURA DE SOLUCIONES DE OFFICE CON CÓDIGO ADMINISTRADO.	2-56
ILUSTRACIÓN 2-7 PARTES FUNDAMENTALES DEL PROCESO DE CARGA DE CÓDIGO UTILIZANDO <i>VISUAL STUDIO TOOLS FOR OFFICE LOADER</i> .	2-57
ILUSTRACIÓN 2-8 CHEQUEOS DE SEGURIDAD PARA LOS PAQUETES DE EXPANSIÓN XML.	2-59
ILUSTRACIÓN 3-1 DIAGRAMA DE ESTADOS PPTS Y PTS.	3-70
ILUSTRACIÓN 3-2 DIAGRAMA DE CASOS DE USO.	3-75
ILUSTRACIÓN 3-3 ELEMENTOS FUNDAMENTALES DE LA SPO.	3-85
ILUSTRACIÓN 3-4 EJEMPLOS DE INTERCONEXIÓN LÓGICA.	3-86
ILUSTRACIÓN 3-5 ARQUITECTURA DETALLADA DE LA SPO.	3-88
ILUSTRACIÓN 3-6 DIAGRAMA DE CONTEXTO DE LA SPO.	3-92
ILUSTRACIÓN 3-7 EJEMPLO DE LA PRESENCIA DE VARIAS ENTIDADES EXTERNAS DEL MISMO TIPO.	3-93
ILUSTRACIÓN 3-8 CONSIDERACIONES PARA LA ENTIDAD ESTUDIANTE.	3-94
ILUSTRACIÓN 3-9 ENTIDAD EXTERNA SPO REMOTA.	3-94
ILUSTRACIÓN 3-10 DIAGRAMA 0 DE LA SPO (SEGMENTO 1/4).	3-95
ILUSTRACIÓN 3-11 DIAGRAMA 0 PARA LA SPO (SEGMENTO 2/4).	3-97
ILUSTRACIÓN 3-12 DIAGRAMA 0 DE LA SPO (SEGMENTO 3/4)	3-99
ILUSTRACIÓN 3-13 DIAGRAMA 0 PARA LA SPO (SEGMENTO 4/4)	3-100
ILUSTRACIÓN 3-14 ENTIDAD RELACIÓN PARA LA SPODB (1/3).	3-108
ILUSTRACIÓN 3-15 ENTIDAD RELACIÓN PARA LA SPODB (2/3).	3-109
ILUSTRACIÓN 3-16 ENTIDAD RELACIÓN PARA LA SPODB (3/3).	3-109
ILUSTRACIÓN 3-17 DIAGRAMA DE LA SPODB.	3-110
ILUSTRACIÓN 3-18 TABLA ‘T_CARRERA’ DE LA SPODB.	3-112
ILUSTRACIÓN 3-19 TABLA ‘T_CONFIGURACION’ DE LA SPODB.	3-112
ILUSTRACIÓN 3-20 ARCHIVO DE DATOS XML QUE SE ESTÁ EDITANDO EN WORD 2003.	3-115
ILUSTRACIÓN 3-21 ELEMENTOS COMPLEJOS PARA EL ESQUEMA DEL DOCUMENTO INTELIGENTE (1/2).	3-120
ILUSTRACIÓN 3-22 ELEMENTOS COMPLEJOS PARA EL ESQUEMA DEL DOCUMENTO INTELIGENTE (2/2).	3-120
ILUSTRACIÓN 3-23 ENUMERACIÓN DEFINIDA PARA EL ESQUEMA DEL DOCUMENTO INTELIGENTE.	3-121
ILUSTRACIÓN 3-24 ÁRBOL JERÁRQUICO DEL ESQUEMA XML PARA EL DOCUMENTO INTELIGENTE.	3-122
ILUSTRACIÓN 3-25 AYUDAS DE ERROR QUE PRESENTA WORD CUANDO SE ETIQUETA MAL UN DOCUMENTO.	3-123
ILUSTRACIÓN 3-26 DIAGRAMA DEL ESQUEMA XML DISEÑADO PARA EL FORMULARIO DE INFOPATH.	3-127
ILUSTRACIÓN 3-27 ESQUEMAS XML QUE UTILIZARÁ EL FORMULARIO DE INFOPATH.	3-127
ILUSTRACIÓN 3-28 CAPAS LÓGICAS DE LA SPO.	3-128
ILUSTRACIÓN 3-29 CAPA LÓGICA DE INTERFAZ DE USUARIO DE LA SPO.	3-129
ILUSTRACIÓN 3-30 CAPA LÓGICA DE NEGOCIO DE LA SPO.	3-130
ILUSTRACIÓN 3-31 CAPA LÓGICA DE ACCESO A DATOS Y SERVICIOS.	3-130
ILUSTRACIÓN 3-32 RELACIÓN DE PROCESOS CON LA CAPA LÓGICA DE INTERFAZ DE USUARIO.	3-132
ILUSTRACIÓN 3-33 INTERFAZ DE USUARIO DE MICROSOFT WORD 2003.	3-133
ILUSTRACIÓN 3-34 INTERFACES DEL PANEL DE TAREAS DE ACCIÓN.	3-133
ILUSTRACIÓN 3-35 DIVISIÓN EN SECCIONES DE UN DOCUMENTO A SER AUTO-GENERADO.	3-141
ILUSTRACIÓN 3-36 CLASES PARA LA LÓGICA DE LA INTERFAZ DE USUARIO DEL DOCUMENTO INTELIGENTE.	3-144

ILUSTRACIÓN 3-37 DISEÑO DE INTERFAZ DE USUARIO DEL FORMULARIO DE INFOPATH.	3-148
ILUSTRACIÓN 3-38 CLASE PARA LA LÓGICA DE INTERFAZ DE USUARIO DEL FORMULARIO DE INFOPATH.	3-150
ILUSTRACIÓN 3-39 RELACIÓN DE PROCESOS CON LA CAPA LÓGICA DE NEGOCIOS.	3-153
ILUSTRACIÓN 3-40 CLASES QUE DEFINEN LAS ENTIDADES PROPIAS DE LA SPO.	3-154
ILUSTRACIÓN 3-41 CLASES QUE DEFINEN LA LÓGICA INTELIGENTE DE LA SOLUCIÓN.	3-156
ILUSTRACIÓN 3-42 CLASES PARA LA LÓGICA DEL SERVICIO WEB.	3-157
ILUSTRACIÓN 3-43 RELACIÓN DE PROCESOS CON LA CAPA LÓGICA DE ACCESO A DATOS Y SERVICIOS.	3-158
ILUSTRACIÓN 3-44 CLASE PARA EL ACCESO A SERVICIOS WEB.	3-159
ILUSTRACIÓN 3-45 CLASES PARA EL ACCESO A FUENTES DE DATOS.	3-161
ILUSTRACIÓN 3-46 PAQUETES ASOCIADOS A CADA PROCESO DEFINIDO EN EL DIAGRAMA 0 DE LA SPO.	3-164
ILUSTRACIÓN 3-47 DIRECTORIO DONDE SE ENCUENTRA EL ARCHIVO DE MANIFIESTO.	3-171
ILUSTRACIÓN 3-48 CERTIFICADO DIGITAL UTILIZADO POR EL DOCUMENTO INTELIGENTE.	3-173
ILUSTRACIÓN 3-49 ESTRUCTURA DEL ARCHIVO DE MANIFIESTO FIRMADO.	3-173
ILUSTRACIÓN 3-50 DETALLES DE LA FIRMA DIGITAL PARA EL CÓDIGO DEL DOCUMENTO INTELIGENTE.	3-175
ILUSTRACIÓN 3-51 CUADRO DE DIALOGO PLANTILLAS Y COMPLEMENTOS.	3-176
ILUSTRACIÓN 3-52 CUADRO DE DIALOGO BIBLIOTECA DE ESQUEMAS.	3-177
ILUSTRACIÓN 3-53 ARCHIVOS QUE COMPONEN EL DOCUMENTO INTELIGENTE	3-177
ILUSTRACIÓN 3-54 ALGUNAS DE LAS ENTRADAS DEL REGISTRO DE WINDOWS ASOCIADAS CON EL DOCUMENTO INTELIGENTE.	3-178
ILUSTRACIÓN 3-55 PANTALLA INICIAL DEL INSTALADOR PARA EL FORMULARIO DE INFOPATH.	3-181
ILUSTRACIÓN 3-56 ENTRADAS DEL REGISTRO DE WINDOWS PARA EL FORMULARIO DE INFOPATH.	3-181
ILUSTRACIÓN 3-57 FORMULARIO DE INFOPATH LUEGO DE SER INSTALADO.	3-182
ILUSTRACIÓN 3-58 BARRA DE HERRAMIENTAS PERSONALIZADA DOCUMENTO INTELIGENTE.	3-185
ILUSTRACIÓN 3-59 "FORMULARIO DE CONFIGURACIÓN" OPCIÓN "SINCRONIZAR DATOS SAE"	3-186
ILUSTRACIÓN 3-60 CUADRO DE DIÁLOGO DE DATOS PARA SINCRONIZACIÓN.	3-186
ILUSTRACIÓN 3-61 BARRA DE HERRAMIENTAS PERSONALIZADA DOCUMENTO INTELIGENTE.	3-188
ILUSTRACIÓN 3-62 VALIDACIÓN INMEDIATA DE INFORMACIÓN.	3-188
ILUSTRACIÓN 3-63 MENSAJE DE ERROR EN LOS DATOS.	3-189
ILUSTRACIÓN 3-64 CONFIRMACIÓN DE ELIMINACIÓN.	3-189
ILUSTRACIÓN 3-65 ACTUALIZACIÓN DE INFORMACIÓN.	3-191
ILUSTRACIÓN 3-66 BARRA DE HERRAMIENTA PERSONALIZADA DOCUMENTO INTELIGENTE.	3-193
ILUSTRACIÓN 3-67 CUADRO DE DIÁLOGO PARA CAMBIO DE ESTADO.	3-194
ILUSTRACIÓN 3-68 BARRA DE HERRAMIENTAS PERSONALIZADA DEL DOCUMENTO INTELIGENTE.	3-200
ILUSTRACIÓN 1-1 COMPONENTES .NET QUE DEBEN ESTAR INSTALADOS EN EL EQUIPO.	240
ILUSTRACIÓN 1-2 CUADRO DE DIÁLOGO PARA LA INSTALACIÓN DE PERSONALIZADA DE OFFICE 2003.	241
ILUSTRACIÓN 1-3 COMPONENTES DE OFFICE 2003 QUE DEBEN SER INSTALADOS EN EL EQUIPO.	242
ILUSTRACIÓN 1-4 MENÚ CONTEXTUAL DEL INSTALADOR DEL OFFICE 2003.	243
ILUSTRACIÓN 2-1 VENTANA ENTERPRICE MANAGER DE SQL SERVER SECCIÓN DATABASES.	244
ILUSTRACIÓN 2-2 CUADRO DE DIÁLOGO PARA ADJUNTAR LA BASE DE DATOS EN SQL SERVER.	244
ILUSTRACIÓN 2-3 PANTALLA PRINCIPAL DEL CERTIFICADO DIGITAL.	245
ILUSTRACIÓN 2-4 ASISTENTE PARA IMPORTACIÓN DE CERTIFICADOS.	246
ILUSTRACIÓN 2-5 CUADRO DE DIÁLOGO PARA LA SELECCIÓN DEL ALMACÉN DE CERTIFICADOS.	246
ILUSTRACIÓN 2-6 CUADRO DE DIÁLOGO SELECCIONAR ALMACÉN DE CERTIFICADOS.	247
ILUSTRACIÓN 2-7 MENSAJE FINAL DE IMPORTACIÓN DEL CERTIFICADO.	247
ILUSTRACIÓN 2-8 DIRECTORIO PARA ENCONTRAR EL ARCHIVO DE CONFIGURACIÓN DE LA BASE DE DATOS.	248
ILUSTRACIÓN 2-9 TEXTO DEL ARCHIVO DE CONFIGURACIÓN DE LA BASE DE DATOS	248
ILUSTRACIÓN 2-10 PANTALLA DEL ADMINISTRADOR DEL SERVICIO DE SQL SERVER.	249
ILUSTRACIÓN 2-11 PANTALLA DE CONFIGURACIÓN DEL .NET FRAMAWORK.	250
ILUSTRACIÓN 2-12 MENÚ HERRAMIENTAS DE MS WORD 2003.	251
ILUSTRACIÓN 2-13 CUADRO DE DIÁLOGO PLANTILLAS Y COMPLEMENTOS OPCIÓN PAQUETES DE EXPANSIÓN XML.	251
ILUSTRACIÓN 2-14 ARCHIVO DE MANIFIESTO XML PARA EL PAQUETE DE EXPANSIÓN XML.	252
ILUSTRACIÓN 2-15 CUADRO DE DIÁLOGO DEL PROCESO DE DESCARGA DEL PAQUETE DE EXPANSIÓN.	252
ILUSTRACIÓN 2-16 CUADRO DE DIÁLOGO PLANTILLAS Y COMPLEMENTOS LUEGO DE LA DESCARGA DEL PAQUETE DE EXPANSIÓN.	253
ILUSTRACIÓN 2-17 ESQUEMA XML ADJUNTADO AL DOCUMENTO.	253
ILUSTRACIÓN 2-18 LA PLANTILLA (ARCHIVO .DOT) ASOCIADO AL DOCUMENTO.	254
ILUSTRACIÓN 2-19 MENSAJE DE AVISO PARA ERRORES CON EL CERTIFICADO DIGITAL.	254

ILUSTRACIÓN 2-20 INTERFAZ DE USUARIO DEL DOCUMENTO INTELIGENTE.....	255
ILUSTRACIÓN 1-1 BARRA DE HERRAMIENTAS PERSONALIZADA DOCUMENTO INTELIGENTE.....	257
ILUSTRACIÓN 1-2 “FORMULARIO DE CONFIGURACIÓN” OPCIÓN “SINCRONIZAR DATOS SAE”.....	258
ILUSTRACIÓN 1-3 BARRA DE HERRAMIENTAS PERSONALIZADA DOCUMENTO INTELIGENTE.....	259
ILUSTRACIÓN 1-4 MENSAJE DE ERROR EN LOS DATOS.....	259
ILUSTRACIÓN 1-5 VALIDACIÓN INMEDIATA DE INFORMACIÓN.....	260
ILUSTRACIÓN 1-6 ACTUALIZACIÓN DE INFORMACIÓN.....	261
ILUSTRACIÓN 1-7 CONFIRMACIÓN DE ELIMINACIÓN.....	261
ILUSTRACIÓN 1-8 BARRA DE HERRAMIENTA PERSONALIZADA DOCUMENTO INTELIGENTE.....	262
ILUSTRACIÓN 1-9 FORMULARIO DE INFOPATH (1/3).....	266
ILUSTRACIÓN 1-10 FORMULARIO DE INFOPATH (2/3).....	267
ILUSTRACIÓN 1-11 MENÚ CONTEXTUAL PARA EL FORMULARIO DE INFOPATH.....	268
ILUSTRACIÓN 1-12 FORMULARIO DE INFOPATH (3/3).....	268
ILUSTRACIÓN 2-1 PANEL DE TAREAS DE ACCIÓN PARA LA GENERACIÓN DE DOCUMENTOS.....	269
ILUSTRACIÓN 2-2 MEMORANDO GENERADO POR EL DOCUMENTO INTELIGENTE.....	270
ILUSTRACIÓN 2-3 COMPROBACIÓN DE ARTÍCULOS Y CALIFICATIVOS PERSONALES.....	271
ILUSTRACIÓN 2-4 COMPROBACIÓN DE REMITENTE.....	271
ILUSTRACIÓN 2-5 ESCRITURA AUTOMÁTICA DE NOTAS EN PALABRAS.....	272
ILUSTRACIÓN 2-6 FORMATO ESPECIAL DE LOS DOCUMENTOS GENERADOS.....	272
ILUSTRACIÓN 2-7 DOCUMENTO GENERADO Y SUS ETIQUETAS.....	273
ILUSTRACIÓN 2-8 AYUDA RELATIVA AL CONTEXTO.....	274
ILUSTRACIÓN 2-9 ENVÍO DE DOCUMENTOS POR CORREO ELECTRÓNICO.....	275
ILUSTRACIÓN 2-10 “FORMULARIO DE CONFIGURACIÓN” OPCIÓN: “SEGURIDAD DE ESCRITURA”.....	276
ILUSTRACIÓN 2-11 “FORMULARIO DE CONFIGURACIÓN” OPCIÓN: “CAMBIO DE CONTRASEÑA”.....	276
ILUSTRACIÓN 2-12 PROTECCIÓN DE PÁRRAFOS Y SECCIONES.....	277
ILUSTRACIÓN 2-13 FORMULARIO DE CONFIGURACIÓN OPCIÓN ARCHIVOS DE TEXTO.....	278
ILUSTRACIÓN 3-1 BARRA DE HERRAMIENTAS PERSONALIZADA DEL DOCUMENTO INTELIGENTE.....	278
ILUSTRACIÓN 3-2 GENERACIÓN DE REPORTES.....	279
ILUSTRACIÓN 3-3 GENERACIÓN DE REPORTES 2.....	280
ILUSTRACIÓN 3-4 “FORMULARIO DE CONFIGURACIÓN” OPCIÓN: “DIRECCIONES URL”.....	281

ÍNDICE DE TABLAS

TABLA 2-1 LISTA DE ELEMENTOS AGREGADOS AL MODELO DE OBJETOS DE WORD.....	2-29
TABLA 2-2 ARCHIVOS QUE COMPONEN UNA PLANTILLA DE INFOPATH.....	2-34
TABLA 3-1 DESCRIPCIÓN DEL CASO DE USO “ADMINISTRACIÓN DE PROYECTOS DE TITULACIÓN”.....	3-76
TABLA 3-2 DESCRIPCIÓN DEL CASO DE USO “SINCRONIZACIÓN CON LA INFORMACIÓN DEL SAE”.....	3-77
TABLA 3-3 DESCRIPCIÓN DEL CASO DE USO “INGRESO DE NUEVOS PPTS”.....	3-77
TABLA 3-4 DESCRIPCIÓN DEL CASO DE USO “ACTUALIZACIÓN DE LA INFORMACIÓN DE LOS PPTS”.....	3-78
TABLA 3-5 DESCRIPCIÓN DEL CASO DE USO “ADMINISTRACIÓN DE ESTADOS”.....	3-78
TABLA 3-6 DESCRIPCIÓN DEL CASO DE USO “IDENTIFICACIÓN DEL ESTADO ACTUAL DE UN PPT O PT”..	3-78
TABLA 3-7 DESCRIPCIÓN DEL CASO DE USO “MODIFICACIÓN DEL ESTADOS DE UN PPT O PT”.....	3-79
TABLA 3-8 DESCRIPCIÓN DEL CASO DE USO “ELABORACIÓN AUTOMÁTICA DE CARTAS Y MEMORANDOS”.....	3-79
TABLA 3-9 DESCRIPCIÓN DEL CASO DE USO “VISUALIZACIÓN DE INFORMACIÓN SENSITIVA AL CONTEXTO”.....	3-80
TABLA 3-10 DESCRIPCIÓN DEL CASO DE USO “ENVÍO DIRECTO DE DOCUMENTOS VÍA E-MAIL”.....	3-80
TABLA 3-11 DESCRIPCIÓN DEL CASO DE USO “EDICIÓN DEL CONTEXTO DE LOS DOCUMENTOS”.....	3-81
TABLA 3-12 DESCRIPCIÓN DEL CASO DE USO “EDICIÓN DEL TEXTO COMÚN”.....	3-81
TABLA 3-13 DESCRIPCIÓN DEL CASO DE USO “ELABORACIÓN AUTOMÁTICA DE REPORTES”.....	3-81
TABLA 3-14 DESCRIPCIÓN DEL CASO DE USO “UTILIZACIÓN DE FILTROS DE INFORMACIÓN”.....	3-82
TABLA 3-15 DESCRIPCIÓN DEL CASO DE USO “CONFIGURACIÓN DE DIRECCIONES”.....	3-82
TABLA 3-16 DICCIONARIO DE DATOS PARA LA SPO (1/3).....	3-97
TABLA 3-17 DICCIONARIO DE DATOS PARA LA SPO (2/3).....	3-98
TABLA 3-18 DICCIONARIO DE DATOS PARA LA SPO (3/3).....	3-99
TABLA 3-19 ALMACENES DE DATOS Y ENTRADAS DEL DICCIONARIO DE DATOS ASOCIADAS.....	3-104
TABLA 3-20 ALMACENES DE DATOS Y SUS TABLAS EQUIVALENTES EN LA SPODB.....	3-105
TABLA 3-21 ENTRADAS DEL DICCIONARIO DE DATOS Y SUS EQUIVALENTES EN LAS TABLAS DE LA SPODB.....	3-107
TABLA 3-22 DESCRIPCIÓN DE ÁREAS PARA LA INTERFAZ DE USUARIO DEL PANEL DE TAREAS.....	3-134
TABLA 3-23 INTERFAZ DE USUARIO PARA EL INGRESO Y LA ACTUALIZACIÓN DE PROYECTOS.....	3-135
TABLA 3-24 INTERFAZ DE USUARIO PARA EL CAMBIO DE ESTADOS DE UN PROYECTO.....	3-136
TABLA 3-25 INTERFAZ DE USUARIO PARA LA CONFIGURACIÓN DEL SISTEMA.....	3-136
TABLA 3-26 INTERFAZ PARA LA SINCRONIZACIÓN CON INFORMACIÓN DEL SAE.....	3-138
TABLA 3-27 INTERFAZ DE DIRECCIONES Y PERMISOS DE EDICIÓN DE TEXTO.....	3-138
TABLA 3-28 INTERFAZ PARA CAMBIO DE NOMBRE DE USUARIO Y CONTRASEÑAS.....	3-139
TABLA 3-29 INTERFAZ DE USUARIO PARA LA EDICIÓN DE LOS DATOS DE LA CARRERA.....	3-139
TABLA 3-30 INTERFAZ DE USUARIO PARA EL TEXTO COMÚN REPETITIVO.....	3-140
TABLA 3-31 DESCRIPCIÓN DE LAS ÁREAS QUE COMPREDEN LA INTERFAZ DE USUARIO DE LOS FORMULARIOS.....	3-140
TABLA 3-32 DESCRIPCIÓN DE LAS ÁREAS QUE COMPONEN LA INTERFAZ DEL FORMULARIO DE INFOPATH.....	3-148
TABLA 3-33 PRUEBA PARA LA SINCRONIZACIÓN CON LA INFORMACIÓN DEL SAE.....	3-185
TABLA 3-34 PRUEBA DE INGRESO DE NUEVOS PPTS.....	3-187
TABLA 3-35 PRUEBA DE ACTUALIZACIÓN DE LA INFORMACIÓN DE LOS PPTS.....	3-190
TABLA 3-36 PRUEBA DE IDENTIFICACIÓN DEL ESTADO ACTUAL DE UN PROYECTO.....	3-192
TABLA 3-37 PRUEBA DE LA MODIFICACIÓN DEL ESTADO DE UN PROYECTO.....	3-193
TABLA 3-38 PRUEBAS PARA LA GENERACIÓN AUTOMÁTICA DE DOCUMENTOS.....	3-196
TABLA 3-39 PRUEBA PARA LA VISUALIZACIÓN DE INFORMACIÓN SENSITIVA AL CONTEXTO.....	3-197
TABLA 3-40 PRUEBAS PARA EL ENVÍO DIRECTO VÍA E-MAIL.....	3-198
TABLA 3-41 PRUEBAS PARA LA EDICIÓN DEL CONTEXTO DE LOS DOCUMENTOS.....	3-199
TABLA 3-42 PRUEBA DE EDICIÓN DEL TEXTO COMÚN (REPETITIVO).....	3-199
TABLA 3-43 PRUEBA PARA LA UTILIZACIÓN DE FILTROS DE INFORMACIÓN.....	3-200
TABLA 3-44 PRUEBA PARA LA CONFIGURACIÓN DE DIRECCIONES URL.....	3-201
TABLA 3-45 PRUEBA DE ACEPTACIÓN NÚMERO UNO.....	3-202
TABLA 3-46 PRUEBA DE ACEPTACIÓN NÚMERO DOS.....	3-203
TABLA 3-47 PRUEBA DE ACEPTACIÓN NÚMERO TRES.....	3-204
TABLA 3-48 PRUEBA DE ACEPTACIÓN NÚMERO CUATRO.....	3-204
TABLA 3-49 PRUEBA DE ACEPTACIÓN NÚMERO CINCO.....	3-205
TABLA 3-50 PRUEBA DE ACEPTACIÓN NÚMERO SEIS.....	3-205
TABLA 1-1 POSIBLES PANTALLAS QUE PRESENTA EL FORMULARIO “CAMBIO DE ESTADOS”.....	265

RESUMEN

La creación de sistemas informáticos que agilicen las tareas cotidianas, es una práctica muy común en nuestros días, las opciones para desarrollar una solución son varias, y los requerimientos cada vez son más exigentes.

En el caso específico del trabajo de oficina, especialmente a nivel de secretarías, los problemas por lo general giran entorno a la manipulación de documentos, ya sea para archivarlos o para generarlos. En las instituciones grandes, estos problemas son más notorios, ya que existe un bajo rendimiento del personal al tratar de encontrar un determinado archivo, o conseguir los datos para generar algún otro.

Existen varios mecanismos que ayudan a solucionar este inconveniente, y en algunos casos gran inconveniente; pero en especial se puede mencionar una alternativa bastante interesante y actual, que ofrece una gran ayuda en el momento de crear un sistema computacional para el manejo de documentos.

Esta alternativa son los denominados Documentos Inteligentes, que no son más que los típicos documentos electrónicos con los cuales los usuarios están acostumbrados a trabajar, pero que incluyen características especiales que minimizan el trabajo; como por ejemplo, inserción de texto preescrito, sensibilidad al contexto de los documentos, ayuda dinámica para el usuario en relación a su ubicación dentro de los documentos, entre otras.

Lo cierto es que técnicamente los Documentos Inteligentes fundamentan su funcionamiento en la utilización del estándar XML o Lenguaje de Marcado eXtensible.

El XML básicamente, permite identificar toda la información contenida en un archivo de texto, mediante la utilización de etiquetas personalizadas que se ajustan a las necesidades del usuario. Ofrece muchas facilidades para el

intercambio de información entre distintas aplicaciones o sistemas; razón por la cual, Microsoft decidió incluir el XML en el núcleo de uno de sus paquetes comerciales más vendidos en todo el mundo, el Microsoft Office.

Y justamente dos de las aplicaciones de Office, el MS Word y el MS Excel, en su versión 2003, son las herramientas que hacen posible la creación de Documentos Inteligentes con productos Microsoft.

Por años las aplicaciones de Office han permitido que los desarrolladores utilicen su estructura interna para la creación de soluciones personalizadas, esto es posible gracias a la arquitectura basada en componentes COM, con la que están desarrolladas.

Las soluciones personalizadas de Office, hasta antes del apareamiento de la tecnología .NET, tenían un gran pero, el de la seguridad. Este siempre será un factor muy importante a ser considerado en el momento de seleccionar una alternativa para el desarrollo de cualquier aplicación.

Sin embargo, la tecnología .NET ofrece características bastante atractivas, ya que el código se ejecuta bajo un ambiente “administrado”, en donde se utiliza un lenguaje intermedio que es compilado por un componente especial de código denominado *Framework* .NET, que además se encarga entre otras cosas de administrar la utilización de recursos de memoria, y las políticas de seguridad para la ejecución del código, lo que permite crear soluciones mejor estructuradas, flexibles y seguras.

La combinación de estas dos tecnologías: el XML y el .NET, para la creación de Documentos Inteligentes, permite hacer de la personalización de las aplicaciones de Office una tarea muy profesional, práctica, segura y funcional, con alcances delimitados principalmente por las capacidades o conocimientos del desarrollador.

En este punto, hay que tener claro que los Documentos Inteligentes vistos desde el lado de Microsoft, son una forma de personalización de las aplicaciones de Office, que explota muchas de las características del XML, lo cual no quiere decir que otros métodos de personalización; tales como, los macros con código VBA o los COM *Add-Ins* hayan desaparecido, sino que por el contrario pueden coexistir con esta nueva alternativa para la personalización.

Y lo que es más, ahora con un nuevo enfoque: el Office 2003 dejó de ser una herramienta unipersonal utilizada de manera aislada, para convertirse en una herramienta cooperativa, ya que gracias a la inclusión del XML en el núcleo de las aplicaciones de Office, ahora existe el soporte para la tecnología de Servicios Web XML, que es una buena alternativa para la creación de aplicaciones distribuidas.

Con los Servicios Web, los usuarios pueden tener acceso a información remota que se intercambia en formato XML, y los desarrolladores pueden crear soluciones personalizadas más sofisticadas, funcionales y que rompan las barreras de la distancia.

Finalmente, también hay que mencionar que Microsoft, al dar un gran impulso a la utilización del XML en las aplicaciones de escritorio, incluyó en la versión del Office 2003 el MS Office InfoPath, una nueva aplicación diseñada específicamente para facilitar el trabajo con formularios electrónicos, basada en su totalidad en la utilización del XML; cuenta con muchos beneficios para el almacenamiento, transferencia, manipulación y presentación de la información, y además puede ser personalizada con la utilización de código .NET.

PRESENTACIÓN

Este proyecto denominado: “XML Y .NET PARA EL DESARROLLO DE SOLUCIONES PERSONALIZADAS DE OFFICE”, utiliza la técnica más actual en cuanto a la personalización de aplicaciones de oficina con productos Microsoft.

Aquí se presentarán algunas de las ventajas de la utilización del XML en los computadores de escritorio, de la tecnología .NET para el desarrollo de aplicaciones, y del Office 2003 como una plataforma para el desarrollo de soluciones personalizadas. Entre los objetivos específicos planteados para este proyecto se pueden mencionar:

- Familiarizarse con las herramientas tanto a nivel de usuario como de desarrollador que se requieran para la creación de soluciones basadas en aplicaciones de Microsoft Office, aprovechando las características de la versión Office 2003 que permiten la interoperabilidad con .NET y XML.
- Comparar alternativas para la creación de soluciones personalizadas de Office, tomando en consideración las ventajas que se ofrecen en el tratamiento de la información corporativa.
- Diseñar e implementar un sistema distribuido utilizando Servicios Web XML, que ofrezca como interfaz de usuario un Documento Inteligente (interfaz personalizada de Word) y un Formulario de Infopath, que facilite el acceso a la información sobre Proyectos de Titulación de las carreras de electrónica, accediendo a datos alojados en diferentes equipos.
- Analizar mecanismos para proveer seguridad a diferentes niveles, utilizando certificados y firmas digitales, directivas de seguridad para la ejecución de código, bloqueo de párrafos en los documentos, etc.

Tanto el marco teórico, como los detalles del diseño y la implementación del sistema planteado como objetivo de este proyecto, se han incluido en tres capítulos; una pequeña descripción del contenido de cada uno de ellos, se presenta a continuación:

En el Capítulo 1 se mencionan algunas de las características más importantes de la versión de Office 2003; se hace un pequeña reseña sobre los mecanismos de personalización de las aplicaciones de Office que han existido por años; se exponen algunas de las ventajas de la tecnología .NET y los detalles particulares que deben ser considerados para la personalización de las aplicaciones de Office.

En el Capítulo 2 se presentan ideas generales sobre el XML y su utilización dentro del Office; se dan a conocer las características más importantes del InfoPath; se presenta una descripción sintetizada de lo que en Microsoft se conoce como la Tecnología Inteligente, en donde los principales exponentes son las Etiquetas y los Documentos Inteligentes; se mencionan algunas herramientas existentes en el mercado, compatibles con el ambiente de desarrollo de Visual Studio .NET, que pueden ser utilizadas para facilitar el desarrollo de aplicaciones personalizadas de Office; finalmente se indican algunas de las consideraciones de seguridad para el desarrollo de este tipo de soluciones.

En el Capítulo 3 se presenta una descripción detallada de las características de la solución personalizada de Office que fue desarrollada con el objetivo de automatizar las tareas de oficina que realizan las secretarias de las Carreras de Electrónica, en lo referente a la elaboración de cartas y memorandos relacionados con los Proyectos de Titulación de los estudiantes; se presentan las consideraciones del diseño, incluyendo un análisis de los requerimientos y la situación actual; se detallan los aspectos más importantes sobre la arquitectura de la solución y la forma en la que fueron implementados cada uno de sus elementos, también se mencionan las políticas de seguridad empleadas y los aspectos necesarios para que la solución pueda ser instalada en cualquier computador.

En el Capítulo 4 se presentan conclusiones y recomendaciones en base a las experiencias tanto positivas como negativas que se obtuvieron durante el desarrollo de este proyecto.

Adicionalmente, los Anexos A y B proporcionan información compacta pero bastante útil para incursionar en el desarrollo de soluciones personalizadas de Office, haciendo una pequeña referencia sobre el modelo de objetos de Word 2003, la utilización del VBE (*Visual Basic Editor*) y el examinador de objetos, este último una herramienta de mucha utilidad para el trabajo con los modelos de objetos.

Los Anexos C y D presentan una lista de los Ensamblados de Interoperabilidad Primaria (PIAs) necesarios para el desarrollo de soluciones basadas en Office 2003 con código .NET, y la guía de instalación de los mismos.

El Anexo E presenta la nomenclatura utilizada en los Diagramas de Flujo de Datos y el Diccionario de Datos, que fueron las metodologías empleadas para el diseño de la solución personalizada.

En los Anexos F y G se encontrarán el manual de instalación y el manual de usuario, y finalmente en el Anexo H se presenta un pequeño análisis de la relación Costo Beneficio de la implementación del sistema.

CAPÍTULO 1. SOLUCIONES PERSONALIZADAS DE MICROSOFT OFFICE 2003

En este capítulo se realiza una breve introducción al desarrollo de soluciones personalizadas de Office, incluyendo:

- La importancia de la información contenida en los documentos
- Particularidades de la versión de Office 2003
- La Arquitectura de Componentes COM y el Modelo de Objetos de Office
- Alternativas para el desarrollo de soluciones de Office con código no administrado
- Características básicas de la tecnología .NET
- Introducción al desarrollo de soluciones personalizadas de Office con código administrado
- Interacción entre la tecnología COM (no administrada), con la tecnología .NET (administrada)

1.1 MICROSOFT OFFICE 2003 Y LAS SOLUCIONES PERSONALIZADAS

1.1.1 INTRODUCCIÓN

En este proyecto se estudia una alternativa que existe en el mercado para el manejo de la información, la misma que consiste en la utilización de los documentos corporativos creados con el paquete de aplicaciones de Microsoft Office 2003, de tal forma que al ser personalizados se puedan convertir en verdaderas herramientas eficaces y prácticas para la recopilación y presentación de la información.

Considerando que en todas las empresas, los documentos son parte fundamental del trabajo diario, y en la mayoría de casos representan la primera forma de

interacción con los empleados, socios y clientes, parecería una buena idea incluir funcionalidad extra al proceso cotidiano de su creación.¹

1.1.2 DESARROLLO DE SOLUCIONES BASADAS EN LAS APLICACIONES DE OFFICE

El desarrollo de soluciones basadas en el paquete de oficina (*suite* ofimática)² Microsoft Office, consiste en la personalización de cualquiera de sus aplicaciones (Word, Excel, PowerPoint, etc.), de tal manera que se pueda lograr que determinadas funciones y servicios sean mejores, más rápidos y eficientes³.

La versión Office 11⁴, comercialmente conocida como Office System 2003, es hasta el momento la más completa de todas, y posee un conjunto de herramientas que permiten la creación de aplicaciones mucho más robustas, seguras y escalables.

1.1.3 IMPORTANCIA DEL DESARROLLO DE LAS SOLUCIONES PERSONALIZADAS DE OFFICE

La importancia de la creación de este tipo de soluciones de oficina, radica en que la estructura bastante flexible que posee el Office, permite combinar la funcionalidad existente en cada una de sus aplicaciones, con otras funciones adicionales, con lo cual se ahorra mucho tiempo y esfuerzo, permitiendo concentrar todo el trabajo de desarrollo en implementar algo que en realidad no se ha realizado antes, y que constituya un verdadero aporte para los procesos empresariales, y que se adapte a las necesidades de los usuarios.

¹ Revista *Computerworld* de noviembre de 2004, artículo *Documents get smart*”, escrito por Kym Gilhooley.

² Se conoce como *suite* ofimática a un paquete de programas informáticos diseñados para el trabajo de oficina; sus aplicaciones son capaces de interactuar mutuamente y por lo general poseen una interfaz de usuario homogénea [Ref. 13].

³ Idea redactada tomando como fuente la introducción de la Referencia 11.

⁴ Las versiones de Office, tienen un nombre comercial y un número que representa su versión, por ejemplo la versión de Office XP es el Office 10, por lo que su sucesor el Office System 2003 es el Office 11.

1.1.4 NOVEDADES PARA LOS DESARROLLADORES EN EL OFFICE 2003

El Office 2003 permite a los desarrolladores utilizar la interfaz de sus aplicaciones para crear soluciones altamente interconectadas, logrando que tanto los usuarios como los documentos, interactúen directamente con las fuentes de datos corporativas, lo que facilita la tarea de recopilación y presentación de la información, ya sea dentro de la empresa o a los clientes y proveedores que se encuentran fuera de ella.

1.1.4.1 Manejo de Datos con XML

El Office 2003 incluye en el núcleo de todas sus aplicaciones el soporte para el manejo de datos XML, lo que quiere decir que cada una de sus aplicaciones se pueden comunicar (enviando y recibiendo información) mediante datos en formato XML, sin importar ni la procedencia ni la distancia que exista entre la fuente y el destino de los mismos.

1.1.4.2 Consumo de Servicios Web XML

Los Servicios Web XML⁵ facilitan a los desarrolladores la transferencia de datos entre aplicaciones y fuentes externas, permiten integrar datos fácilmente a cualquier solución, sin abandonar el ambiente de desarrollo de Office.

Más detalles de la utilización de Servicios Web XML dentro del Office, se encuentran en la Sección 2.4.

1.1.4.3 Desarrollo de Aplicaciones Inteligentes

El concepto de Aplicaciones Inteligentes en el Office tiene sus orígenes desde la versión de Office XP, en donde Microsoft introdujo un elemento innovador denominado Etiquetas Inteligentes [Sección 2.5.1].

La tecnología de Etiquetas Inteligentes permite a los desarrolladores crear soluciones que reconocen inteligentemente términos específicos dentro de

⁵ Un Servicio Web es como una caja negra que se puede utilizar sin conocer los detalles de su implementación, utiliza protocolos de Internet y formatos de datos estándar (HTTP y XML); tiene una interfaz definida en términos de los mensajes que se pueden aceptar y generar; los clientes del servicio pueden implementarse en cualquier plataforma [Ref. 40].

documentos y mensajes de correo, permitiendo a los usuarios llevar a cabo acciones personalizadas asociadas a dichos términos.

En Office 2003 no sólo se ha mejorado la funcionalidad de las Etiquetas Inteligentes haciéndolas aún más sofisticadas, sino que también se ha ampliado el concepto, permitiendo a los desarrolladores crear soluciones a nivel de documentos realizando tareas específicas según el contexto de los mismos, a este tipo de soluciones se las denomina Documentos Inteligentes [Sección 2.5.2].

1.2 DESARROLLO CON ALTERNATIVAS QUE UTILIZAN CÓDIGO NO ADMINISTRADO

Se presenta una descripción corta de las tecnologías de desarrollo con código no administrado⁶, que han sido utilizadas por años y continúan íntimamente relacionadas con la creación de las aplicaciones personalizadas de Office.

1.2.1 COMPONENT OBJECT MODEL

COM (*Component Object Model*), que en español se traduce como Modelo de Objetos Componentes, es la clave que hace posible la creación de las soluciones personalizadas de Office, gracias a una característica denominada automatización.

Se dice que una aplicación desarrollada bajo la arquitectura COM, soporta automatización (antes denominada automatización OLE⁷) si: puede utilizar objetos implementados en otras aplicaciones (cliente de automatización), y/o exponer sus objetos para que sean utilizados por otras aplicaciones (servidor de automatización).

⁶ La diferencia entre lo que se denomina código no administrado y código administrado, se aclara en la Sección 1.3.1.1.

⁷ OLE (*Object Linking and Embedding* / Vinculación e Incrustación de Objetos), es un mecanismo que permite a los usuarios crear y editar documentos que contengan elementos u "objetos" creados en distintas aplicaciones.

Las aplicaciones que exponen sus interfaces de automatización, deben proporcionar una librería de objetos, que no es más que un archivo que contiene información sobre los objetos.

1.2.2 EL MODELO DE OBJETOS DE OFFICE

Sin duda alguna, una parte esencial del desarrollo de cualquier solución personalizada de Office, es comprender correctamente sus objetos en cuanto a su funcionamiento, tanto a nivel individual como colectivo; el Office posee decenas de objetos (con métodos, eventos y propiedades)⁸, por lo que aprender el comportamiento de todos es complejo.

Los Modelos de Objetos describen los objetos disponibles en cada una de las aplicaciones, convirtiéndose en una guía necesaria utilizada por los desarrolladores.

En vista de que la mayoría de objetos están relacionados entre sí, los Modelos de Objetos muestran una jerarquía de objetos, con la cual los desarrolladores encuentran los caminos que se deben seguir, para seleccionar adecuadamente los objetos y métodos que se deben utilizar para lograr que una aplicación haga cualquier cosa que se necesite hacer.

Todas las aplicaciones de Office tienen un Modelo de Objetos independiente, y lógicamente existe uno general que contiene las características de los objetos comunes para todas.

A pesar de su importancia, en vista de que es un tema bastante extenso, en este proyecto no se profundiza en la descripción de los Modelos de Objetos de las aplicaciones de Office; para obtener mayor información se pueden consultar el Anexo A, o la Referencia 20.

⁸ Propiedad: Característica o atributo, que define o describe un objeto (tamaño, posición, forma).
Método: Acción que se puede realizar con un objeto (grabar, borrar, imprimir).
Evento: Acción que causa que un objeto reaccione (abrir, cerrar, dar un *clic*); un objeto también puede generar eventos.

1.2.3 GENERALIDADES DE VISUAL BASIC PARA APLICACIONES

El Visual Basic para Aplicaciones (VBA) es la combinación de un Ambiente de Programación Integrado (IDE - *Integrated Design Environment* o *Integrated Development Environment*), denominado Editor de Visual Basic (VBE) y el lenguaje de programación Visual Basic (VB).

El término “para aplicaciones” proviene del hecho de que tanto el lenguaje de programación Visual Basic como las herramientas de desarrollo en el Editor de Visual Basic, están integradas en las aplicaciones de Office tradicionales.⁹

Una guía sobre los elementos del VBE, y la utilización del Examinador de Objetos, que es una herramienta de mucha utilidad para consultar información sobre el Modelo de Objetos de las aplicaciones de Office (incluida en el VBE), se encuentra en el Anexo B.

1.2.3.1 El Desarrollo de Aplicaciones de Office Utilizando VBA

Hay muchas personas que piensan que el VBA es un IDE exclusivo para desarrollar macros¹⁰, sin embargo cuando se escribe código VBA, se puede acceder a decenas de objetos con todas sus propiedades y métodos, permitiendo crear soluciones personalizadas de Office bastante sofisticadas.

El desarrollo con VBA resulta práctico para soluciones relativamente simples, y que por lo general funcionan en un ambiente unipersonal, no obstante para aplicaciones más complejas, que implican múltiples usuarios, y distribuidas, se presentan problemas de rendimiento, deficiencias de seguridad y dificultades de interacción con datos de diversas fuentes.

⁹ En la versión de Office 2003 no todas las aplicaciones nuevas como InfoPath o OneNote tienen integrado el VBE, pero sí las tradicionales como Word, Excel, PowerPoint, Access, etc.

¹⁰ Un macro es un conjunto de instrucciones secuenciales guardadas en un archivo (.vsmacros) que se pueden ejecutar en cualquier momento [Ref. 41].

1.2.3.1.1 Características Principales del Desarrollo con VBA

- La programación con VBA, no es orientada a objetos, por el contrario es un tipo de programación estructurada, orientada a procedimientos.
- Las soluciones de Office creadas con VBA, se ejecutan en el mismo espacio de memoria de la aplicación anfitrión, su ejecución es rápida.
- El VBE establece de forma predeterminada las referencias a las librerías de objetos relacionadas con la aplicación anfitrión; por ejemplo, si se abre un VBE desde Microsoft Word, la librería de objetos de Word ya tendrá una referencia predeterminada.
- El VBA permite el acceso a bases de datos, utilizando referencias a librerías de tipos de objetos COM relacionados con manejo de datos, tales como ADO¹¹.

1.2.4 LOS COMPLEMENTOS EN LAS APLICACIONES DE OFFICE

Los complementos o *Add-Ins* son programas suplementarios que añaden características personalizadas a las aplicaciones de Office. En Office existen dos tipos de complementos: los complementos de aplicación específica y los complementos COM.

1.2.4.1 Complementos de Aplicación Específica

Los Complementos de Aplicación Específica o simplemente complementos, como su nombre lo sugiere funcionan únicamente en la aplicación de Office para la cual fueron creados; por ejemplo, si se quiere realizar la misma tarea para Word y para Excel, hay que desarrollar dos complementos independientes.

1.2.4.2 Complementos COM

Los Complementos COM o COM Add-Ins, existen en el Office desde la versión de Office 2000, y hasta antes del apareamiento de la versión de Office 2003 eran la

¹¹ ADO (*ActiveX Data Objects*) es una interfaz a nivel de aplicación que utiliza OLE-DB, una librería de interfaces COM que habilita el acceso universal a diversas fuentes de datos [Ref. 42].

manera más común, versátil y sobretodo profesional de personalizar y extender la funcionalidad de sus aplicaciones.

Los Complementos COM utilizan todas las ventajas de la tecnología COM, lo cual hace posible crear un complemento único que puede funcionar en múltiples aplicaciones de Office, lo que es una gran ventaja sobre los Complementos de Aplicación Específica.

1.3 GENERALIDADES DE LA TECNOLOGÍA .NET Y EL CÓDIGO ADMINISTRADO

La base de la tecnología .NET es el denominado *Framework* .NET, que en esencia es un entorno de programación multilenguaje, diseñado para cumplir con varias funciones.

1.3.1 ELEMENTOS PRINCIPALES DEL *FRAMEWORK* .NET

1.3.1.1 *Common Language Runtime*

El CLR (*Common Language Runtime* o Motor en Tiempo de Ejecución), es el elemento del *Framework* .NET, que da origen al concepto de código administrado; todo el código que pasa por el Motor en Tiempo de Ejecución se denomina código administrado, y todo el resto de código, se conoce como código no administrado.

1.3.1.2 Clases de Programación Unificadas

La biblioteca de clases ofrecida por el *Framework* .NET, es una colección completa, unificada, jerárquica y orientada a objetos, que se puede utilizar en todos los lenguajes de programación soportados por la tecnología .NET (VB .NET, C#, C++, J#), permitiendo a los desarrolladores elegir libremente el lenguaje de programación, obteniendo los mismos resultados.

1.3.2 EL CÓDIGO ADMINISTRADO EN LAS SOLUCIONES PERSONALIZADAS DE OFFICE

Las soluciones personalizadas de Office desarrolladas con código administrado tienen algunas características particulares, que difieren con las soluciones que utilizan código no administrado.

1.3.2.1 Interoperabilidad COM

Se puede considerar que en varios aspectos, el *Framework* .NET es una evolución de COM, lo cual no significa que tengan muchas similitudes entre sí, ya que por el contrario son bastante diferentes y constituyen un cambio radical para los desarrolladores que han trabajado por años bajo la arquitectura COM.

No obstante, en vista de que Office fue desarrollado en base a COM y OLE, no se puede prescindir de las interfaces COM expuestas por sus aplicaciones, es por ello que Microsoft ha integrado dentro del *Framework* .NET, los mecanismos necesarios que permiten interactuar a COM con .NET, en lo que se conoce como Interoperabilidad COM [Sección 1.4].

1.3.2.2 Ausencia de los Tipos de Datos *Variant*

Entre las principales características de los lenguajes administrados de .NET, está el hecho de que el CLR sólo entiende un sistema común de tipos de datos, en donde todos se heredan de un único tipo raíz (*Object*), esto permite que los datos sean tratados de una manera polimórfica sin la necesidad de recurrir a ningún otro tipo externo como el *Variant*, utilizado en los componentes COM.

1.3.2.3 Eliminación de Elementos no Utilizados

El Visual Basic y por ende el VBA, ya ofrecían la característica de eliminación automática de elementos no utilizados; sin embargo, a pesar de cumplir la misma

función del recolector de basura de .NET, los algoritmos¹² utilizados para la eliminación son diferentes en cada uno de los ambientes.

1.3.2.4 La Compilación JIT

La compilación JIT (*Just In Time*) del MSIL, está considerada como un proceso rápido y eficiente, pero en vista de que es una compilación bajo demanda, y el CLR además realiza comprobaciones de seguridad y tipos, los componentes que personalizan el Office, podrían ejecutarse ligeramente más lento que los componentes desarrollados con código no administrado.

1.4 INTERACCIÓN ENTRE EL CÓDIGO ADMINISTRADO Y EL NO ADMINISTRADO

Sabiendo que las aplicaciones de Office son desarrolladas en base a COM (código no administrado), es importante saber como interactúan con la tecnología .NET (código administrado).

1.4.1 DIFERENCIAS ENTRE COM Y .NET

Existen varios factores que hacen incompatibles a la tecnología de componentes COM, con la tecnología .NET, entre los que se pueden citar:

- En .NET, el CLR es el encargado de administrar la duración de los objetos; en COM, el cliente que consume el componente es quien administra el tiempo de vida de los objetos.
- En .NET, para saber la disponibilidad de un servicio ofrecido por un objeto se recurre a un proceso denominado reflexión¹³; en COM se solicita una

¹² Para ver más detalles del funcionamiento del algoritmo utilizado por el recolector de basura de .NET, se puede revisar el artículo "*Garbage Collection: Automatic Memory Management in the Microsoft .NET Framework*", de la revista MSDN, escrito por Jeffrey Richter [Ref. 48].

¹³ La Reflexión se utiliza para crear dinámicamente instancias de un tipo, enlazar el tipo a un objeto existente u obtener el tipo a partir de un objeto existente. A continuación, se pueden invocar los métodos del tipo u obtener acceso a los campos y propiedades correspondientes [Ref. 43].

interfaz que ofrece el servicio y acorde a su disponibilidad se obtiene o no un puntero a la interfaz.

- En .NET, los objetos describen sus características mediante manifiestos de ensamblado como metadatos¹⁴; en COM, las características de los objetos son descritas a través de las librerías de tipos.
- En .NET, las DLLs son auto descriptivas; es decir, la definición exacta de su contenido se encuentra dentro de si mismas; en COM a pesar de que existe la posibilidad de tener la definición embebida en los DLLs (por lo general se hace con las librerías de tipos en archivos independientes), esta definición no necesariamente tiene que ser completa, por lo que no se puede decir que en general las DLLs de COM son auto descriptivas.
- En .NET, los objetos pueden ser movidos de la memoria por razones de rendimiento, teniendo en consideración que siempre que se mueva un objeto se actualizarán todas las referencias que existan a dicho objeto; en COM una vez que se ha obtenido un puntero a un objeto, se crea una dependencia con la ubicación del objeto; es decir, no existe ningún mecanismo para trabajar con objetos que tengan ubicación dinámica.

1.4.2 OBJETIVO DE LA INTEROPERABILIDAD

Migrar toda la tecnología de componentes COM incluida en las aplicaciones de Office, sería un proceso bastante demorado que no se puede hacer de la noche a la mañana.

Razón por la cual, el objetivo principal de la interoperabilidad entre COM y .NET, es lograr que tanto los clientes administrados como los no administrados crean que están utilizando objetos de su misma tecnología, de una manera completamente transparente.

¹⁴ El Manifiesto de Ensamblado es el elemento que establece la identidad del ensamblado, especifica los archivos, tipos y recursos que componen el ensamblado, así como también especifica el conjunto de permisos necesarios para que el ensamblado se ejecute correctamente.

Los Metadatos son la información sobre el ensamblado, contenida en el Manifiesto de Ensamblado y requerida por el CLR para realizar los procesos de depuración, recolección de basura, control de seguridad, etc. [Ref. 44].

Para cumplir con este objetivo se deben resolver problemas que se presentan tanto en tiempo de diseño como en tiempo de ejecución.

En tiempo de diseño hay que considerar que para utilizar un complemento COM, es necesario conocer las características que definen a sus objetos; el inconveniente se presenta ya que la tecnología .NET sólo entiende metadatos de .NET, no entiende las librerías de tipos de los componentes COM.

En tiempo de ejecución, el problema principal se presenta con los tipos de datos utilizados por cada tecnología; hay que recordar que .NET utiliza tipos de datos generales que en ciertas ocasiones no son compatibles con los tipos de datos que utiliza COM; esto implica que se debe hacer una conversión en tiempo de ejecución para que cada tecnología trabaje con los tipos de datos que soporta.

La solución que se le ocurrió a Microsoft, para solucionar estos inconvenientes, fue crear un ensamblado denominado Ensamblado de Interoperabilidad, que cumpla dos funciones fundamentales:

- Proporcionar toda la información necesaria para construir un contenedor al que se puede llamar en tiempo de ejecución¹⁵, que sirva como intermediario entre el complemento COM y el CLR de .NET.
- Representar una librería de tipos COM, con metadatos correspondientes, o viceversa.

1.4.3 ENSAMBLADOS DE INTEROPERABILIDAD

Los Ensamblados de Interoperabilidad son los que en realidad permiten que toda la información de un componente COM pueda ser entendida al establecer una Referencia en tiempo de diseño desde un cliente .NET, y que todos los métodos puedan ser procesados normalmente en tiempo de ejecución.

¹⁵ Contenedor al que se puede llamar en tiempo de ejecución, es el nombre que recibe en español lo que en la literatura en inglés se conoce como *Runtime Callable Wrapper* (no es una traducción exacta).

La diferencia fundamental entre un Ensamblado de Interoperabilidad y un ensamblado común de .NET, es que los Ensamblados de Interoperabilidad únicamente contienen la definición de tipos, no la implementación; la definición se obtiene de una librería de tipos, y la implementación sigue siendo la misma proporcionada por el componente COM asociado a dicha librería.

1.4.3.1 Ensamblados de Interoperabilidad Primaria

Un mismo componente COM puede estar relacionado con varios Ensamblados de Interoperabilidad; ya que cualquier desarrollador puede crear su propio Ensamblado de Interoperabilidad, basado en la librería de tipos asociada al componente, e incluso sin darse cuenta ya que al utilizar el ambiente de desarrollo de Visual Studio .NET se genera automáticamente un Ensamblado de Interoperabilidad¹⁶.

Tener varios Ensamblados de Interoperabilidad para un mismo componente COM, produce un problema de compatibilidad, ya que a pesar de tener como fuente la misma librería de tipos y de cumplir la misma función, no se puede garantizar que todos los metadatos y las instrucciones para crear el contenedor en tiempo de ejecución embebidas en cada uno de ellos, sean exactamente iguales (no todos los desarrolladores piensan igual), ya que basta con que el nombre de un metadato no coincida, para que los Ensamblados de Interoperabilidad sean incompatibles.

Para evitar este problema, existen los Ensamblados de Interoperabilidad Primaria (PIA - *Primary Interop Assembly*), que realizan exactamente la misma función que cualquier Ensamblado de Interoperabilidad, pero con la gran diferencia de que estos han sido creados y firmados digitalmente por el desarrollador del componente COM.

¹⁶ Cuando se establece una Referencia a un objeto COM y el ambiente de desarrollo de .NET no encuentra ningún Ensamblado de Interoperabilidad asociado, ejecuta una herramienta denominada Tibimp (*Type Library Importer*) incluida en el *Framework* .NET, que genera un Ensamblado de Interoperabilidad básico.

En vista de que el propietario del componente COM, conoce a fondo todos los detalles de la implementación, puede incluir sin problema personalizaciones en el PIA; tales como, renombrar u ocultar elementos, añadir atributos, etc., de tal forma que al establecer una referencia en tiempo de diseño, el desarrollador siempre tenga disponibles los mismos métodos, propiedades y eventos, y que en tiempo de ejecución, el contenedor que se construye tenga toda la información necesaria para lograr un mejor rendimiento.

Existe un único PIA por cada componente COM, por lo que actúa como un estándar que permite la compatibilidad entre los sistemas que lo utilizan, y además permite trasladar las aplicaciones sin ningún problema, siempre y cuando el PIA se encuentre instalado en el equipo en el cual va a correr la aplicación¹⁷.

Cuando un PIA está instalado, el ambiente de desarrollo de .NET ya no genera un Ensamblado de Interoperabilidad automáticamente, sino que más bien utiliza directamente el PIA; igualmente, en el caso de existir varios Ensamblados de Interoperabilidad para un mismo componente COM, puede establecerse una referencia manualmente a cualquiera de estos Ensamblados de Interoperabilidad, pero el ambiente de desarrollo siempre utilizará el PIA.

1.4.3.2 Los PIAs de Office 2003

El Office 2003 fue diseñado para el trabajo con .NET, por lo que sus ensamblados de Interoperabilidad Primaria, ya vienen incluidos en el paquete de instalación; sin embargo, si no se realiza una instalación completa del producto, los PIAs no se instalan automáticamente.

En vista de que los PIAs son fundamentales para tener acceso al modelo de objetos de Office, cuando se desarrollan soluciones personalizadas con código administrado para cualquiera de sus aplicaciones (Word, Excel, etc.), es una

¹⁷ Todos los PIAs se instalan en un directorio especial del sistema, al que se accede ejecutando la instrucción *assembly* desde la consola de comandos, todos los ensamblados que se encuentran en este directorio, se dice que están en el Cache Global de Ensamblados (*Global Assembly Cache*).

prioridad revisar si se encuentran instalados todos los PIAs necesarios, en el directorio <drive>:\WINDOWS\assembly.

Hay que recordar, que al desarrollar una solución personalizada de Office, el Visual Studio .NET generará por su cuenta un Ensamblado de Interoperabilidad por cada PIA que no encuentre y que sea necesario para la solución, lo cual resultaría en los problemas de incompatibilidad, anteriormente mencionados.

Un listado completo de los PIAs de Office 2003 se encuentra en el Anexo C, y la guía de instalación en el Anexo D.

1.4.4 LOS CONTENEDORES EN TIEMPO DE EJECUCIÓN

Cuando se utiliza un componente COM desde una aplicación bajo código administrado, el CLR utiliza la información contenida en el Ensamblado de Interoperabilidad del componente, para construir el contenedor en tiempo de ejecución, el cual a su vez, utilizando los metadatos también incluidos en el ensamblado de interoperabilidad, puede crear instancias del objeto COM en tiempo de ejecución; con esto se logra ocultar las diferencias entre los modelos de programación administrado y no administrado.

1.4.4.1 Tipos de Contenedores

El contenedor creado e invocado en tiempo de ejecución, puede ser de dos tipos:

- Si un cliente .NET va a utilizar métodos de un objeto COM, recibe el nombre de "Contenedor al que se puede Llamar en Tiempo de Ejecución" (RCW – *Runtime Callable Wrapper*).
- Si un cliente COM va a utilizar métodos de un objeto .NET, recibe el nombre de "Contenedor COM al que se puede Llamar" (CCW – *COM Callable Wrapper*).

Lógicamente, el tipo más común es el RCW, en vista de que la idea fundamental es seguir utilizando los componentes COM ya escritos, pero siempre existirá la posibilidad de crear un nuevo componente COM no administrado, que pueda

utilizar métodos de un componente .NET administrado, en donde necesariamente el contenedor será de tipo CCW.

En la Ilustración 1-1 se muestra el funcionamiento de los contenedores, como puente entre el código administrado y el no administrado.

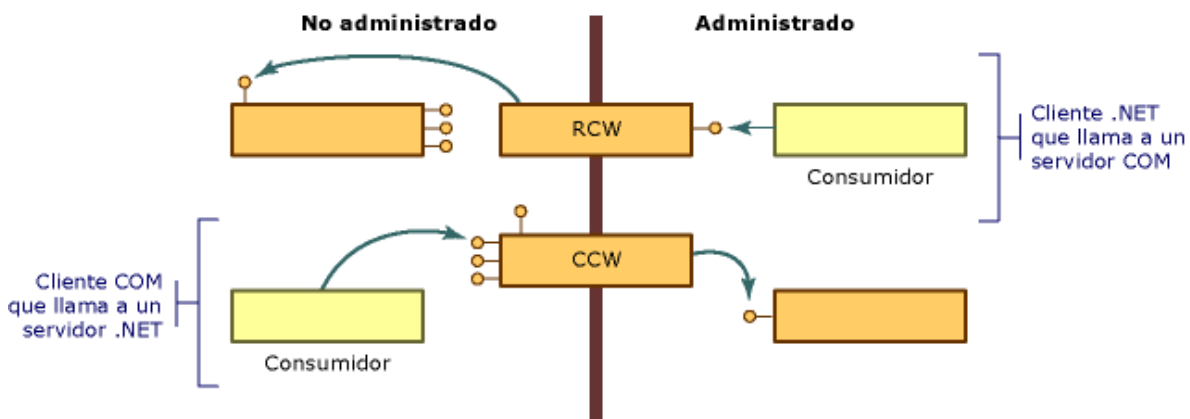


Ilustración 1-1 Contenedores en Tiempo de Ejecución de objetos COM (RCW) o .NET (CCW).¹⁸

1.4.4.2 Relación Servidor – Contenedor – Cliente

El CLR siempre creará un solo contenedor (RCW o CCW) por cada servidor (objeto COM o .NET), y todos los clientes, sin importar cuantos sean, se comunicarán con su servidor por intermedio de dicho contenedor.

En la Ilustración 1-2 se muestra un ejemplo en el que existen: dos objetos COM (servidores) y dos clientes .NET. Cada uno de los objetos COM exponen una interfaz diferente (INew y INewer); uno de los clientes .NET consume sólo la interfaz INewer, mientras que el otro consume las dos interfaces disponibles; sin embargo, solo existe un contenedor RCW por cada uno de los dos servidores COM.

¹⁸ Ilustración 1-1: tomada de la Referencia 52.

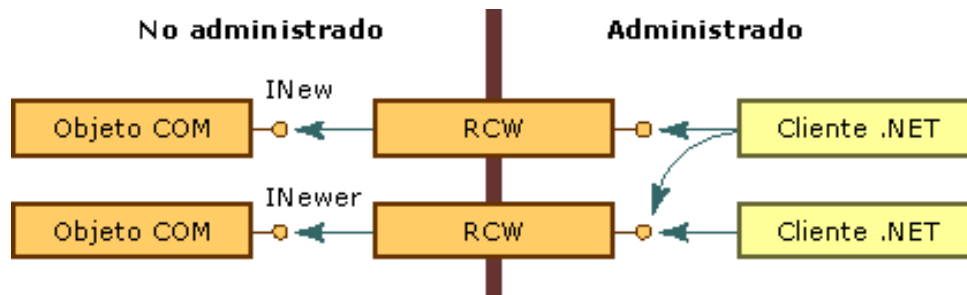


Ilustración 1-2 Representación de la relación unívoca entre contenedor y objeto.¹⁹

1.4.4.3 Los Contenedores y las Interfaces

Un objeto COM adicionalmente a las interfaces que exponen los métodos, propiedades y eventos propios del objeto, posee interfaces especiales tales como la IUnknown²⁰, la IDispatch, la IErrorInfo, etc., que un cliente .NET no entendería.

Los contenedores son los encargados de consumir (contenedor RCW) o de exponer (contenedor CCW) dichas interfaces especiales, de tal forma que en el lado del cliente siempre estén únicamente las interfaces necesarias.

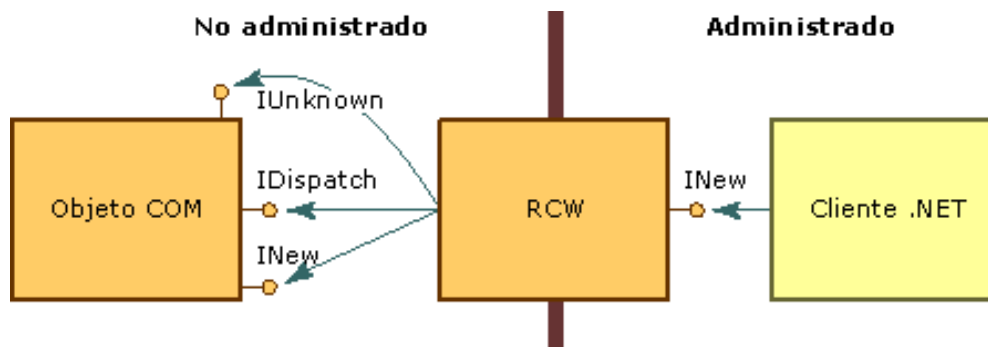


Ilustración 1-3 Interfaces COM y el RCW.²¹

¹⁹ Ilustración 1-2: tomada de la Referencia 51.

²⁰ IUnknown es una interfaz que permite a los clientes conseguir punteros a otras interfaces en un objeto dado, y administrar la existencia del objeto. Todas las otras interfaces COM están heredadas directa o indirectamente de IUnknown [Ref. 45].

²¹ Ilustración 1-3: tomada de la Referencia 49.

En la Ilustración 1-3 se muestra un ejemplo de un objeto COM, que expone las interfaces IUnknown y IDispatch, adicionalmente a la interfaz INew. El contenedor RCW consume las interfaces especiales y expone únicamente la interfaz INew en un formato entendible para el cliente .NET.

El RCW consume las interfaces especiales para ofrecer características que si son propias de la tecnología .NET; por ejemplo, la interfaz IDispatch es utilizada para enlazarse en tiempo de ejecución a objetos COM mediante la Reflexión. Para ver una lista de las interfaces especiales COM que consume y puede consumir un RCW, se puede revisar la Referencia 49.

1.4.4.4 Funciones Principales de los Contenedores

En vista de que los contenedores más comunes son los RCW, a continuación se señalan algunas de sus características principales, las mismas que son aplicables a los contenedores CCW, pero en sentido inverso.

- Crear y enlazar el objeto subyacente COM
- Consumir las interfaces expuestas por el componente COM y exponer las interfaces necesarias en un formato administrado
- Calcular las referencias de interoperabilidad [Sección 1.4.4.4.1]
- Administrar el tiempo de vida de los objetos COM empaquetados
- Traducir los valores HRESULT²² a excepciones .NET²³

1.4.4.4.1 Cálculo de Referencias de Interoperabilidad

Cálculo de referencias de interoperabilidad, es el nombre que recibe el proceso que permite cambiar los tipos de datos diferentes entre COM y .NET, en tiempo

²² Todas las funciones COM (casi en su totalidad) retornan un código de error HRESULT, que es un número de 32 bits que utiliza el bit de signo para representar éxito o falla, y los otros 31 bits restantes son para indicar el código del error, con algunos bits reservados.

²³ Las excepciones son la forma de manejar los errores en .NET; tienen varias ventajas sobre la utilización de códigos de error; tales como: las excepciones siempre se ejecutan, así el programador no haya puesto código explícito para manejar la excepción, no devuelven tan solo un código, sino que indican el tipo de error cometido, los parámetros incorrectos e incluso sugerencias para solucionar el error [Ref. 50].

de ejecución. Este proceso se aplica tanto a los argumentos (parámetros), como a los valores devueltos por los métodos utilizados en una aplicación.

La idea es que tanto los clientes como el servidores, envíen y reciban datos que les sean conocidos, para lo cual todos aquellos valores que son diferentes entre un ambiente y otro, son empaquetados en tipos de datos equivalentes, dejando intactos todos los tipos de datos comunes.

Por ejemplo, si un objeto COM expone un método que recibe un parámetro de tipo 'BSTR', en .NET ese parámetro, luego de hacer el cálculo de referencias de interoperabilidad se verá como un tipo 'STRING', y si por el contrario desde .NET se devuelve un valor 'STRING' el objeto COM recibirá un dato de tipo 'BSTR'.

1.4.4.5 Naturaleza de los Contenedores

En vista de que los contenedores son generados a partir de clases que se encuentran en el CLR, éstos tienen una naturaleza administrada, lo que quiere decir que se rigen a las reglas del código administrado, por lo que el recolector de basura es el encargado de eliminar todos aquellos elementos que los contenedores dejen de utilizar.

CAPÍTULO 2. DESARROLLO DE FORMULARIOS, ETIQUETAS Y DOCUMENTOS INTELIGENTES PARA MICROSOFT OFFICE 2003 UTILIZANDO .NET Y XML

En este capítulo, se señalan algunas alternativas para la creación de formularios electrónicos con Microsoft Office InfoPath, así como también Documentos Inteligentes con Microsoft Word 2003, y Etiquetas Inteligentes que pueden funcionar en varias de las aplicaciones de Office 2003; se mencionan también particularidades sobre:

- El estándar XML
- La utilización del XML dentro del Office 2003
- Los beneficios de la utilización de Servicios Web XML para la transferencia de información
- Las características generales del InfoPath
- La manera en la que un usuario puede interactuar con los elementos XML embebidos en un documento de Word
- La tecnología de Etiquetas Inteligentes dentro de las aplicaciones de Office
- La tecnología de Documentos Inteligentes, propuesta por Microsoft para documentos de Word y Excel
- Algunas herramientas que permiten la creación de soluciones personalizadas de Office con código administrado
- Opciones de seguridad para la implementación de soluciones de Office con código administrado

2.1 XML EN LAS APLICACIONES DE OFFICE

XML son las siglas de, *Extensible Markup Language* o Lenguaje de Marcado eXtensible, y presenta las siguientes ideas generales:

- Es un lenguaje estándar basado en etiquetas que permite describir datos.
- Se basa en texto, no depende de ningún lenguaje de programación.
- Por lo general se utiliza para compartir información, en vista de que facilita la comunicación entre sistemas incompatibles: programas, redes de computadores, estructuras de datos, y aplicaciones corriendo en sistemas operativos diferentes.
- Es altamente extensible, lo que permite que cualquier persona pueda crear las etiquetas personalizadas que se ajusten a sus necesidades.

2.1.1 CARACTERÍSTICAS PRINCIPALES DE UN ARCHIVO XML

- Siempre empieza con una declaración XML.
- Contiene elementos, cada uno de los cuales representa a un dato determinado del archivo.
- Todos los elementos se anidan dentro de un elemento principal, denominado elemento raíz.
- Las etiquetas se escriben dentro de corchetes angulares (< >).
- Los datos se encuentran entre dos etiquetas, una de apertura y una de finalización.
- En ocasiones se presentan elementos vacíos (cuando el dato al que se desea representar no existe), para lo cual habrá una sola etiqueta con un carácter de barra diagonal situado inmediatamente antes del corchete angular de cierre (/ >).
- Los elementos pueden tener atributos, pero no es imprescindible, los atributos se incluyen entre los corchetes angulares de la etiqueta.
- Los archivos XML, por lo general deben estar asociados a por lo menos un espacio de nombres²⁴.

²⁴ Los espacios de nombres son identificadores que cumplen la misma función de los nombres de dominio en Internet; es decir, evitar que existan nombres duplicados. En el caso del XML el espacio de nombres es el que permite diferenciar un elemento de otro.

2.1.2 TIPOS DE ARCHIVOS XML

Archivos de datos XML: Contienen todos los datos etiquetados y estructurados dentro de un archivo de texto con una extensión **.xml** [Ref. 21].

Archivos de esquema XML: Definen la estructura y validan la información de los archivos de datos XML; es decir, determinan el conjunto de etiquetas que puede contener un archivo de datos XML y el tipo de datos que se debe poner entre cada etiqueta, lo más usual es encontrarlos como archivos **.xsd** [Ref.22].

Archivos de transformación XML: Permiten transformar el formato de presentación y/o contenido de archivos de datos XML; a estos archivos por lo general se los reconoce por su extensión **.xslt** [Ref. 23 y 24].

Para poder explotar al máximo las características del XML en el Office 2003, es necesario conocer y estar familiarizado con los fundamentos de creación, estructura, forma de utilización, terminología estándar, etc., de estos tipos de archivos XML, y adicionalmente del denominado lenguaje de ruta XML o XPATH [Ref. 25], que es utilizado para extraer con facilidad los datos de un archivo.

Para ello se recomienda revisar las Referencias indicadas, ya que contienen toda la información necesaria que permitirá entender adecuadamente el XML.

2.1.3 IMPORTANCIA DEL XML EN LAS APLICACIONES DE OFFICE

Entre otras cosas, el XML dentro del Office es importante porque:

- Facilita la creación de documentos completamente estructurados, que separan los datos de su formato de presentación.
- Se puede incluir XML dentro de los documentos ordinarios de uso cotidiano, sin alterar la forma de trabajo de los usuarios finales.
- Las aplicaciones de Office pueden abrir o modificar cualquier archivo XML, sin importar ni la herramienta que se utilizó para crearlo, ni la plataforma bajo la que fue creado.
- Se incluyen tareas de importación y exportación de datos, entre las aplicaciones de Office y otras aplicaciones capaces de manejar XML.

- Permite que los documentos de Office, se conviertan en el punto de llegada y/o partida de los datos dentro de un proceso de intercambio de información corporativa, que puede ser por ejemplo, entre un documento y una base de datos, dos documentos, un documento y una página Web, etc.
- Se pueden seleccionar los datos más importantes de un documento, para etiquetarlos y poder reutilizarlos en cualquier momento que se necesiten.
- Las aplicaciones de Office se convierten en herramientas sumamente fáciles para que los usuarios finales estén en contacto directo con datos XML, sin recurrir en todo momento a la ayuda de los desarrolladores.

2.1.4 LOS ESQUEMAS XML EN EL OFFICE 2003

Microsoft ha desarrollado esquemas XML que pueden definir todas las características de formato y contenido de los documentos de Word 2003 y de Excel 2003; estos esquemas se denominan *WordprocessingML* y *SpreadsheetML* respectivamente.

Adicionalmente a los esquemas predeterminados se pueden adjuntar esquemas personalizados definidos por el usuario para la creación de archivos con extensión **.xml**; la idea es que estos esquemas se ajusten a las necesidades específicas del negocio, ampliando considerablemente las posibilidades de reutilización de la información.

2.1.5 LOS FORMATOS DE ARCHIVO XML EN EL OFFICE 2003

Word 2003, Excel 2003, e InfoPath 2003, soportan como un formato de archivo nativo, adicional a los ya soportados, el **.xml**, lo que permite que estas aplicaciones puedan guardar, abrir y modificar archivos en dicho formato.

Al guardar o abrir un archivo **.xml**, en Word 2003 o Excel 2003, existen tres opciones bien definidas que son: utilizar los esquemas predeterminados (*WordprocessingML*, *SpreadsheetML*), utilizar esquemas personalizados, o aplicar transformaciones XML.

Al utilizar los esquemas predeterminados, se garantiza que al grabar el archivo **.xml**, se almacenará toda la información que tanto Word 2003 como Excel 2003 necesitan saber, para volver a abrir un documento tal y como si hubiera sido guardado en sus formatos nativos predeterminados, **.doc** y **.xls** respectivamente.

Si se utiliza un esquema personalizado, se garantiza que la información será almacenada respetando las etiquetas y reglas de validación que se hayan establecido en dicho esquema, sin agregar ningún dato adicional; sin embargo, cuando se vuelva a abrir este archivo XML con la aplicación en la que fue creado, se podrá visualizar sin problema la información, pero la apariencia del documento no será igual a como fue almacenada.

Y al utilizar una transformación XML se garantiza que el archivo se almacenará o se abrirá con el formato (HTML, texto plano, etc.) y las reglas (tipo de fuente, alineación, etc.) que se indiquen en el archivo de transformación.

2.1.6 LOS DESARROLLADORES, EL XML Y EL OFFICE 2003

Una de las principales ventajas para los desarrolladores, es que se han incluido métodos para el manejo de datos XML dentro de los modelos de objetos de las aplicaciones de Office, lo que permite acceder a la información XML directamente desde un lenguaje de programación.

Los desarrolladores se guían por las etiquetas establecidas en los esquemas, razón por la cual los datos de interés, pueden estar distribuidos por todo el documento e incluso cambiar de posición, no obstante siempre se podrá acceder a ellos mientras permanezcan etiquetados. Esta es una característica muy innovadora, ya que por mucho tiempo los desarrolladores de aplicaciones de Office, debían conocer de antemano la posición exacta de un dato para realizar alguna acción asociada, creando documentos rígidos en los cuales no se podía alterar la posición de ningún elemento, en vista de que siempre existía el riesgo de que la aplicación deje de funcionar correctamente.

El XML, adicionalmente, permite hacer una validación de los tipos de datos en el instante en el que son ingresados, mediante reglas que se establecen en los esquemas; esto permite que el desarrollador no tenga que escribir código asociado a los datos para validarlos, a menos que la validación requiera reglas especiales que no estén relacionadas simplemente con el tipo de dato que se debe ingresar.

Para mayor información de la interacción del XML con el Office 2003, se puede revisar la Referencia 1, que contiene documentación del XML orientada a su integración en las aplicaciones de Office.

2.2 DESARROLLO DE SOLUCIONES UTILIZANDO XML Y WORD 2003

Los documentos de Word tienen estructuras ocultas utilizadas, por ejemplo, para dar formato, indexar elementos (tablas, figuras, referencias, etc.), determinar oraciones y párrafos, etc., y que en general definen todas las características de presentación del documento.

Por años, estas estructuras han pasado desapercibidas al estar embebidas en los archivos **.doc**, pero con la creación del ya mencionado esquema *WordprocessingML*, se puede tener una idea más clara de los elementos estructurales que se encuentran en un documento de Word, ya que este esquema define un conjunto bastante extenso de etiquetas, cada una de las cuales representa a un elemento de las estructuras subyacentes de un documento.

El formato principal de los archivos de Word (**.doc**) es un formato binario, que prácticamente no puede ser abierto con ninguna otra aplicación; sin embargo, ahora con el Word 2003, cualquier archivo que puede ser almacenado como **.doc**, también puede ser almacenado como **.xml**, con la gran ventaja de que el XML es texto fácil de entender con tan solo leerlo, puede ser abierto por múltiples aplicaciones, y además puede ser utilizado o modificado mediante programación,

lo que da la posibilidad a los desarrolladores de crear soluciones personalizadas de Office utilizando las ventajas del XML.

2.2.1 TRABAJO CON ESQUEMAS XML

Aprender las particularidades del *WordprocessingML*, es un caso de estudio bastante interesante, ya que por ejemplo, permitiría escribir aplicaciones, en cualquier lenguaje de programación, que generen archivos de texto XML basados en este esquema, los cuales luego podrían ser abiertos sin problema en Word 2003, respetando todas las condiciones de formato que se hayan especificado.

Esta sección no está enfocada al estudio de este esquema, pero la mejor información para aprender sobre este tema, se la encuentra en el Microsoft Office Word 2003 XML *Software Development Kit* (SDK), cuyo paquete de instalación (wdxmlsdk.msi), se puede obtener de la dirección en la Referencia 26.

2.2.1.1 Biblioteca de Esquemas XML

En Word 2003 se puede tener acceso a una Biblioteca de Esquemas, que proporciona información de los espacios de nombres, y las ubicaciones físicas de un conjunto de esquemas personalizados XML, y de archivos de transformación asociados a dichos esquemas.

Todos los esquemas y archivos de transformación, que se encuentran en la Biblioteca de Esquemas, tienen su correspondiente entrada en el Registro de Windows (a nivel de usuario o de máquina), y en un mismo computador pueden coexistir varias Bibliotecas de Esquemas dependiendo de las configuraciones de los usuarios.



Ilustración 2-1 Cuadro de Diálogo – “Biblioteca de esquemas”.

En la Ilustración 2-1 se muestra el cuadro de diálogo **Biblioteca de esquemas**, para acceder a él desde Word: menú **Herramientas** comando **Plantillas y complementos** → **Esquema XML** → **Biblioteca de esquemas**.

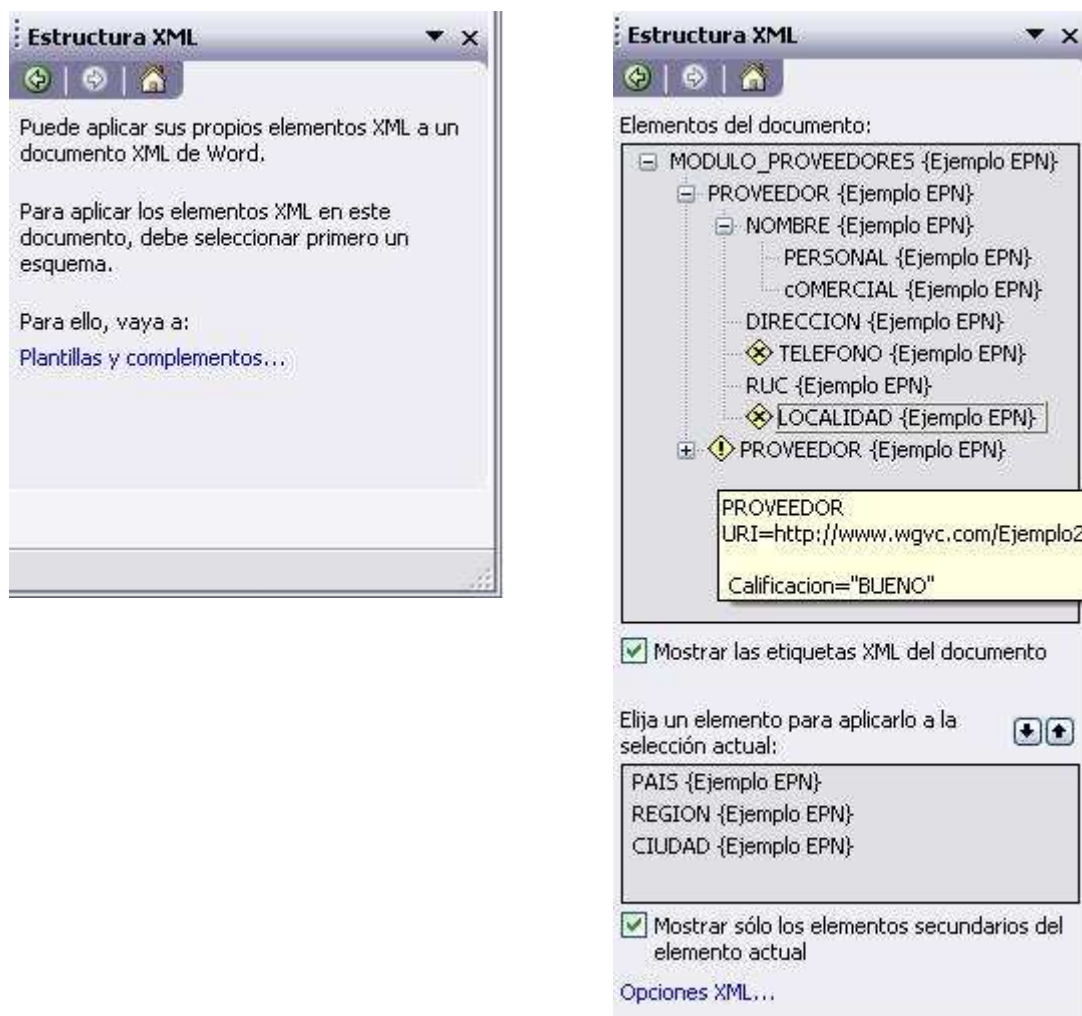
Para entender mejor la función que cumple la Biblioteca de Esquemas, es importante recordar que un archivo de datos XML, posee uno o varios espacios de nombres asociados [Sección2.2.1.1], los cuales indican los esquemas que se deben utilizar para saber las características de cada uno de los elementos del archivo.

El cuadro de diálogo **Biblioteca de esquemas**, se puede manejar intuitivamente; sin embargo, para más información sobre su utilización (agregar o quitar esquemas, asociar archivos de transformación, poner alias, etc.), opciones y elementos característicos, se puede consultar la ayuda de Microsoft Office Word 2003²⁵.

²⁵ Se accede a la ayuda de Word presionando F1 cuando está abierta la aplicación.

2.2.1.2 Panel de Tareas - Estructura XML

Para poder trabajar con los elementos XML de un documento de Word, una excelente ayuda es el panel de tareas **Estructura XML**, al que se accede utilizando en Word 2003 el menú **Ver** comando **Panel de tareas** → **Estructura XML**, y que ofrece una variedad de funciones para trabajar con los esquemas adjuntos a un documento.



Documento sin esquema adjunto

Documento con esquema adjunto

Ilustración 2-2 Panel de tareas – “Estructura XML”.

Si el documento no tiene ningún esquema adjunto, el panel de tareas presentará un mensaje y un hipervínculo que permitirá abrir el cuadro de diálogo **Plantillas y complementos** tal como se muestra en la parte izquierda de la Ilustración 2-2,

caso contrario el panel de tareas se presentará de una manera similar a la mostrada en la parte derecha de la misma figura.

2.2.2 XML Y EL MODELO DE OBJETOS DE WORD 2003

Como ya se mencionó en la Sección 2.1.6, los desarrolladores gracias a las modificaciones que se han realizado a los modelos de objetos de las aplicaciones de Office, pueden utilizar funciones que les permitan manejar los datos XML incluidos en los documentos.

En el caso de Word 2003, se han agregado objetos (con sus respectivos métodos y propiedades), colecciones, enumeraciones, y métodos a ciertos objetos existentes, lo cual permite a los desarrolladores automatizar tareas XML, que se pueden realizar visualmente (abrir o guardar archivos XML, adjuntar esquemas, asignar atributos, etc.), directamente desde un lenguaje de programación.

En la Tabla 2-1 se muestra una lista de los Objetos, Colecciones y Enumeraciones, agregadas en el Modelo de Objetos de Word 2003.

OBJETOS	COLECCIONES	ENUMERACIONES
XMLChildNodeSuggestio	XMLChildNodeSuggestion	WdXMLNodeLevel
XMLNamespace	XMLNamespaces	WdXMLNodeType
XMLNode	XMLNodes	WdXMLSelectionChangeReaso
XMLSchemaReference	XMLSchemaReferences	WdXMLValidationStatus
XSLTransform	XSLTransforms	

Tabla 2-1 Lista de elementos agregados al modelo de objetos de Word.²⁶

Para saber el funcionamiento de cada uno de estos elementos, se puede consultar adicionalmente el modelo de objetos de Word [Anexo A, o Ref. 20], o la documentación incluida en el Microsoft Office Word 2003 XML *Software Development Kit* (SDK) [Ref. 26].

²⁶ La tabla está hecha en base a la información de la Referencia 26, sección *Concepts* → *Developing with XML Documents in Word* → *XML and the Word Object Model*.

2.3 DESARROLLO DE SOLUCIONES UTILIZANDO INFOPATH 2003

En las empresas existen muchos procesos de negocios para recolección y entrega de información que requieren llenar formularios, ya sea de forma manual o electrónica.

Microsoft Office InfoPath 2003 es una aplicación que está diseñada para facilitar en todo sentido, el trabajo que tenga que ver con formularios electrónicos: diseño de formularios, utilización (llenado) o envío de los mismos.

2.3.1 CARACTERÍSTICAS PRINCIPALES DE INFOPATH

En Office 2003 se nota claramente que Microsoft confía mucho en todas las ventajas que puede agregar la utilización del XML en sus aplicaciones, y el InfoPath es una clara muestra de esto, ya que su principal característica es que absolutamente toda la arquitectura de un formulario está basada en archivos XML.

Cuando se habla de InfoPath, se debe tener en consideración, que por un lado están los usuarios de formularios, y por el otro están los diseñadores de formularios, con herramientas completamente diferentes.

Los usuarios de formularios tienen funciones que les permiten combinar o exportar datos de varios formularios en uno solo, exportar datos a Excel, realizar tareas de búsqueda y reemplazo, aplicar formato al texto ingresado, hacer revisión ortográfica, insertar elementos (símbolos, tablas, imágenes, etc.), trabajar con paneles de tareas, y algunas otras funciones que faciliten el proceso de ingreso y visualización de la información.

Y por otro lado los diseñadores, son los que hacen posible la existencia de los formularios, y para ello poseen herramientas que les permiten: utilizar orígenes de datos, trabajar con conexiones de datos, realizar cálculos, validar información, definir reglas, filtrar datos, agregar controles, configurar formatos, y en general

todas las funciones que les faciliten la tarea de diseñar la interfaz que verán los usuarios del formulario.

Los diseñadores diseñan plantillas (archivos **.xsn**), y los usuarios utilizan formularios (archivos **.xml**); todos los formularios tienen asociada una plantilla a la que siempre estarán vinculados, lo que permite que estén actualizados continuamente, ya que cualquier cambio que se haga en las plantillas se verá reflejado en los formularios.

Las plantillas pueden alojarse en una máquina local o remota (compartidas en sitios de la Intranet o del Internet); sin embargo, cuando se utilizan plantillas remotas, el InfoPath crea una copia local, permitiendo trabajar con el formulario aún cuando se pierda la conexión con el sitio de origen de la plantilla, no obstante la copia se encarga de detectar cualquier modificación hecha en la plantilla original y se actualizará automáticamente cuando se restablezca una conexión.

Una gran desventaja, al utilizar formularios de InfoPath, es que se requiere que cada usuario tenga instalado el programa en su computador, o tener acceso a una copia compartida en un servidor.

2.3.2 CARACTERÍSTICAS DE LOS FORMULARIOS DE INFOPATH

- Son dinámicos, interactivos y flexibles
- Poseen una interfaz similar a la de todas las aplicaciones de Office
- Evitan errores en el ingreso de datos, porque se basan en validaciones, y restricciones de formato
- Pueden ser diseñados fácilmente a partir de esquemas XML, datos XML, estructuras de tablas de bases de datos, o cualquier otro origen de datos estructurado
- Permiten múltiples vistas para presentar los datos
- Utilizan conexiones a orígenes de datos externos para el envío y recepción de la información, como bases de datos, o servicios Web XML

- Se pueden abrir, para modificar o agregar datos, todas las veces que sean necesarias, no es imprescindible llenar un formulario de una sola vez
- Para su entrega final pueden ser impresos, enviados como mensajes de correo electrónico, publicados en páginas Web o en portales de documentos compartidos

2.3.3 DISEÑO DE FORMULARIOS DE INFOPATH

En general, el proceso de diseño de un formulario en InfoPath, no es complicado se basa en el sistema WYSIWYG ("*what you see is what you get*"), las herramientas para hacerlo son fáciles de manejar, y están disponibles desde paneles de tareas bien organizados.

En la Ilustración 2-3 se muestra un ejemplo de tres de los paneles de tareas: **Tareas de diseño**, **Diseño** y **Controles**, disponibles para diseñar un formulario de Infopath.

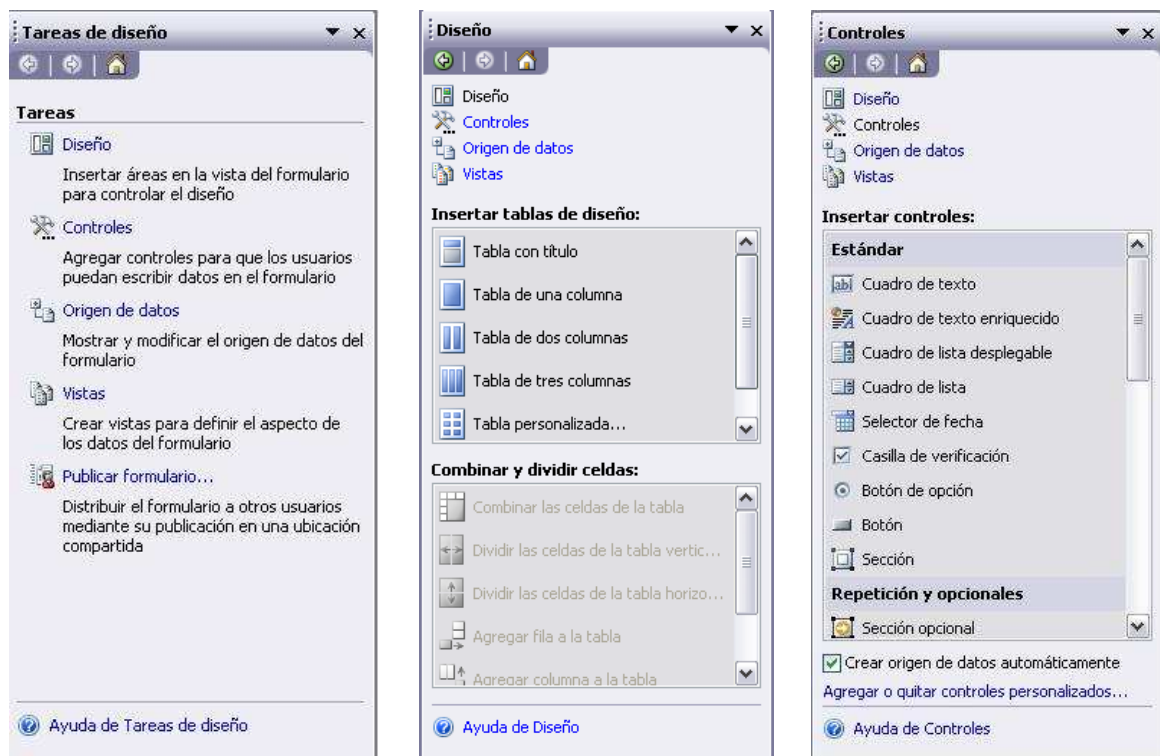


Ilustración 2-3 Paneles de tareas para el diseño de formularios en InfoPath.

En general, un desarrollador con experiencia en el diseño de cualquier tipo de formularios electrónicos, no debe tener ningún problema en adaptarse a las características de diseño que ofrece el InfoPath, e incluso un usuario avanzado puede crear sus propio formularios sin necesidad de escribir código.

Para obtener información sobre como diseñar y utilizar los formularios de InfoPath se puede recurrir a la ayuda, bastante amplia, incluida con la aplicación²⁷.

2.3.4 INFOPATH Y EL .NET 2003

Al igual que las demás aplicaciones de Office, el InfoPath también posee un modelo de objetos; sin embargo, éste no permite realizar procesos de automatización al cien por ciento, lo que significa que no es posible crear una instancia de un formulario de InfoPath directamente desde código de programación, pero se pueden realizar tareas como abrir, registrar, o cerrar formularios.

En un principio, InfoPath no soportaba ningún otro tipo de personalización que no sea con lenguajes de *script*, pero desde el *Service Pack 1* (paquete de archivos de actualización) para InfoPath 2003, se incluyó el soporte para la personalización con .NET y código administrado.

En general, todas las plantillas de InfoPath contienen un conjunto de archivos que trabajan simultáneamente para definir cada uno de los aspectos sobre la manera en que se mira y comporta un formulario, casi todos estos archivos son XML, en donde se puede notar claramente la fuerte interrelación entre el InfoPath y el XML²⁸.

²⁷ Para ingresar al panel de tareas "Ayuda de InfoPath", se debe presionar F1 cuando la aplicación está abierta.

²⁸ InfoPath es la única aplicación de Office que puede generar archivos de esquema, transformaciones, y datos, en base a lo que el desarrollador configura visualmente. Access también permite generar archivos de datos y esquemas XML, pero no archivos de transformación de formato.

Nombre	Extensión	Descripción
Definición del formulario	.xsf	Archivo generado en InfoPath que contiene información sobre todos los demás archivos y componentes utilizados en una plantilla de formularios, incluyendo personalizaciones de interfaz de usuario, esquemas XML, vistas, código de formulario y opciones de implementación. Este archivo constituye el manifiesto del formulario.
Esquema XML	.xsd	Archivos de Esquema XML que se utilizan para restringir y validar los archivos de los documentos XML subyacentes de un formulario.
Vista	.xsl	Archivos lógicos de presentación que se utilizan para presentar, ver y transformar los datos contenidos en los archivos de los documentos XML del formulario.
Plantilla XML	.xml	Archivo .xml que contiene los datos predeterminados que se muestran en una vista al crear un nuevo formulario.
Plantilla del componente de XML	.xct	Representaciones de archivos .xml de los controles de edición utilizados para crear y rellenar un formulario.
Presentación	.htm, .gif, .bmp y otras	Archivos utilizados en combinación con los de vista para crear una interfaz de usuario personalizada.
Código de formulario	.dll	Ensamblado de código administrado compilado que contiene la lógica de programación utilizada para implementar un comportamiento de edición específico, validación de datos, controladores de eventos, control del flujo de datos y otras lógicas empresariales personalizadas.
Archivos binarios	.dll, .exe	Cualquier componente personalizado que proporcione funcionalidad adicional a la lógica empresarial. Normalmente, se trata de ensamblados que se han agregado al proyecto y a los que se hace referencia en el entorno Microsoft Visual Studio® .NET durante el desarrollo de la plantilla de formularios.
Plantilla de formulario	.xsn	Formato de archivo comprimido (.cab) que comprime todos los archivos del formulario en uno solo.

Tabla 2-2 Archivos que componen una plantilla de InfoPath.²⁹

Dentro de este conjunto de archivos están los archivos de código (administrado o no administrado), en la Tabla 2-2 se muestra una lista de todos los archivos que componen una plantilla de formularios de InfoPath utilizando código administrado.

Por lo general, todos los archivos que conforman una plantilla se entregan empaquetados en un archivo de **Plantilla de formulario** [Tabla 2-2], y para tener acceso a ellos, se debe seleccionar la opción **Extraer archivos del formulario**, del menú **Archivo** (InfoPath en modo de diseño); sin embargo, existe un utilitario denominado *Kit* de Herramientas de InfoPath 2003 para Visual Studio .NET

²⁹ Esta tabla está tomada de la ayuda de MSDN, artículo: "Arquitectura de las plantillas de formularios con código administrado", cuya dirección de ayuda se encuentra en la Referencia 46.

[Sección 2.7.1], que permite visualizar todos estos archivos dentro del directorio de un proyecto cuando se utiliza el Visual Studio .NET para el desarrollo.

2.4 UTILIZACIÓN DE SERVICIOS WEB XML DENTRO DEL OFFICE 2003

La importancia de la utilización de los Servicios Web en Office 2003³⁰, radica en que sus aplicaciones, pueden ser clientes directos de un Servicio Web XML. Esto entre otras cosas, ayuda para que el desarrollo de soluciones personalizadas de Office se beneficie significativamente, permitiendo: buscar información en fuentes de datos diversas, tener acceso a hardware de propósito especial (telecontrol, telemetría), almacenar información en bases de datos remotas, etc., y todo bajo una misma interfaz con la cual el usuario está familiarizado.

Utilizar un Servicio Web XML para desarrollar una aplicación personalizada de Office, es bastante similar a utilizar un objeto COM, simplemente se configuran las referencias a un determinado Servicio Web XML, y luego se utilizan los métodos expuestos por dicho servicio.

2.4.1 PROTOCOLOS Y ESTÁNDARES ASOCIADOS CON LOS SERVICIOS WEB XML

SOAP – Protocolo de Acceso Simple a Objetos:³¹ Los datos XML enviados o recibidos por un Servicio Web XML son empaquetados utilizando el protocolo SOAP, razón por la cual, tanto el servidor como los clientes, deben tener instalada

³⁰ También se pueden consumir Servicios Web XML, desde aplicaciones personalizadas de Office XP, instalando el *Web Services Toolkit* de Microsoft Office XP, pero el caso de estudio de este proyecto es la versión de Office 2003.

³¹ SOAP (*Simple Object Access Protocol*): Es un protocolo basado en mensajes XML, para intercambiar información entre aplicaciones cliente y Servicios Web XML. Actualmente, SOAP se utiliza más ampliamente en conjunción con el HTTP (*Hypertext Transport Protocol*), como su protocolo de transporte; sin embargo, SOAP es un protocolo extensible, que puede utilizar otros protocolos si los necesita.

la librería *Microsoft Soap Type Library* (MSSOAP1.dll o posterior)³², que es la encargada de proporcionar el soporte para este protocolo.

UDDI - Descripción, Descubrimiento e Integración Universales.³³ Mediante UDDI los Servicios Web XML pueden exponer los métodos que ofrecen, así como también cualquier información descriptiva que se proporcione sobre los mismos, incluso muchas veces se ofrecen páginas de prueba para los métodos ofrecidos.

WSDL - Lenguaje de Descripción de Servicios Web:³⁴ Por intermedio de este lenguaje, se describen los métodos expuestos por el servicio (ej. nombres, si requieren parámetros o no, si retornan datos o no, etc.) comprueba que todos los datos, enviados y recibidos sean correctos, únicamente se puede trabajar con tipos de datos XSD (en forma independiente, o como arreglos) tales como: *xsd:boolean*, *xsd:unsignedbyte*, *xsd:dateTime*, *xsd:double*, *xsd:short*, *xsd:int*, *xsd:float*, *xsd:string*, y *xsd:anyType*, esto quiere decir que en el caso de Office, por ejemplo, no es posible enviar ni recibir ningún tipo de objeto de los modelos de objetos (ej. *Application*, *Document*, *Range*).

³² *Microsoft Soap Type Library*: es requerida sólo por los sistemas Windows, viene incluida en las versiones de los sistemas operativos actuales (Windows XP, Windows 2003), para el caso de otros sistemas operativos de Windows, se puede instalar la librería manualmente, mediante los *service packs* u otras aplicaciones que la contengan (ej: *Soap Toolkit 2.0 redistributable files*), y en el caso de otros sistemas operativos debería consultarse su equivalente y la forma de instalación.

³³ UDDI - *Universal Description, Discovery, and Integration*: Marco de trabajo independiente de la plataforma, que funciona como directorio (similar a una agenda de teléfonos) y proporciona un modo de encontrar y registrar servicios Web en Internet. La especificación UDDI llama a tres elementos: las páginas blancas, que proporcionan información de contacto comercial; las páginas amarillas, que organizan los Servicios Web en categorías (por ejemplo, servicios de autorización de tarjetas de crédito) y las páginas verdes, que proporcionan información técnica detallada acerca de servicios individuales [Ref. 44].

³⁴ WSDL - *Web Services Description Language*: es un formato XML que describe a los Servicios Web XML, definiendo la manera en la que un cliente debe comunicarse con un servicio, de tal forma que siempre se envíen las peticiones en un formato SOAP bien estructurado.

2.5 VISIÓN GENERAL DE ETIQUETAS Y DOCUMENTOS INTELIGENTES

Las Etiquetas y Documentos Inteligentes son lo más avanzado que existe en el campo de la personalización de las aplicaciones de Office, y permite que los desarrolladores puedan ofrecer a los usuarios funciones que no se podían desarrollar antes.

2.5.1 ETIQUETAS INTELIGENTES

Las Etiquetas Inteligentes o *Smart Tags*, son una tecnología propietaria de Microsoft, que apareció en las aplicaciones de Word, Excel y Outlook de Office XP; posteriormente, se incorporó en el Microsoft Internet Explorer³⁵, y en la actualidad en el Office 2003, también se brinda soporte en Access 2003, PowerPoint 2003, Menú Persona³⁶, y panel de tareas Referencia.

El objetivo principal de esta tecnología, es permitir que las aplicaciones que la utilizan, puedan reaccionar al contenido del texto mientras se lo está escribiendo.

Al realizar la corrección automática (ortográfica o gramatical) que ofrecen las aplicaciones de Office, se analiza el texto mientras se lo ingresa, si se encuentra algún error, en ocasiones es corregido inmediatamente, pero por lo general se resalta el error subrayando el texto (líneas rojas ortografía, verdes gramática), de tal manera que al hacer un *clic* derecho sobre el mismo, se muestra un menú de opciones que permite realizar varias acciones relacionadas con el texto de error; tales como, cambiar de idioma, auto corregir, agregar al diccionario, etc.

³⁵ Las versiones de Microsoft Office XP y 2003 instalan un componente adicional que habilita el soporte para Etiquetas Inteligentes en Microsoft Internet Explorer ya que éste último no soporta nativamente esta tecnología.

³⁶ El Menú Persona es un menú especial asociado con los nombres de las personas en Office 2003, tiene bastante utilidad para la nueva característica de espacio de trabajo compartido integrada en Office 2003, en donde varios usuarios pueden trabajar sobre el mismo documento, y en este caso el Menú Persona muestra la información de estado y una lista acciones apropiadas para los nombres, como por ejemplo enviar correo electrónico.

Y justamente esta función es la que cumplen las Etiquetas Inteligentes, pero en forma personalizada, en donde el desarrollador es quien configura, en base a los requerimientos de los usuarios, que patrones, palabras o frases, se deben reconocer, y que acciones se pueden realizar con relación a las mismas.

2.5.1.1 Características Principales de la Tecnología de Etiquetas Inteligentes

Plataforma de Servicios Hypermedia:³⁷ Al utilizar Etiquetas Inteligentes, se pueden integrar al entorno de trabajo de las aplicaciones algunas opciones que permiten la utilización de recursos locales y enlaces a sitios de Internet, que ayudan a los usuarios a estar mejor informados y comunicados con las personas relacionadas al negocio.

Fácil configuración: Los usuarios pueden especificar las etiquetas que desean activar en sus computadores personales, y los proveedores de contenido pueden determinar cuando estará disponible o no una Etiqueta Inteligente.

Reacción al contexto: Las Etiquetas Inteligentes responden al contenido relacionado con el contexto de los documentos de trabajo de los usuarios, esto significa que cualquier cadena de texto ingresada por el usuario, que coincida con listas o patrones de cadenas incluidas en una Etiqueta Inteligente, puede ser reconocida.

Flexibilidad en la obtención de datos: La lista de cadenas de texto que se pueden reconocer con una Etiqueta Inteligente, tiene una naturaleza estática o dinámica, lo que significa que el desarrollador, a más de poder incluir listas de términos relativamente cortas y que es muy difícil que se modifiquen, también puede enlazarse con fuentes de datos externas que varían constantemente, tales como: bases de datos, Servicios Web XML, archivos XML, o cualquier otro tipo de archivo estructurado.

³⁷ Hypermedia es la utilización de datos, texto gráficos, sonido y video como elementos en un sistema de hipertexto. Todas las formas de información están enlazadas entre sí permitiendo al usuario moverse de la una a la otra [Ref. 31].

Ejecución inmediata: Las acciones que se ofrecen por intermedio de una Etiqueta Inteligente se ejecutan inmediatamente, en vista de que el usuario no necesita abandonar la aplicación en la que están trabajando, para realizar tareas que incluso pueden implicar el acceso a datos corporativos; es mucho más sencillo utilizar una lista de opciones ofrecida por la Etiqueta Inteligente, que abrir otra aplicación para ejecutar una acción, lo que se ve reflejado en eficiencia y rapidez de respuesta.

Plataforma de Servicios Windows: La tecnología de Etiquetas Inteligentes, no es abierta y sólo puede ofrecer sus servicios en aplicaciones propietarias de Microsoft que corran bajo una plataforma Windows.

2.5.2 DOCUMENTOS INTELIGENTES

Antes de hablar sobre la tecnología de Documentos Inteligentes, es importante mencionar que no es una exclusividad de Microsoft; las dos principales empresas que se disputan este mercado son: ADOBE que permite crear Documentos Inteligentes en su formato estándar **.pdf** (*Portable Document Format*), y Microsoft que permite hacerlo en sus formatos estándar **.doc** y **.xls** de Word y Excel respectivamente.³⁸

En general, un Documento Inteligente es una solución personalizada basada en documentos, que explota las características del XML para separar el formato de la información contenida en los documentos, esto permite que cualquier persona pueda entender, leer, modificar, etc., un documento, y que además una porción de código compilada, pueda realizar acciones acorde a la lógica del negocio y en función de los datos implícitos en el documento.

³⁸ Adobe y Microsoft lanzaron al mercado sus productos Adobe Acrobat 6.0, y Microsoft Office System 2003, ambos con la capacidad de crear Documentos Inteligentes en el año 2003. Este proyecto de titulación no abarca la creación de Documentos Inteligentes con Adobe, pero si se quiere incursionar en el tema, se puede comenzar revisando las características de la herramienta *Adobe LiveCycle Designer*, que es la que permite desarrollar este tipo de soluciones con Adobe.

2.5.2.1 Características Principales de la Tecnología de Documentos Inteligentes

División por secciones XML: Todo Documento Inteligente tiene asociada una estructura XML, mediante la cual se pueden determinar secciones (delimitadas con etiquetas XML), cada una de las cuales tiene asociada una acción, tal como: presentar información de ayuda (gráficos, texto, etc.), agregar datos auxiliares (fragmentos de documentos, resultados de búsquedas, etc.), manipular controles (cuadros de texto, botones, etc.), que los usuarios pueden ejecutar a través de paneles de tareas personalizados. Las acciones son variables, dependen de la sección en la que se encuentre el usuario.

Sensibilidad al contexto del documento: Al igual que en las Etiquetas Inteligentes, las acciones también dependen del contexto de la información ingresada, pero en este caso relacionado con el contenido del documento. Mientras las Etiquetas Inteligentes trabajan aisladamente (no conocen el contenido del documento), los Documentos Inteligentes pueden trabajar con todos los elementos de la estructura XML del documento; esto quiere decir, que si por ejemplo, se ingresa el nombre de varias personas en un documento de Word, con una Etiqueta Inteligente se podrían realizar acciones independientes sobre cada uno de ellos, mientras que con un Documento Inteligente, se podrían establecer relaciones entre los nombres y realizar acciones conjuntas.

Trabajo con fuentes de datos: Si bien es cierto que los Documentos Inteligentes se pueden utilizar a nivel personal, toda su potencialidad se ve reflejada en ambientes corporativos, por la gran facilidad que presentan para interactuar con fuentes de datos, internas o externas a la organización. Los desarrolladores pueden crear documentos que tengan una comunicación en dos direcciones con diversas fuentes de datos, permitiendo ubicar automáticamente datos devueltos de una consulta en lugares específicos del documento, o tomar la información ingresada por el usuario para agregarla a una fuente de datos corporativa.

Información estructurada y reutilizable: El contenido XML de un Documento Inteligente, permite que toda la información relevante que se ingresa en un documento esté completamente estructurada, lo que facilita la creación de

fuentes de datos XML concisas y reutilizables, que se ajusten a determinadas reglas predefinidas, minimizando la posibilidad de errores que pueden cometer los usuarios.

Fácil distribución y actualización: Los Documentos Inteligentes tienen soporte para utilizar métodos de distribución centralizada desde un servidor, lo que significa que los administradores no tienen que configurar todos los equipos de una organización para instalar una solución. Se puede trabajar con los Documentos Inteligentes de una manera similar a los formularios de InfoPath, en base a plantillas publicadas en sitios de la red, permitiendo actualizar la lógica y los componentes de un Documento Inteligente fácilmente.

2.5.2.2 Elementos característicos de los Documentos Inteligentes de Microsoft

Esquema XML: Es Necesario que todos los Documentos Inteligentes tengan asociado un esquema XML, que permitirá determinar los elementos y atributos XML, que aparecerán dentro de un documento y sobre los cuales se basará la lógica del mismo.

Paquete de Expansión XML: Es una colección de archivos que incluye: código de implementación compilado en archivos **.dll**, archivos de transformación **.xsl**, archivos **.xml**, archivos de imagen (ej. **.bmp**, **.jpg**), o cualquier otro tipo de archivo que sea necesario para el correcto funcionamiento de un Documento Inteligente. El Paquete de Expansión se genera en base a los requerimientos del Código de Implementación.

Archivo de Manifiesto: Es un archivo XML que describe los aspectos mas importantes de un Documento Inteligente, tales como el nombre del Paquete de Expansión, la ubicación física de todos los archivos que componen el Paquete de Expansión, la frecuencia de búsqueda de actualizaciones, etc., también puede incluir la información del Esquema de la Solución; actúa como un elemento central que contiene la información de todas las partes que conforman el Documento Inteligente.

Cuando se realiza una instalación o actualización por red en un ambiente corporativo, el manifiesto es el pilar fundamental que permitirá que un computador pueda descargar los archivos necesarios, desde alguna ubicación compartida del servidor.

Panel de Tareas Acciones del Documento: Los Documentos Inteligentes pueden tener implícitos: menús personalizados, botones específicos, barras de tareas, etc., pero la característica propia de este tipo de soluciones es la utilización de un panel de tareas personalizado (panel de tareas **Acciones del documento**), dinámico y completamente programable que puede incluir ayudas, controles, y gráficos, para que el usuario no tenga ningún problema en su utilización.

Código de Implementación: Existe mucha flexibilidad para desarrollar el código que contendrá la lógica del Documento Inteligente, pudiendo utilizar varios lenguajes de programación de naturaleza administrada y no administrada (Visual Basic 6.0, Visual Basic .NET, C#, o C++)³⁹; este código actuará directamente sobre los elementos XML que se encontrarán dispersos por todo el documento.

2.6 DESARROLLO DE ETIQUETAS Y DOCUMENTOS INTELIGENTES

2.6.1 DESARROLLO DE ETIQUETAS INTELIGENTES

Los desarrolladores tienen dos opciones bien definidas para crear Etiquetas Inteligentes, la primera es utilizar un archivo XML, y la segunda es mediante un complemento de software (archivo DLL), administrado o no administrado.⁴⁰

³⁹ Hay que recordar que los modelos de objetos de Office 2003, poseen métodos para manipular el contenido XML que se puede incluir dentro de un documento, lo que hace posible crear este tipo de soluciones incluso desde lenguajes de programación no administrados.

⁴⁰ Sólo a partir del Office 2003 se incluyó el soporte para el desarrollo de Etiquetas Inteligentes con código administrado; en Office XP típicamente se utiliza Visual Basic, C++ (no administrado).

Cada uno de estos dos métodos tiene sus ventajas y desventajas, y el desarrollador es quien debe decidir el más adecuado según el propósito, sin querer decir con esto que los dos métodos son completamente equivalentes; al utilizar un archivo XML el proceso será muy fácil y rápido, pero con muchas limitantes de funcionalidad en comparación con un archivo DLL.

2.6.1.1 El Microsoft Office Smart Tag List para Etiquetas Inteligentes

Cuando se crean Etiquetas Inteligentes utilizando archivos XML, lo que en realidad se está especificando es una lista que contiene los términos o patrones⁴¹ que se pueden reconocer, y las acciones que los usuarios pueden ejecutar.

Estos archivos XML deben ser creados bajo las especificaciones del esquema *Microsoft Office Smart Tag List* (MOSTL) definido por Microsoft. En el esquema se incluyen todos los elementos necesarios para definir una Etiqueta Inteligente, entre los que se puede mencionar: nombre, idioma de reconocimiento, descripción, sitio de información para el desarrollador, sitio de descarga, sitio de actualización, versión, frecuencia de actualización, tipo, lista de términos, ubicación de archivos externos con listas de términos, etc.

2.6.1.2 Desarrollo de Etiquetas Inteligentes utilizando Código

Las Etiquetas Inteligentes creadas con un lenguaje de programación no utilizan el *Microsoft Office Smart Tag List*, para reconocer y ejecutar acciones, ya que el desarrollador es quien implementa todo lo necesario para que el Office sepa que hacer cuando encuentra una de estas etiquetas.

Este método es mucho más complejo que la utilización de archivos XML, pero las posibilidades de personalización se amplían notablemente. Al utilizar código, el desarrollador tiene a su disposición todos los recursos que necesite, para hacer lo

⁴¹ El reconocimiento de patrones en las etiquetas inteligentes definidas por un archivo XML, solo es posible en el Office 2003, ya en Office XP, este tipo de reconocimiento solo se podía realizar mediante código.

que su habilidad y conocimientos le permitan (enlaces a bases de datos, consumo de servicios Web, creación de cuadros de diálogo, etc.).

En sí, la creación de Etiquetas Inteligentes, utilizando un lenguaje de programación, es un proceso comparativamente largo, y la idea fundamental es crear dos clases (pueden estar implementadas en un mismo DLL, o en DLLs independientes), una de reconocimiento y una de acción,

La clase de Reconocimiento: Es la que se encarga de asociar las cadenas de texto escritas en un documento a un Tipo de Etiqueta Inteligente; el conjunto de métodos y propiedades necesarias que se deben implementar, para que las aplicaciones de Office interactúen con un objeto de reconocimiento, están definidas en dos interfaces denominadas: ISmartTagRecognizer o ISmartTagRecognizer2.

La clase de Acción: Es la que implementa las acciones invocadas por el usuario, con relación a un determinado Tipo de Etiqueta Inteligente, y el conjunto de métodos y propiedades que se deben implementar en esta clase están definidas en las interfaces ISmartTagAction o ISmarTagAction2⁴².

El proceso de instalación de una Etiqueta Inteligente, creada mediante código, requiere que los archivos DLLs estén registrados en el Registro de Windows por usuario o por máquina⁴³, con lo cual las aplicaciones de Office conocerán de la existencia de nuevas Etiquetas Inteligentes y podrán ser utilizadas.

2.6.2 DESARROLLO DE DOCUMENTOS INTELIGENTES

Al igual que en las Etiquetas Inteligentes, también existen dos métodos para crear Documentos Inteligentes, utilizando un archivo XML sin escribir nada de código, o utilizando código para crear un archivo DLL (administrado o no administrado).

⁴² Tanto la interfaz ISmartTagRecognizer2, como la ISmarTagAction2 están disponibles a partir de la versión de Office 2003 e implementan varias funciones adicionales.

⁴³ El proceso de registro por máquina es una posibilidad que se incluyó con Office 2003, en XP solo se podía realizar registros por usuario.

En cualquiera de los dos casos, los pasos a seguir para crear un Documento Inteligente son los siguientes:

1. Adjuntar un esquema XML al documento, y asociar los elementos XML con las secciones del documento sobre las cuales se ejecutarán las acciones.
2. Utilizar un lenguaje de programación o el esquema MOSTL para escribir código o XML, que muestre los controles en el panel de tareas **Acciones del documento** e implemente las acciones que se ejecutarán cuando el usuario interactúe con los controles.
3. Almacenar el código y todos los archivos asociados que utilizará el Documento Inteligente en directorios bien identificados, o en el caso de una implementación corporativa, en una ubicación de red de un servidor de archivos o en un servidor Web a la que los usuarios puedan acceder normalmente con los permisos respectivos.
4. Crear el archivo de manifiesto XML, que haga referencia a todos los archivos utilizados por la solución.
5. Adjuntar al documento un archivo de manifiesto XML.
6. Utilizar el documento, o en caso de una implementación corporativa, la opción ideal es distribuir el documento como una plantilla. Cuando un usuario cree una instancia del documento basado en la plantilla, el Documento Inteligente y todos los archivos necesarios serán descargados y registrados localmente en cualquier computador, sin ninguna intervención del usuario.

2.6.2.1 El Microsoft Office *Smart Tag List* para Documentos Inteligentes

El esquema MOSTL (versión para Office 2003), también permite la creación de Documentos Inteligentes que están limitados a tareas relativamente simples pero igualmente útiles, tales como: mostrar títulos, texto de ayuda, y fragmentos de documento, que se podrán insertar en una sección del documento delimitada por un elemento XML; abrir páginas Web mediante la utilización de hipervínculos, o botones (sólo para ejecutar acciones relacionadas con hipervínculos).

En este caso, el archivo XML ya no contendrá una lista de términos para reconocer, sino que mas bien indicará las acciones que se deben realizar una vez

que el usuario ingrese a una determinada sección del documento, definida por los elementos del esquema personalizado adjunto al mismo.

2.6.2.2 Desarrollo de Documentos Inteligentes utilizando Código

Para desarrollar un Documento Inteligente se debe utilizar el modelo de objetos de Documentos Inteligentes, en este caso solo se debe crear una clase que se encarga de la implementación de acciones y la asociación con los elementos XML embebidos en el documento. Todos los métodos y propiedades (17 métodos y 8 propiedades)⁴⁴ necesarios para la implementación, están definidos por la interfaz *ISmartDocument*.

Los Documentos Inteligentes también se pueden desarrollar tanto con código no administrado, como con código administrado. Para utilizar código administrado, es necesario contar con el PIA, que resulta ser el mismo utilizado para el desarrollo de Etiquetas Inteligentes (en su versión más actual).⁴⁵

En la Ilustración 2-4 se muestra la secuencia de pasos que una aplicación anfitrión (Word o Excel), sigue cuando carga un DLL que implementa una solución de Documento Inteligente, lo cual es importante saber como Referencia para implementar los métodos y propiedades respectivos.

⁴⁴ La interfaz *ISmartDocument* requiere que todos los métodos y propiedades estén presentes en el código sin importar que contengan o no una implementación.

⁴⁵ Tanto las interfaces *ISmartTagRecognizer2*, *ISmartTagAction2* e *ISmartDocument*, están definidas dentro de la librería de tipos *Microsoft Smart Tags 2.0*, por esta razón se utiliza el mismo Ensamblado de Interoperabilidad primaria para las Etiquetas Inteligentes y para los Documentos Inteligentes.

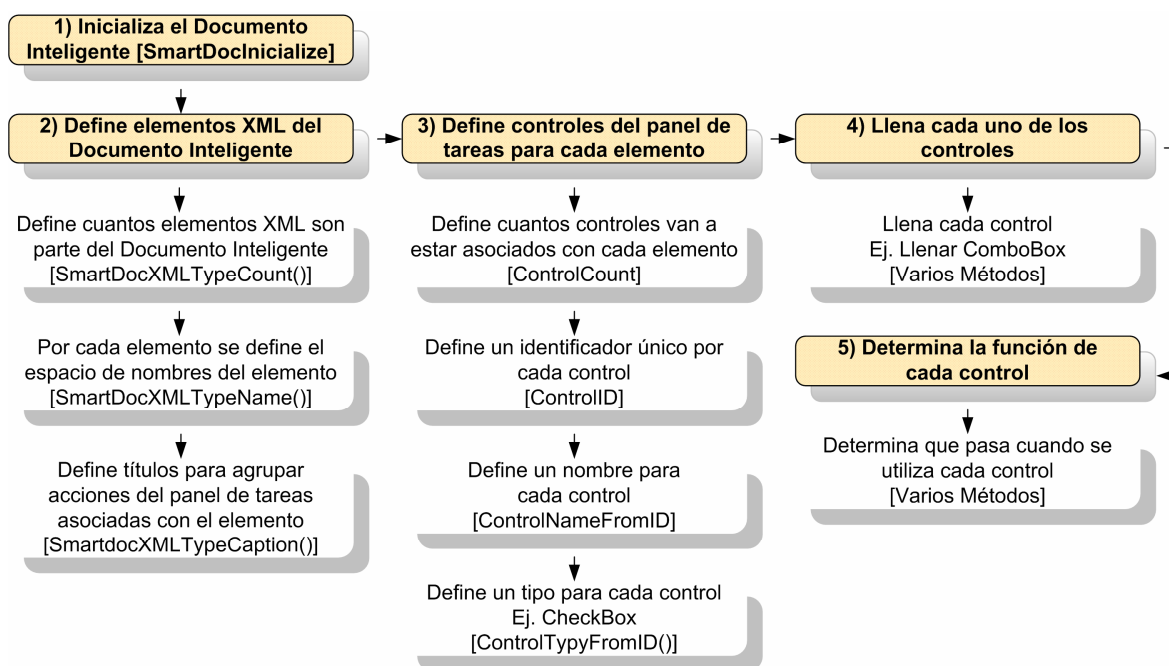


Ilustración 2-4 Secuencia de pasos que sigue una aplicación anfitrión cuando carga una DLL que define un Documento Inteligente.⁴⁶

En el momento de desarrollo, si bien es cierto que todos los puntos de implementación son importantes, se puede decir que lo fundamental y que constituye el pilar para el correcto funcionamiento de un Documento Inteligente, es la definición de los Nombres de Tipo [Ilustración 2-4].

Los Nombres de Tipo deben ser únicos, y para ello se utiliza el espacio de nombres definido por el esquema XML asociado al documento, seguido por un signo de número (#), y por el nombre del elemento XML; por ejemplo, para el esquema MOSTL, el espacio de nombres es:

<http://schemas.microsoft.com/office/smarttags/2003/mostl>

Este esquema contiene un elemento llamado '*caption*', y su Nombre de Tipo será:

<http://schemas.microsoft.com/office/smarttags/2003/mostl#caption>

⁴⁶ Ilustración 2-4: modificación basada en una gráfica de la Referencia 36, fuente en inglés.

2.7 HERRAMIENTAS PARA EL DESARROLLO

Cuando se trata de desarrollar aplicaciones personalizadas de Office con código administrado, la herramienta básica es el Microsoft Visual Studio .NET 2003; sin embargo, existen algunas otras herramientas adicionales diseñadas con un propósito específico, que pueden ser instaladas para facilitar el trabajo del desarrollador.

2.7.1 KIT DE HERRAMIENTAS DE INFOPATH 2003 PARA VISUAL STUDIO .NET

Para desarrollar aplicaciones de InfoPath con código administrado, existe el *Kit* de herramientas de InfoPath 2003, el mismo que permite desarrollar plantillas utilizando el entorno de desarrollo de Visual Studio .NET 2003, aprovechando todas las ventajas del *Framework* .NET.

Con este *Kit* viene incluida la documentación y las características de integración que permiten crear, depurar y generar proyectos de InfoPath con lenguajes propios de la tecnología .NET (Visual C# o Visual Basic .NET); el diseño del formulario se hace directamente desde InfoPath, pero el código de automatización se escribe en Visual Studio .NET.

El proyecto generado contendrá todo el código de inicialización necesario, incluyendo las Referencias a los ensamblados de interoperabilidad, con lo cual el desarrollador estará listo para comenzar a desarrollar la solución. El archivo que instala este *Kit* de herramientas se denomina IPVSPROJ.exe⁴⁷.

Todo lo referente a la utilización de esta herramienta, está descrito en la documentación que viene incluida con la misma, que entre otras cosas contiene:

- Arquitectura de las plantillas de formularios con código administrado
- Los ensamblados de interoperabilidad de InfoPath

⁴⁷ Este *Kit* requiere tener instalado el Visual Studio 2003, y el Microsoft Office InfoPath 2003 *Service pack* 1 o posterior, y se lo puede descargar de la Referencia 32.

- Facilidades para crear proyectos de InfoPath desde Visual Studio .NET 2003
- Facilidades para agregar controladores de eventos
- Facilidades para depurar proyectos de InfoPath
- Facilidades para implementar proyectos de InfoPath

Adicionalmente, para conocer mas detalles sobre como desarrollar aplicaciones de InfoPath con código no administrado se pueden consultar las siguientes Referencias:

- El Microsoft Office InfoPath 2003 *Software Development Kit* (SDK) [Ref. 33], que contiene:
 - Formularios de ejemplo
 - Herramientas para el desarrollo y la implementación de formularios
 - Macros que ayudan en el proceso de desarrollo con código administrado
 - Librerías de código muy bien documentadas
 - Documentación que incluso incluye una pequeña reseña sobre la tecnología que hace posible el funcionamiento de los formularios de InfoPath
- La ayuda del InfoPath en su sección Referencia del Desarrollador de InfoPath, que ofrece una guía sobre:
 - Formularios de ejemplo
 - Validación de datos
 - Eventos
 - Edición estructural
 - Interfaz de usuario
 - Integración en Visual Studio .NET
 - Mejoras del esquema XML

Hay que indicar que en ninguno de los tres casos, los puntos señalados detallan el contenido exacto que se puede encontrar en cada una de las referencias, simplemente se puso énfasis en los puntos que no son comunes, ya que en todos también se encontrará información sobre prácticas de seguridad, referencias al

modelo de objetos de InfoPath, mecanismos para establecer conexiones a fuentes de datos, etc.

2.7.2 HERRAMIENTAS DE VISUAL STUDIO PARA MICROSOFT OFFICE 2003

Estas herramientas mejor conocidas como VSTO por sus siglas en Inglés *Visual Studio Tools for the Microsoft Office System*, son otro complemento adicional que se puede agregar al entorno de desarrollo de Visual Studio .NET.

En este caso, se proporcionan plantillas para la creación de proyectos de automatización de Word y Excel 2003, de igual manera se tienen a disposición todas las características proporcionadas por el ambiente de desarrollo y el *Framework* .NET, para crear las aplicaciones personalizadas de Office utilizando código administrado.

Es importante aclarar que *Microsoft Visual Studio Tools for the Microsoft Office System*, también es el nombre que utiliza la tecnología implícita en la creación de soluciones personalizadas de Office con código administrado; existe el *Microsoft Visual Studio Tools for the Microsoft Office Loader* [Sección 2.8.2], que es un componente COM (no administrado) embebido en el Office 2003 y que es utilizado por las aplicaciones para cargar y descargar los componentes de código administrado.

2.7.3 VERTIGO MANAGED SMART DOCUMENTS WRAPPER

Esta herramienta está diseñada específicamente para facilitar el desarrollo de Documentos Inteligentes utilizando código administrado, en realidad el tipo de soluciones que se crean con esta herramienta incluyen un contenedor o envoltura administrada para la Interfaz *ISmartDocument*, que proporciona varios elementos adicionales que permiten definir el diseño y la funcionalidad del panel de tareas **Acciones del documento** mediante declaraciones XML.

Los elementos esenciales que debe contener un Documento Inteligente son: el archivo o plantilla de documento (Word o Excel), el esquema personalizado XML,

el código de implementación, y el archivo de manifiesto; pero con esta herramienta se agregan dos elementos adicionales, un ensamblado (*SmartDocumentWrapper.dll*) que contiene las clases para definir el contenedor y un archivo de configuración **.xml**, que permite determinar los controles y atributos que se utilizarán en el panel de tareas **Acciones del documento**, utilizando simplemente definiciones mediante elementos XML.

En la Ilustración 2-5 se muestran los archivos básicos que contendrá un paquete de expansión para un Documento Inteligente desarrollado con *Vertigo Managed Smart Documents Wrapper*.

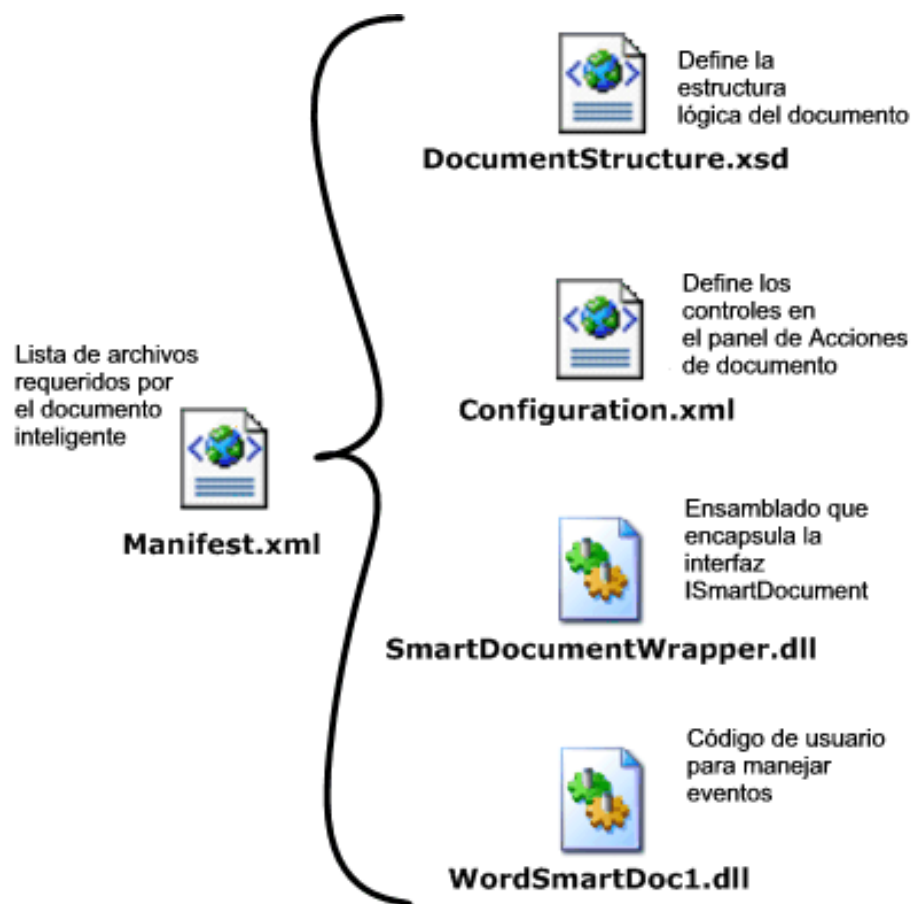


Ilustración 2-5 Archivos del paquete de expansión de una solución creada con *Vertigo Managed Smart Documents Wrapper*.⁴⁸

⁴⁸ Ilustración 2-5: tomada de la Referencia 34, fuente en inglés.

Es importante mencionar que esta herramienta ayuda con la creación de todos los archivos relacionados con el Documento Inteligente, de tal forma que el desarrollador tiene mucho menos trabajo en el momento de crear un Documento Inteligente. De igual forma viene con documentación que explica todas las particularidades del desarrollo, incluye un ejemplo bastante práctico y explicativo, en el cual el desarrollador podrá basarse para crear sus propias soluciones [se puede descargar de la Referencia 35].

2.8 MECANISMOS DE SEGURIDAD Y PRIVACIDAD

La seguridad es un tema muy importante en el desarrollo de soluciones personalizadas de Office, ya que la estructura flexible de sus aplicaciones, permite que desarrolladores mal intencionados, encuentren en Office un mecanismo simple para crear código malicioso de fácil distribución.

Como se ha mencionado anteriormente, en la versión de Office 2003, el código asociado a un documento puede ser administrado, o no administrado, y el modelo de seguridad será diferente e independiente en cada uno de los casos.

Cuando se habla de la seguridad del código no administrado dentro del Office, se hace referencia a macros, complementos no administrados y plantillas globales; sin embargo, este modelo de seguridad aunque es efectivo, es poco flexible, y la configuración del mismo casi siempre se pone bajo la responsabilidad de los usuarios finales.

Sin embargo, la posibilidad de utilizar código administrado en las soluciones personalizadas de Office 2003, permite ofrecer mecanismos de seguridad mucho más estructurados, controlados y confiables, que los usuarios finales no pueden modificar directamente desde las aplicaciones.

2.8.1 SEGURIDAD DE OFFICE 2003 A NIVEL DE DOCUMENTO

Adicionalmente a la seguridad del código, en Office 2003 se incluyen otras características que mejoran el nivel de seguridad a nivel de documento; las opciones que se pueden mencionar, entre nuevas y mejoradas, son las siguientes:

- **Firmas Digitales:** Se puede firmar el documento, con lo cual se asegura que todo el contenido está bajo la responsabilidad de su creador. La firma del documento, es un proceso independiente a la firma del código.
- **Fuentes Confiables:** Se puede restringir la ejecución de archivos que no se encuentren dentro de fuentes confiables; existen funciones que simplifican la administración de fuentes confiables.
- **Control de Acceso a Archivos:** Se ofrecen tres niveles de protección para restringir la utilización de un archivo: protección contra la apertura del archivo, protección contra la modificación del archivo (sólo lectura), y la recomendación de sólo lectura (se pregunta al usuario si quiere abrir en modo de escritura/lectura).
- **Remover Información Personal:** Esta es una característica de privacidad que permite eliminar de un documento, la información que se agrega automáticamente; tal como, ubicación, asunto, autor, título, etc., que muchas ocasiones es muy valiosa para las organizaciones y no debe ser transmitida.
- **Restricciones de Formato:** Los documentos se pueden limitar evitando cambios de formato, permitiendo que el usuario edite el contenido pero no el estilo de un documento.
- **Restricciones de Edición:** Se puede bloquear determinadas porciones de un documento, con las siguientes posibilidades: solo lectura, rastreo de cambios, sólo agregar comentarios, sólo agregar datos de formularios y establecer permisos por secciones y usuarios.

Algunas de estas opciones de seguridad, se pueden establecer desde el momento de la instalación de Office 2003, mediante el Asistente de Instalación

Personalizada (*Custom Installation Wizard* - CIW)⁴⁹, también se pueden implementar mediante políticas de seguridad del sistema o configurarlas directamente desde la interfaz de usuario, utilizando cuadros de diálogo y paneles de tareas.

2.8.2 SEGURIDAD DE LAS APLICACIONES PERSONALIZADAS DE OFFICE 2003 QUE UTILIZAN CÓDIGO ADMINISTRADO

Como se ha venido señalando, la naturaleza no administrada de Office, obliga a contar con la ayuda de ciertos elementos que permitan la interacción del código administrado con las aplicaciones de Office.

Para todo lo que tiene que ver con la seguridad, es importante saber que la relación entre los componentes de código administrado, los ensamblados de interoperabilidad, y las aplicaciones de Office, es manejada por el CLR del *Framework* .NET, que es una librería dinámica no administrada denominada *mscoree.dll*.

Cuando se configura la **Seguridad de macros** [**Error! No se encuentra el origen de la referencia.**] en las aplicaciones de Office, no solo se están aplicando restricciones a los macros embebidos en el documento (código VBA), sino también al código no administrado adjunto (complementos COM, plantillas globales, etc.), lo que implica que según el nivel de seguridad establecido (alto, medio, bajo, etc.), será necesario firmar cualquier tipo de código no administrado que tenga acceso a los recursos de una determinada aplicación.

En una solución personalizada de Office que utilice código administrado, la librería *mscoree.dll*, es la que interactúa directamente con las aplicaciones (Word, Excel, etc.); sin embargo, no está firmada digitalmente, y en cuestión de seguridad será vista como un simple complemento no administrado que está intentando utilizar algún recurso de las aplicaciones. Por esta razón, no podría pasar por las

⁴⁹ El CIW se incluye en un utilitario denominado *Kit* de Recursos de Office, que se puede obtener de la Referencia 39.

políticas de seguridad establecidas para el código no administrado; es decir, al no ejecutarse la librería `mscorlib.dll` es imposible que se ejecute el código administrado de la solución.

En vista de que el esquema de seguridad para el código no administrado utilizado por el Office es poco flexible (es una protección de todo o nada), es necesario que se lo independice del modelo de seguridad del código administrado, para poder explotar sin problemas las características de seguridad ofrecidas por el *Framework*.NET.

Para lograr esta independencia, es necesario agregar una capa de código no administrado que tenga una firma digital, esta capa de código, por un lado deberá pasar sin problemas las políticas de seguridad del código no administrado, y por el otro deberá encapsular la ejecución del CLR, esto permitirá que la librería `mscorlib.dll` se ejecute y que el código administrado de una solución personalizada de Office, se rija exclusivamente por las políticas de seguridad establecidas en el *Framework*.NET.

En Office 2003 esta capa de código, incluida en forma predeterminada, se denomina *Visual Studio Tools for Office Loader*, y todos los proyectos creados con las herramientas mencionadas en la Sección 2.7 la utilizan para permitir que las aplicaciones puedan cargar y descargar el código administrado.

Se puede prescindir del *Visual Studio Tools for Office Loader*, y crear contenedores COM personalizados que cumplan la misma función, pero como es de suponerse, esto es un tema de estudio mucho más complejo y sale del alcance de este proyecto; sin embargo, se puede revisar la utilización de la herramienta COM *Shim Wizard* [Ref. 38], que ayuda con este propósito.

2.8.3 APLICACIONES PERSONALIZADAS DE OFFICE 2003 QUE UTILIZAN EL VISUAL STUDIO TOOLS FOR OFFICE LOADER

En la Ilustración 2-6 se muestra la Arquitectura típica de una solución corporativa basada en documentos personalizados de Office 2003, que se distribuye por una red y que al utilizar código administrado se ajusta a las políticas de seguridad de .NET.

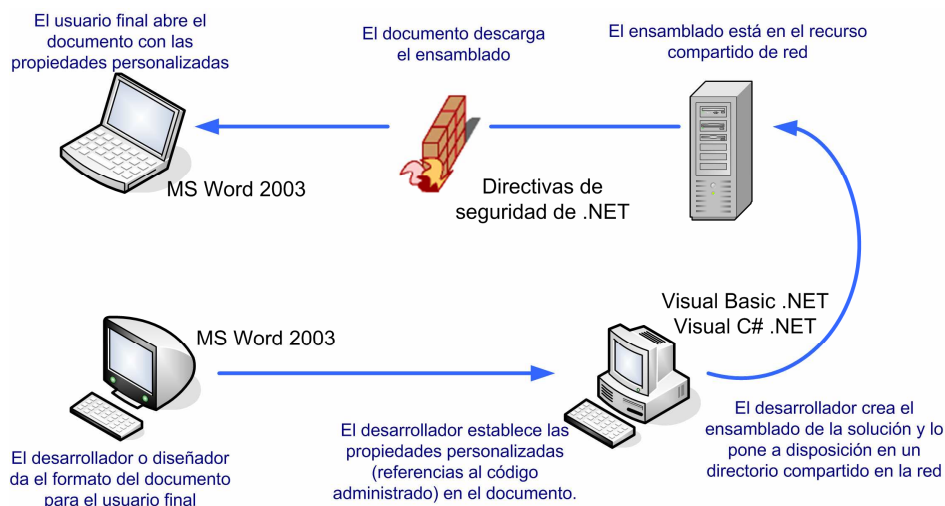


Ilustración 2-6 Arquitectura de soluciones de Office con código administrado.⁵⁰

En la Ilustración 2-6 la parte de **Directivas de seguridad de .NET** (máquina del usuario final) es la que se ejecuta con la ayuda del *Visual Studio Tools for Office Loader* o cualquier otro componente adicional con funciones similares.

En caso de no existir ningún problema, la secuencia de pasos que se sigue para lograr que el usuario final abra un documento que contiene propiedades personalizadas con código administrado, es la siguiente:

1. El usuario abre el documento
2. Se leen las propiedades personalizadas (configuraciones que indican las Referencias de código administrado)
3. Si existen Referencias a algún ensamblado se carga el CLR (utilizando el *Loader*)
4. Se revisan si el documento está es una zona de confianza
5. Se crea el dominio de la aplicación (AppDomain)⁵¹

⁵⁰ La Ilustración 2-6 es una modificación de un gráfico en la Referencia 58.

6. Se descarga el ensamblado compilado, desde la ubicación de red compartida (en caso de tener una instalación corporativa)
7. Se chequean permisos de ejecución de código
8. Se disparan los eventos con código
9. Se abre el documento

En síntesis para completar el proceso de apertura de un documento personalizado con código administrado, se debe tener autorización para abrir el documento y para ejecutar el código asociado.

En la Ilustración 2-7 se muestra gráficamente la secuencia de los nueve pasos mencionados. Es importante recalcar que esto es un modelo de ejecución utilizando el *Visual Studio Tools for Office Loader*, en caso de utilizar un contenedor personalizado la secuencia podría variar.

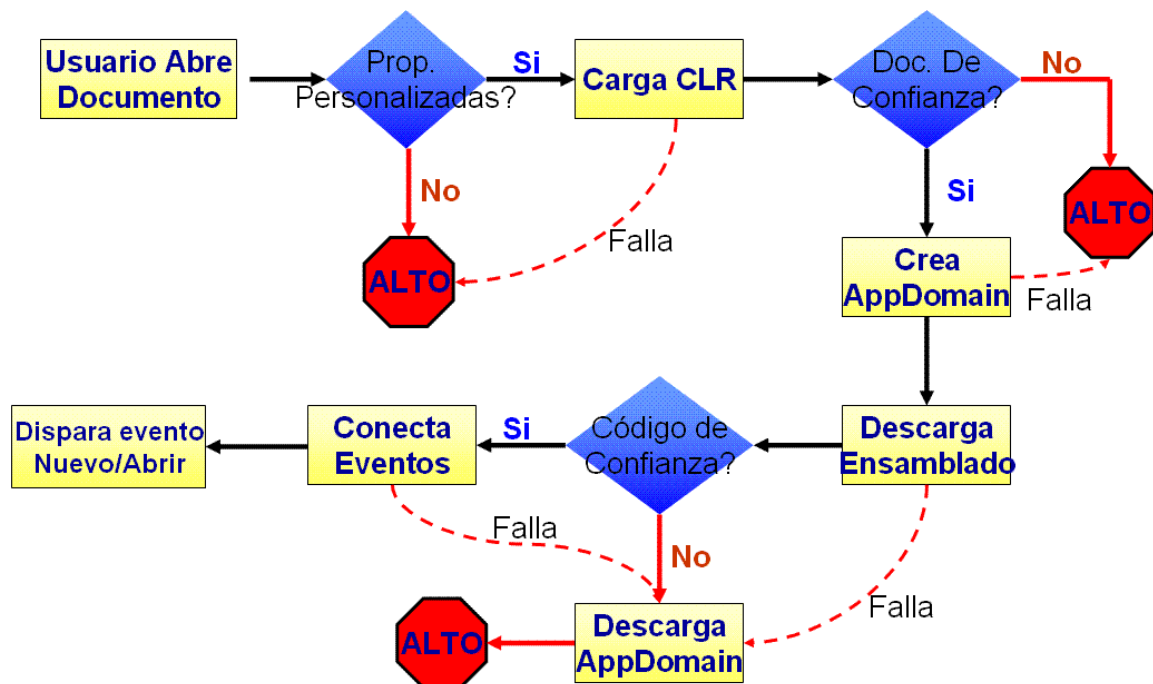


Ilustración 2-7 Partes fundamentales del proceso de carga de código utilizando *Visual Studio Tools for Office Loader*.⁵²

⁵¹ Los Dominios de Aplicación son unidades de procesamiento que pueden ser utilizadas por el CLR para proporcionar aislamiento entre aplicaciones, y garantizar que el código que se ejecuta en una aplicación no afecte negativamente a otras aplicaciones relacionadas.

⁵² Ilustración 2-7: tomada de una presentación en PowerPoint, distribuida por la Universidad .NET de Microsoft Argentina; se encuentra en el CD adjunto.

En la Ilustración 2-7 hay que tener presente que para abrir el documento solo basta tener la autorización, y para que se ejecute el código hay que tener la autorización de ejecución y no debe ocurrir ningún error, todas las condiciones de ALTO indican que el código no se ejecutará; sin embargo, esto no significa que el documento no se abrirá.

2.8.4 SEGURIDAD EN LOS DOCUMENTOS DE OFFICE 2003 QUE UTILIZAN XML

En el caso de los Documentos Inteligentes la firma digital es un requisito indispensable para la ejecución. En la Sección **¡Error! No se encuentra el origen de la referencia.** se menciona que entre otras cosas, para que exista un Documento Inteligente, se debe tener asociado un esquema XML y un paquete de expansión XML.

El esquema XML adjunto, por su función y limitaciones (solo define los nombres y atributos de los elementos XML) no representa una amenaza significativa de seguridad, pero el paquete de expansión XML al contener la lista de todos los archivos necesarios para la ejecución del Documento Inteligente si (se puede incluir en la lista cualquier tipo de código malicioso), convirtiéndose en un punto fundamental a ser protegido en cualquier implementación.

Por esta razón, el archivo de manifiesto XML que es el que en realidad define al paquete de expansión XML siempre deberá estar firmado digitalmente⁵³.

Existen varias herramientas que permiten firmar archivos, pero en particular se puede utilizar la denominada XMLSign.exe, que está especialmente diseñada para firmar archivos XML, su utilización es bastante sencilla, y cumple eficientemente con su propósito. Se puede obtener más información sobre el funcionamiento de esta herramienta en la Referencia 36.

⁵³ En un ambiente de desarrollo se puede manipular el Registro de Windows para que se permita la ejecución de un documento que tenga asociado un paquete de expansión sin una firma digital, para ver como hacer esto se puede consultar la Referencia 36, en la sección *Testing and Troubleshooting*.

Las aplicaciones de Office son muy estrictas en el proceso de chequeo de seguridad con los paquetes de expansión XML, y los pasos internos que se siguen, están sintetizados en la Ilustración 2-8.

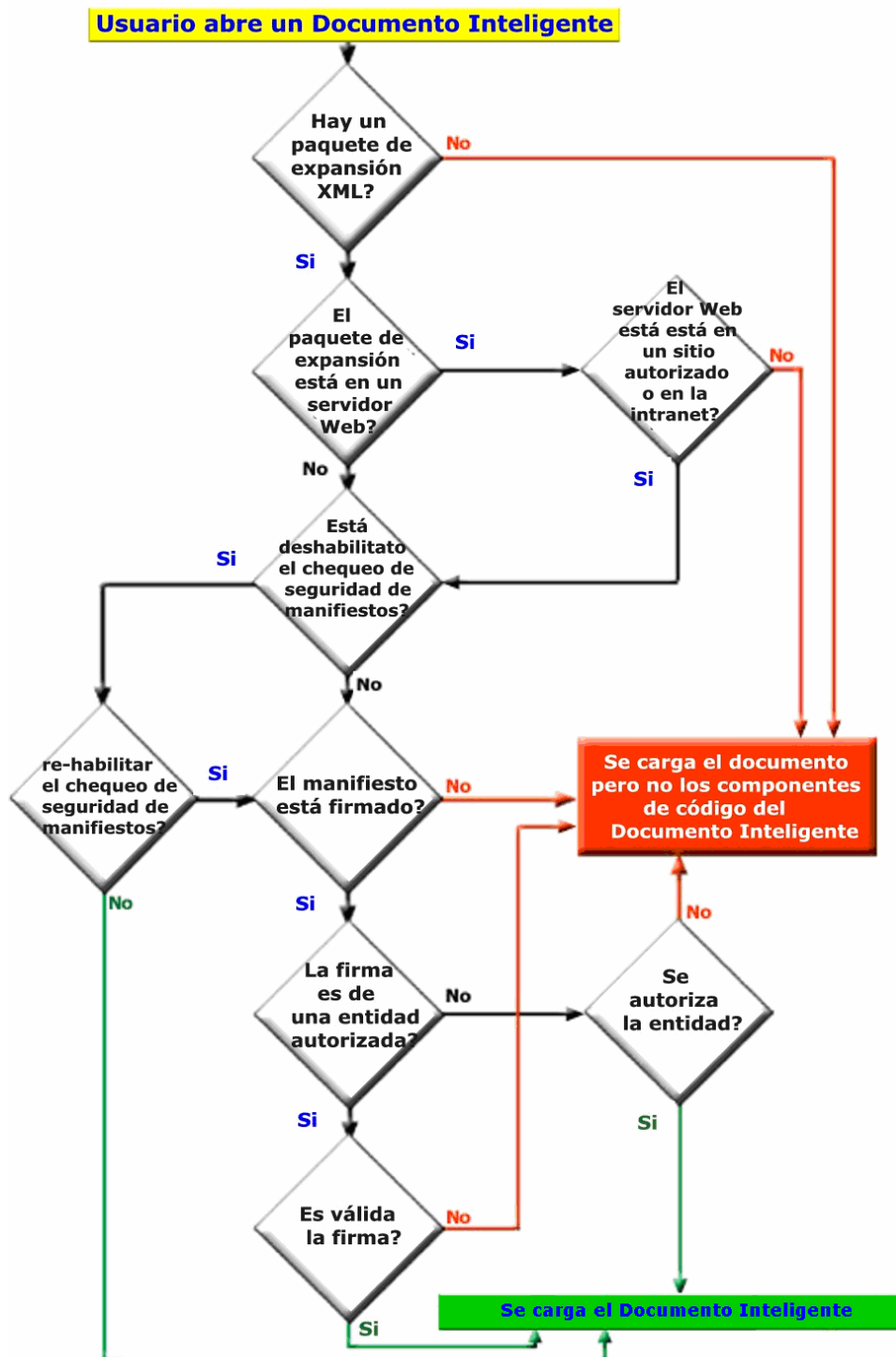


Ilustración 2-8 Chequeos de seguridad para los paquetes de expansión XML.⁵⁴

⁵⁴ Ilustración 2-8: tomada de la Referencia 37, fuente en inglés.

CAPÍTULO 3. DISEÑO E IMPLEMENTACIÓN DE UNA SOLUCIÓN PERSONALIZADA DE MICROSOFT OFFICE 2003 CON .NET Y XML

La solución personalizada de Office que se desarrolla en este proyecto para probar las características de las tecnologías mencionadas en los Capítulos anteriores, es una automatización para que las secretarías de cada una de las Carreras de Electrónica de la Escuela Politécnica Nacional, puedan generar las cartas y memorandos relacionados con los Proyectos de Titulación de los estudiantes.

Es importante mencionar: que en la actualidad todas las secretarías de las Carreras, utilizan el Microsoft Word para elaborar las cartas y memorandos, y como se verá en este Capítulo la mayor parte de la solución personalizada desarrollada se basa en la utilización de un Documento Inteligente de Word, razón por la cual se espera que el impacto de implementación sea bajo para las personas que lo utilizarán.

Entre los aspectos más importantes que se mencionan a lo largo de todo el Capítulo se encuentran:

- Detalles del diseño de la Solución, incluyendo un análisis de los requerimientos y la situación actual
- Arquitectura de la solución y la forma en la que fueron implementados cada uno de sus elementos
- Políticas de seguridad empleadas y aspectos necesarios para que la solución pueda ser instalada en cualquier computador
- Pruebas realizadas con la solución personalizada, comprobando el cumplimiento de cada uno de los requerimientos.

3.1 ANÁLISIS DEL PROBLEMA, SITUACIÓN ACTUAL Y REQUERIMIENTOS

En esta sección se determinan los requerimientos de funcionalidad de la solución personalizada, simplificándolos en un diagrama de casos de uso como resultado del análisis de la problemática actual y teniendo en consideración los aspectos más importantes de la situación actual.

3.1.1 ANÁLISIS DEL PROBLEMA

Luego de haber realizado algunas visitas a las secretarías de las Carreras de Ingeniería en Electrónica (Telecomunicaciones, Redes, y Control) de la Escuela Politécnica Nacional, se pudo observar varios problemas comunes, relacionados al manejo de la información asociada a los Proyectos de Titulación, los cuales se mencionan a continuación:

- No existe un registro completo que contenga toda la información necesaria relacionada con los Proyectos de Titulación, lo que impide obtener datos que serían relevantes tanto para el proceso de aprobación de nuevos Planes de Proyectos de Titulación, así como también para la asignación de Tribunales Examinadores. Algunos ejemplos de la información inexistente son: el estado actual de un proyecto; el número de proyectos que está dirigiendo o calificando actualmente un profesor; un listado de los temas que han sido rechazados y los justificativos; un historial sobre los proyectos concluidos, que permita buscar fácilmente quienes fueron los miembros del Tribunal Examinador, las notas que se obtuvieron, el tiempo de desarrollo, etc.
- Existen muchas plantillas de Word (archivos **.dot**), para la elaboración de cada uno de los documentos que se emiten desde las secretarías, no sólo aquellos relacionados con los Proyectos de Titulación, sino también con documentos internos generales de la Institución, lo que hace que las secretarías pierdan tiempo buscando la plantilla adecuada. En muchas ocasiones en lugar de buscar la plantilla, se utilizan documentos guardados con anterioridad (mas fáciles de encontrar) que se utilizan como modelo.

- Especialmente en la carrera de Telecomunicaciones, los documentos que se elaboran con relación a los Proyectos de Titulación, son almacenados electrónicamente en carpetas (directorios) según el año actual, utilizando como nombre del documento los apellidos representativos de los estudiantes que presentan el proyecto, lo cual utiliza muchos recursos de disco para el almacenamiento, y principalmente no proporciona un buen sistema de búsqueda (hay que recurrir al buscador del sistema operativo para encontrar un documento específico).
- Si bien es cierto que en la Carrera de Ingeniería en Electrónica y Redes de Información la cantidad de Proyectos de Titulación presentados hasta la actualidad no es muy grande, en las otras Carreras de Electrónica, el número de registros supera los trescientos, por lo que se debe buscar un mejor mecanismo de almacenamiento de esta información (actualmente en Microsoft Excel), para facilitar su tabulación y búsqueda.⁵⁵
- En ninguna de las Carreras de Electrónica se tiene a disposición un listado de los Temas de los Planes de Proyecto de Titulación, que han sido presentados y que están en proceso de aprobación, razón por la cual es difícil controlar la aprobación de Planes de Proyectos de Titulación similares si se presentan casi simultáneamente en Carreras distintas.

3.1.2 SITUACIÓN ACTUAL

Para hacer el análisis de la situación actual, se van a considerar todos los aspectos relacionados con los Proyectos de Titulación, desde antes de la aprobación de los mismos, hasta la culminación con el Grado Oral que deben rendir los estudiantes.

Todo Proyecto de Titulación debe regirse a un Plan de Proyecto de Titulación, razón por la cual, de aquí en adelante en varias ocasiones para facilitar la

⁵⁵En la Carrera de Telecomunicaciones desde octubre de 2000 hasta diciembre de 2005, se aprobaron 545 Temas de Proyectos de Titulación, en la Carrera de Redes de Información, desde junio de 2004 hasta diciembre de 2005, se aprobaron 23.

nomenclatura, se hará referencia a cada uno de ellos como PT o PPT, respectivamente.

3.1.2.1 Proceso de Aprobación

El proceso existente para la aprobación de un PPT, y la elaboración del documento de notificación dirigido al director y al codirector (en caso de existir), es similar en cada una de las Carreras de Electrónica, y se podría resumir en los siguientes pasos:

1. El o los estudiantes presentan el PPT
2. Las secretarías ponen un sello con la fecha en la que se receipta el PPT
3. Cada uno de los PPTs son tratados para su aprobación, en sesiones de la Subcomisión Académica Permanente correspondiente
4. Las secretarías apuntan todas las recomendaciones, observaciones, cambios y sugerencias que se van haciendo a cada uno de los PPTs, durante una sesión
5. Los días posteriores a una sesión, los estudiantes pueden acercarse a la secretaría respectiva, para preguntar por el resultado de la aprobación de su PPT, el mismo que podría ser:
 - La negación definitiva por motivos diversos y que deben ser debidamente justificados por la Subcomisión Académica
 - La aprobación puede permanecer pendiente, si la Subcomisión Académica considera que hay que hacer varias modificaciones significativas
 - La Subcomisión Académica puede delegar al Coordinador de Carrera la aprobación de un PPT, en caso de requerir pequeñas modificaciones, las mismas que serán revisadas por este último para la aprobación definitiva
 - La Subcomisión Académica puede aprobar el proyecto si considera que no hay que hacer ningún cambio adicional
6. Cuando un PPT ha sido aprobado, la secretaria redacta un memorando indicando el particular, dirigido al director y al codirector (en caso de existir) del proyecto. Para la redacción de este documento, las secretarías tienen a

disposición tres plantillas de Word que permiten elaborar documentos de texto diferente, según cualquiera de los siguientes casos:

- Documento de Aprobación remitido por el Coordinador de Carrera, cuando la Subcomisión Académica aprobó directamente el PPT
 - Documento de Aprobación remitido por el Director de la Escuela de Ingeniería, cuando el director del PPT aprobado es el Coordinador de la Carrera
 - Documento de Aprobación por Delegación remitido por el Coordinador de Carrera, cuando la Subcomisión Académica delegó la aprobación
7. Finalmente, se registra el título, nombre del director, nombre del codirector (en caso de existir), nombre del estudiante o de los estudiantes que presentan el PPT, y la fecha de aprobación, en un archivo de Excel.

Durante todo este proceso existen algunos puntos de interés que se deben mencionar:

- Las sesiones son de periodicidad distinta en cada una de las carreras
- En el caso de que el PPT no sea aprobado y requiera cambios, las secretarías leen verbalmente sus apuntes, sin entregar ningún documento escrito al estudiante, o al director del proyecto
- Las correcciones a un PPT deben ser presentadas la cantidad de veces que sean necesarias para su aprobación
- Durante el transcurso del proceso de aprobación, los estudiantes pueden cambiar de PPT todas las veces que así lo deseen, o hacer modificaciones de los actores de uno que hayan presentado, incluyendo: cambio de director, ingreso de un codirector, separación de dos estudiantes que presentaron un PPT conjunto, etc.
- En caso de que algún estudiante cambie completamente el PPT, cualquier otro estudiante podría retomar el tema, haciendo varias modificaciones

3.1.2.2 Proceso Posterior a la Aprobación

Una vez aprobado un PPT, todavía existe la posibilidad de que los estudiantes cambien completamente el mismo, con lo que volverían al proceso anterior, razón por la cual aunque este tipo de casos no es frecuente, debe ser considerado.

En general, luego de un tiempo indeterminado⁵⁶, los estudiantes presentan los borradores de su trabajo escrito para la calificación, para lo cual la Subcomisión Académica Permanente, deberá nombrar un Tribunal Examinador del proyecto escrito; este tribunal tiene un plazo máximo de diez días término luego de la notificación de su designación, para entregar las notas con su respectivo informe.

Cada estudiante tiene que recopilar varios documentos, que permitan que se lo declare apto para rendir el Grado Oral. Una vez que el Tribunal Examinador del proyecto escrito ha entregado las calificaciones, y el estudiante ha presentado toda su documentación, el estudiante está apto para la defensa oral de su proyecto, para lo cual la Subcomisión Académica Permanente de la Carrera, debe designar un Tribunal Examinador y un Presidente que presidirán el Grado Oral del estudiante; por lo general es el mismo Tribunal Examinador del proyecto escrito, pero puede ser distinto.

Finalmente, el presidente que preside el Grado Oral, entrega en secretaría la nota promedio obtenida por el estudiante, y un formulario que indica algunas observaciones y sugerencias de aplicación del Proyecto de Titulación.

Durante todo este proceso, en las secretarías de las respectivas Carreras, redactan varios documentos, para lo cual en total, se utilizan 9 plantillas de Word con texto y formatos preestablecidos. El archivo de Excel en el que se registra la información básica de los Proyectos de Titulación cuando son aprobados, es la mejor fuente de información que poseen las secretarías para completar los datos requeridos en cada uno de estos documentos.

⁵⁶ Según el reglamento vigente para el 2006, en condiciones normales un estudiante tendrá como plazo máximo tres matrículas para terminar su proyecto de titulación, incluida la aprobación.

Los documentos que se deben elaborar, en orden secuencial y dependiendo del estado (en desarrollo, calificación, defensa, etc.) en el que se encuentre un proyecto [ver Ilustración 3-1], son los siguientes:

- Designación del Tribunal Examinador para que califique el Proyecto de Titulación Escrito
- Proveído de las Calificaciones del Proyecto de Titulación Escrito
- Declaración de Aptitud para rendir la Defensa Oral del Proyecto de Titulación
- Designación para integrar el Tribunal Examinador de la Defensa Oral
- Formulario para la Defensa Oral de Ingeniero
- Proveído de las Calificaciones de la Defensa Oral
- Proveído sobre Nota y Promedio de la Graduación

La razón por la que se utilizan nueve plantillas para la elaboración de siete documentos, es que tanto en el nombramiento del Tribunal Examinador para que califique el Proyecto de Titulación Escrito, como en la designación para integrar el Tribunal Examinador de la Defensa Oral, se tiene en consideración que el remitente del memorando debe ser el Director de la Escuela de Ingeniería cuando el director del proyecto es el Coordinador de la Carrera.

3.1.3 REQUERIMIENTOS

Con toda la información sobre la problemática encontrada y la situación actual descrita en las secciones anteriores, se pueden establecer varios requerimientos tanto de funcionalidad, seguridad y transferencia de datos, que debería tener una solución personalizada para que facilite el proceso de recopilación, tabulación, búsqueda, actualización y eliminación de información, ayudando a cumplir el objetivo principal: “Optimizar el proceso de elaboración de los documentos relacionados con los Proyectos de Titulación, que se emiten desde cada una de las secretarías de Electrónica”.

3.1.3.1 Funcionalidad

La solución personalizada deberá permitir que las secretarías de cada una de las Carreras de Electrónica, tengan a su disposición las siguientes funciones principales:

- Manejo de la información relativa a los Proyectos de Titulación, en concordancia con los datos existentes en la base de datos SAE (Sistema de Administración Estudiantil) que maneja la Escuela Politécnica Nacional
- Identificación y modificación del estado en el que se encuentra cada uno de los PPTs o PTs
- Generación automática de las cartas y memorandos mencionados en la Sección 3.1.2
- Generación automática de reportes

3.1.3.1.1 *Manejo de la información relativa a los Proyectos de Titulación*

A medida que se elaboran los documentos relacionados con los PTs, son necesarios algunos datos, los mismos que deberían ser almacenados en una base de datos para su posterior reutilización, ya sea para la generación automática de los documentos, para la elaboración de reportes, o incluso con fines estadísticos.

Toda la información debe tener procesos fáciles de ingreso, actualización, recopilación y eliminación. Un proyecto concluido debería tener asociado el siguiente conjunto de datos:

- Título o Tema
- Nombre del director y codirector (en caso de existir)
- Nombre del estudiante o de los estudiantes que lo presentan
- Fecha de presentación del PPT
- Áreas técnicas relacionadas
- Fechas de presentación de los cambios solicitados al PPT
- Fecha de delegación para aprobación (en caso de existir)
- Fecha de aprobación del PPT
- Fecha de la designación del Tribunal Examinador del trabajo escrito

- Nombre de los miembros del Tribunal Examinador del trabajo escrito
- Calificaciones otorgadas por cada uno de los miembros del Tribunal Examinador del trabajo escrito
- Nombres de cada uno de los miembros del Tribunal Examinador del Grado Oral
- Nombre del presidente que presidirá el Grado Oral
- Fecha y hora del Grado Oral
- Calificación promedio del Grado Oral

Solo por mencionar uno de los beneficios que se obtendrán al tener almacenada toda esta información, es importante hacer notar que con el dato del director del proyecto, se podrá identificar automáticamente la condición en la cual el director es el Coordinador de Carrera, con lo cual el nombre del remitente de la carta o memorando será directamente el Director de la Escuela de Ingeniería, eliminando las tres plantillas de Word (documentos de aprobación y designación de tribunales) que sirven para cubrir esta condición.

Adicionalmente a toda esta información que se recopilará, sería muy beneficioso para las secretarías tener acceso a la información personal (dirección, teléfono, número de cédula, etc.) de los profesores y estudiantes, que intervienen en cada uno de los proyectos, de tal forma que se pueda acceder a ella lo más rápido posible y sin abandonar la aplicación (MS Word), que en este caso estarán utilizando.

La información personal tanto de profesores como de estudiantes se encuentra en el sistema SAE, razón por la cual lo óptimo es reutilizarla, no para que la solución personalizada acceda directamente al SAE, sino para que se importen los datos necesarios a una nueva base de datos (necesario por el sistema basado en períodos que maneja el SAE).⁵⁷

⁵⁷ En la Sección 3.2.2.1 destinada al diseño de la solución, se explicará en detalle el porqué es más conveniente la importación que la utilización directa de los datos del SAE.

Los datos que se requerirán del sistema SAE son:

- Información personal de los profesores
- Información personal de los estudiantes (solo de aquellos que estén aptos para presentar un PPT o que ya lo hayan presentado)
- Información general de las materias que se dictan en las Carreras de Electrónica
- Información general de cada una de las carreras

3.1.3.1.2 Identificación y modificación del estado en el que se encuentra cada uno de los PPTs o PTs

En la actualidad para las secretarías de las Carreras de Electrónica, un PT solo puede tener dos estados, en “Desarrollo” o “Concluido”, esto se debe a que como se dijo anteriormente no se tiene un registro de los PPTs que están en aprobación, o que fueron negados.

Luego de conocer todo el proceso que sigue un PT, desde la primera vez que se presenta un PPT, hasta que se hace la defensa oral del proyecto, parecería bastante práctico aumentar la cantidad de estados de dos a ocho (incluyendo los posibles estados de un PPT), los mismos que en orden serían los siguientes:

- Plan de Proyecto de Titulación (PPT)
 - Negado o Abandonado por algún estudiante
 - Pendiente o en Aprobación
 - Delegado al Coordinador de Carrera para Aprobación
- Proyecto de Titulación (PT)
 - Aprobado o en Desarrollo
 - Con Tribunal Examinador para el Trabajo Escrito o en Calificación
 - Apto para la Defensa Oral o Calificado
 - Con Tribunal Examinador para el Grado Oral
 - Concluido

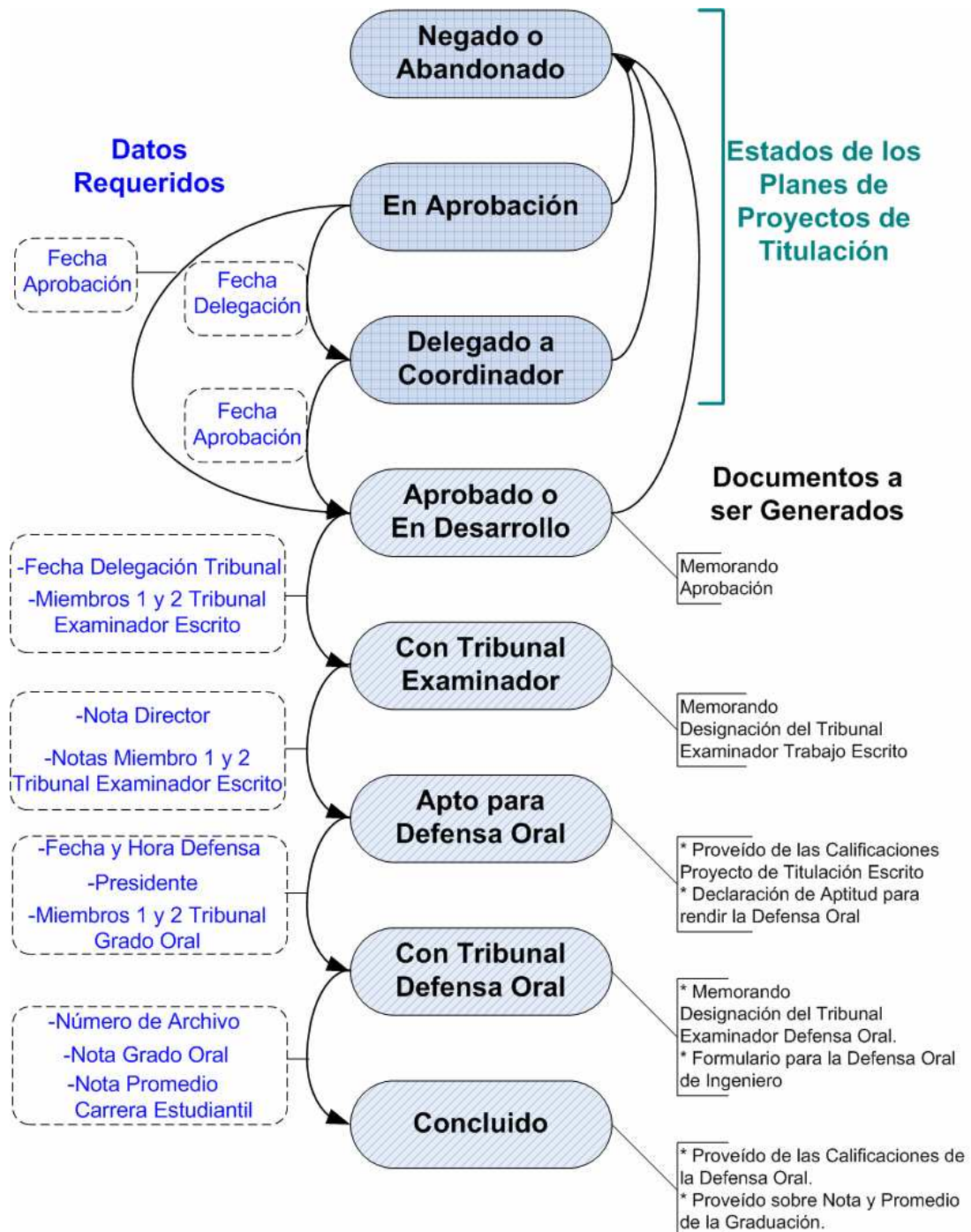


Ilustración 3-1 Diagrama de Estados PPTs y PTs.

En vista de que se pretende aumentar considerablemente la cantidad de estados que deben ser reconocidos por las secretarías, la identificación y cambio de un estado a otro debe ser un proceso bastante fácil, para que no existan confusiones.

La inclusión de estos nuevos estados permitirá tener algunos beneficios, por ejemplo:

- Las secretarías sabrán con exactitud los documentos que deben emitir, según el estado en el que se encuentre un proyecto.
- Se podrá saber si un determinado PPT, fue aprobado directamente por la Subcomisión Académica, o fue delegada la aprobación al Coordinador de Carrera, con lo cual será fácil generar el documento con el texto respectivo, sin que las secretarías deban preocuparse de la plantilla que deben utilizar.

3.1.3.1.3 Generación automática de las cartas y memorandos

La idea fundamental, es optimizar el espacio de disco que se está ocupando en la actualidad. Cada vez que se guarda un documento (carta o memorando), existe mucho texto común (repetitivo) que no aporta ninguna información adicional, el objetivo es almacenar únicamente los datos importantes que si aporten con información, y que el texto repetitivo esté almacenado en un solo lugar de fácil acceso.

Todo el contenido de los documentos se debe auto generar, en base a la información almacenada y al texto preescrito (actualmente en las plantillas).

Deberá existir un archivo que contenga todo el texto repetitivo, el mismo que tiene que poder ser manipulado por las secretarías, con la finalidad de prever los posibles cambios; es decir, que el código de la solución no debería requerir modificaciones si por algún motivo el texto de las partes comunes de alguno de los documentos varía.

3.1.3.1.4 Generación automática de reportes

La solución personalizada deberá tener acceso a la información más importante sobre los PTs o PPTs presentados en cada una de las Carreras de Electrónica, la forma en la que se deberá utilizar esa información, será mediante la generación de reportes con datos de interés común; como por ejemplo, título del proyecto, nombre del director, estado actual, estudiante o estudiantes que los proponen,

fecha de aprobación, miembros del Tribunal Examinador, fecha de culminación, etc.

Toda la información presentada en los reportes debe ser completamente dinámica (dependiente de una bases de datos), y se debe permitir aplicar filtros sobre la misma; estos filtros podrían ser: nombre del director, estado de los proyectos, carrera, texto contenido en el título, etc.

Los controles que permitan proporcionar los datos para filtrar la información deben ser de manejo fácil e intuitivo para las secretarías, lo que implica que se deben realizar las personalizaciones que sean necesarias a la interfaz de la aplicación (Word), ya sea en las barras de herramientas o en los paneles de tareas, para que los usuarios tengan un cómodo acceso, transparente y amigable.

Estos reportes serán de utilidad, ya que servirán de apoyo para la toma de decisiones en cuanto a la aprobación de los proyectos y a la designación de tribunales.

3.1.3.2 Conectividad

La solución personalizada debería funcionar en un ambiente distribuido sin importar el medio físico que se vaya a utilizar, incluyendo una transferencia de información que implique la utilización de Internet (pedido de la Subcomisión Académica Permanente de la Carrera de Electrónica y Redes de Información).

La solución además debería estar diseñada para soportar cualquier cambio en la configuración de la interconexión física, lo que implica que se debe permitir que el usuario configure fácilmente las direcciones de los equipos de donde se extraerá la información.

La información de configuración debe estar almacenada para que sea utilizada de manera predeterminada, de modo que se realice únicamente cuando se estime necesario, en ejecución normal los usuarios no deben preocuparse de realizar

esta configuración, y debe ser lo bastante simple como para que no sea imprescindible la presencia de un profesional especializado.

Adicionalmente, utilizando el servicio de Internet, la solución personalizada debería permitir enviar automáticamente por correo electrónico, los documentos que se auto generarán (como un servicio adicional, ya que los documentos escritos son indispensables), el usuario no debería abandonar la solución personalizada, ni siquiera para buscar la dirección de correo del destinatario.

3.1.3.3 Seguridad

Los requerimientos en cuanto a la seguridad de la solución, deben ser enfocados desde tres puntos específicos, bien diferenciados:

- Seguridad en los datos
- Seguridad en el código
- Seguridad en el Office

3.1.3.3.1 Seguridad en los Datos

En lo referente a los datos, con la finalidad de minimizar problemas que afecten la integridad de la información y para no generar una interdependencia entre cada una de las Carreras de Electrónica, se deberá evitar un ambiente centralizado que implique la existencia de un único punto de vulnerabilidad.

Para esto se deberán utilizar bases de datos independientes que estarán alojadas en cada uno de los computadores de las secretarías, y cuya modificación deberá ser un privilegio exclusivo de la respectiva Carrera.

Para la elaboración de los documentos, el ingreso y la actualización de la información, que es una tarea propia de cada Carrera, la solución personalizada instalada en cada una de ellas, deberá utilizar conexiones directas a la base de dato local, las mismas que deberán realizarse utilizando una autenticación con cuentas de usuario y sus permisos respectivos.

Para el caso de la elaboración de reportes, que es cuando se necesitará la información de las otras Carreras de Electrónica (no indispensable, puede ser un reporte local), se deberá acceder a la información únicamente con permiso de lectura.

3.1.3.3.2 Seguridad en el Código

Todo el código estará escrito en .NET, razón por la cual se deben tomar en consideración las facilidades de seguridad que ofrece esta tecnología, en las máquinas donde se implemente la solución lo único que debe existir son los ensamblados .NET, los mismos que deberán colocarse en un directorio específico oculto, al que se le concederán los permisos específicos como zona de confianza.

Se deberá firmar digitalmente los archivos de mayor importancia, para garantizar la fuente de los mismos.

3.1.3.3.3 Seguridad en el Office

En vista de que la solución personalizada será desarrollada con .NET, y en ninguna de las carreras de electrónica se utilizan soluciones con macros de VBA ni complementos COM, se deberán establecer las políticas de seguridad de macros en “Muy Alto”⁵⁸, que es el nivel más seguro que una aplicación de Office podría tener, con lo cual se logrará tener un ambiente relativamente seguro sin afectar en ningún sentido a la solución personalizada que se instalará.

Para acceder a la solución personalizada, los usuario deberán pasar por un control de acceso, para lo se debe aprovechar la opción de “Contraseña de Apertura”, con la que cuentan las aplicaciones de Office.

Adicionalmente, se deberá proteger la integridad del contenido de los documentos que se auto generarán, por lo que se requerirá recurrir al sistema de protección de párrafos y secciones, que es una característica propia con que cuenta la versión de Office 2003.

⁵⁸ El nivel de seguridad de macros en “Muy Alto”, sólo permite ejecutar las macros que estén instaladas en ubicaciones de confianza. Las demás macros firmadas y sin firmar se deshabilitarán.

3.1.3.4 Casos de Uso

Todo lo descrito en las Secciones anteriores, se puede sintetizar en un Diagrama de Casos de Uso⁵⁹ [Ilustración 3-2], el mismo que permite esquematizar en forma gráfica la funcionalidad y requerimientos de la solución personalizada, desde el punto de vista de los usuarios finales, y servirá de base para el diseño de la misma.

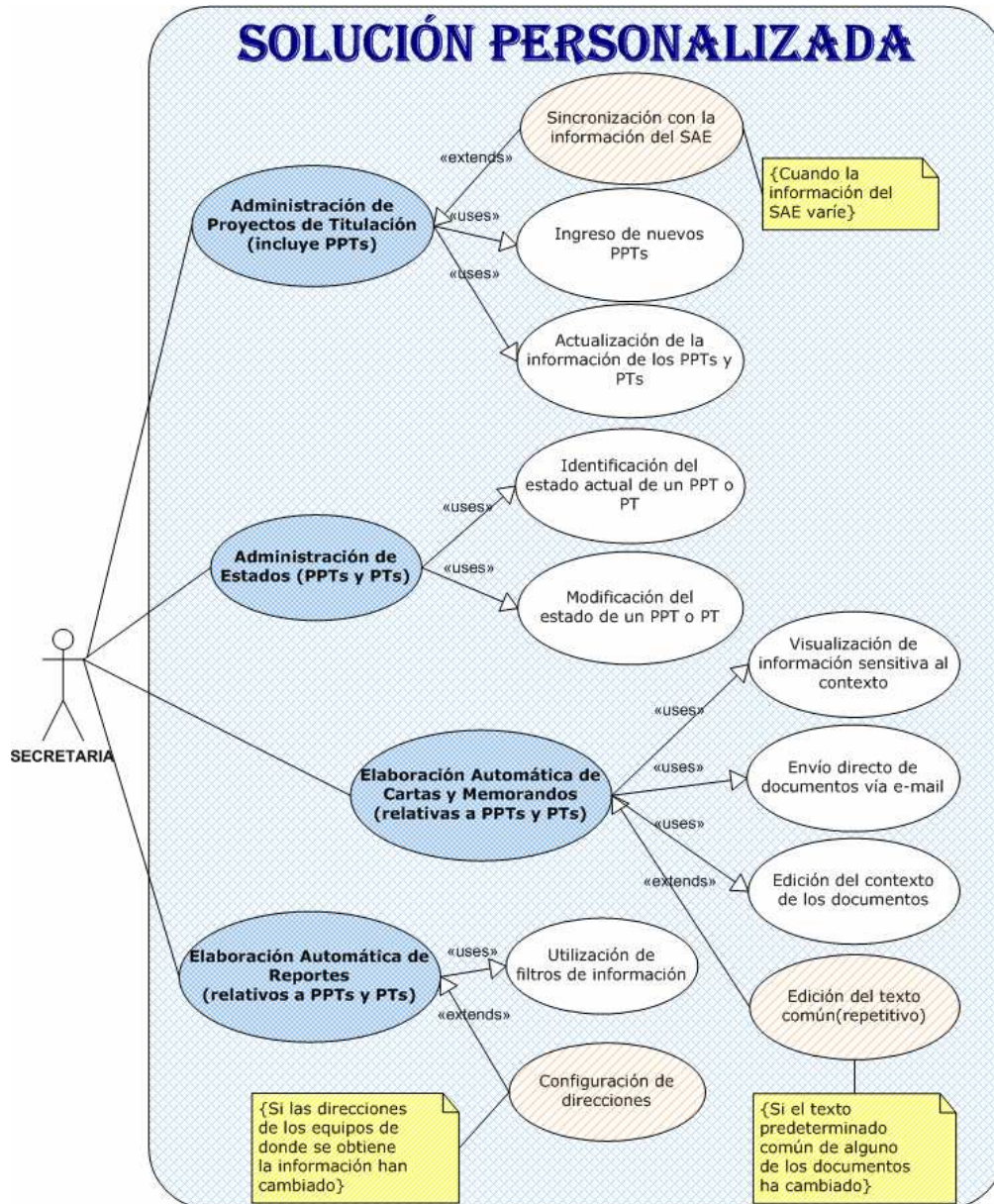


Ilustración 3-2 Diagrama de casos de uso.

⁵⁹ Los diagramas de casos de uso, utilizan el Lenguaje Unificado de Modelado (UML- *Unified Modelling Language*), para representar la forma en la que un sistema interactúa con el mundo exterior, mostrando los actores, los casos de uso y las relaciones entre ellos [Ref. 12].

3.1.3.4.1 Descripción de Casos de Uso

Para esta solución personalizada, se ha considerado la existencia de un solo actor⁶⁰ (la secretaria), en vista de que la idea fundamental es que todas las funciones sean tan fáciles de manejar, que a medida de lo posible no se requiera la presencia de ningún otro actor que configure o administre la solución.

De la Ilustración 3-2 es importante hacer notar que existen tres casos de uso con una relación de extensión (el caso de uso base puede incluir la funcionalidad del caso de uso de extensión bajo una determinada condición)⁶¹:

- Sincronización con la información del SAE
- Edición del texto común (repetitivo)
- Configuración de direcciones

Estos tres casos de uso deberán tener un tratamiento especial en el momento de realizar el diseño de la arquitectura e interfaz de la solución.

A continuación, en las Tablas 3-1 a 3-9 se realiza una pequeña descripción de los casos de uso mostrados en las Ilustración 3-2; en los casos de uso secundarios se ha incluido el flujo de sucesos en vista de que esta información será de mucha utilidad para el diseño de la solución.

Caso de Uso	Administración de Proyectos de Titulación
Función	Permitir el manejo de información de Proyectos de Titulación
Descripción	El actor podrá ingresar información relativa a nuevos Planes de Proyectos de Titulación, así como también modificar y eliminar la información de los existentes. Los datos relativos a los profesores y estudiantes que intervienen en cada uno de los proyectos, depende de una importación de información de la base de datos SAE, es decir que los nombres tanto de profesores como de estudiantes están disponibles únicamente para ser seleccionados (no se puede agregar, modificar o eliminar profesores ni estudiantes).

Tabla 3-1 Descripción del caso de uso “Administración de Proyectos de Titulación”.

⁶⁰ En los diagramas de casos de uso, a los usuarios de la solución se los conoce como actores, en la mayoría de casos existe mas de un actor para la misma solución.

⁶¹ Definición tomada de la Referencia 12.

Caso de Uso	Sincronización con la información del SAE
Descripción	Permite al actor importar información actualizada de la base de datos SAE a la base de datos SPODB; los datos a ser importados corresponden a Profesores, Estudiantes y Materias. En el momento de realizar la importación se verificará la existencia de datos comunes entre la SPODB y el SAE, de tal forma que sólo se importe la información que tenga el SAE pero que no tenga la SPODB.
Flujo de Sucesos	
Camino Básico	<ol style="list-style-type: none"> 1. El usuario solicita la sincronización de los datos 2. El sistema presenta una pantalla simple con un control que permitirá iniciar el proceso 3. El sistema realiza la verificación de datos existentes y copia de la base de datos SAE a la SPODB aquellos que no sean comunes entre las dos bases 4. El sistema muestra la cantidad de datos que fueron leídos y los que fueron importados

Tabla 3-2 Descripción del caso de uso "Sincronización con la información del SAE".

Caso de Uso	Ingreso de nuevos PPTs
Descripción	Permite al actor ingresar un nuevo Plan de Proyecto de Titulación, asociando el PPT con el o los estudiantes que lo presenten, para ello el actor tendrá a disposición una lista de todos los estudiantes que cumplan los requisitos para presentar un PPT y que no lo hayan hecho con anterioridad, esto evitará el ingreso de registros duplicados, ya que una vez que el estudiante presente un Plan será eliminado de la mencionada lista.
Flujo de Sucesos	
Camino Básico	<ol style="list-style-type: none"> 1. El usuario solicita el ingreso de un nuevo PPT 2. El sistema presenta una pantalla vacía para el ingreso de la información del PPT 3. El sistema presenta u oculta los controles para el ingreso de cierta información acorde a los datos que se vayan ingresando 4. El sistema valida y graba la información

Tabla 3-3 Descripción del caso de uso "Ingreso de nuevos PPTs".

Caso de Uso	Actualización de la información de los PPT y PTs
Descripción	Permite al actor modificar o eliminar la información de un Plan de Proyecto de Titulación que ha sido ingresado previamente.

Flujo de Sucesos	
Camino Básico	<ol style="list-style-type: none"> 1. El usuario solicita la modificación de un PPT 2. El sistema presenta una pantalla con un control que permita seleccionar el estudiante cuyo PPT se desea modificar o eliminar 3. El sistema presenta en la pantalla toda la información del PPT luego de realizada la selección del estudiante 4. El sistema valida y graba los cambios realizados 5. En caso de eliminación el sistema presenta una pantalla de confirmación de acción antes de eliminar el registro

Tabla 3-4 Descripción del caso de uso "Actualización de la información de los PPTs".

Caso de Uso	Administración de Estados (PPTs y PTs)
Función	Permitir el cambio del estado en el que se encuentre un Proyecto
Descripción	El actor podrá cambiar el estado en el que se encuentre un Plan de Proyecto de Titulación o un Proyecto de Titulación; cada uno de los cambios requerirán ciertos datos que deberán ser ingresados correctamente para que la transición se realice con éxito, caso contrario el cambio no se efectuará.

Tabla 3-5 Descripción del caso de uso "Administración de Estados".

Caso de Uso	Identificación del estado actual de un PPT o PT
Descripción	Permite al actor seleccionar un estudiante, ver el PPT o los PPTs que ha presentado y el estado en el que se encuentran.
Flujo de Sucesos	
Camino Básico	<ol style="list-style-type: none"> 1. El usuario selecciona el estudiante 2. En caso de que el estudiante haya presentado más de un Proyecto, el sistema presenta una pantalla que muestra la lista de todos los Proyectos asociados al estudiante (caso contrario ver numeral 4) 3. El usuario selecciona el Proyecto con el que desea trabajar 4. El sistema presenta la información general del Proyecto asociado al estudiante incluido el estado

Tabla 3-6 Descripción del caso de uso "Identificación del estado actual de un PPT o PT".

Caso de Uso	Modificación del estado de un PPT o PT
Descripción	Permite al actor cambiar el estado en el que se encuentra un PPT o PT, el cambio de estados tienen que ser sistemático y cada una de las transiciones requiere información adicional.

Flujo de Sucesos	
Camino Básico	<ol style="list-style-type: none"> 1. El usuario selecciona un estudiante 2. En caso de que el estudiante haya presentado más de un Proyecto, el sistema presenta una pantalla que muestra la lista de todos los Proyectos asociados al estudiante (caso contrario ver desde el numeral 4) 3. El usuario selecciona el Proyecto con el que desea trabajar 4. El usuario solicita la modificación de estado del proyecto asociado al estudiante seleccionado 5. El sistema presenta una pantalla con información del proyecto incluyendo el estado actual, también posee opciones de los estados a los que puede ser cambiado el proyecto y espacios en blanco para llenar la información requerida para el cambio de estado 6. El sistema valida y graba la información ingresada, y presenta las nuevas opciones para el cambio al siguiente estado

Tabla 3-7 Descripción del caso de uso "Modificación del Estados de un PPT o PT".

Caso de Uso	Elaboración Automática de Cartas y Memorandos.
Función	Permitir generar documentos automáticamente con información preexistente.
Descripción	El actor podrá seleccionar el nombre del estudiante y el tipo de documento (relacionado con el estado en el que se encuentre el proyecto presentado por el estudiante) que desee generar, y el documento será presentado automáticamente utilizando la información almacenada en la base de datos y el formato y texto preestablecido del documento. Todos los documentos auto-generados tendrán la particularidad de ser sensibles al contexto, es decir que presentarán algún tipo de ayuda en el panel de tareas de acción dependiendo de la ubicación del usuario en el documento. Los documentos estarán listos para ser impresos sin que el actor tenga que preocuparse por realizar alguna modificación, sin embargo podrá modificarlos.

Tabla 3-8 Descripción del caso de uso "Elaboración Automática de Cartas y Memorandos".

Caso de Uso	Visualización de información sensitiva al contexto
Descripción	Permite al actor visualizar diferentes tipos de ayudas (texto informativo, controles de selección, etc.) dependiendo de la ubicación del actor en un documento.

Flujo de Sucesos	
Camino Básico	<ol style="list-style-type: none"> 1. El usuario genera un documento 2. El usuario se moviliza por las diferentes secciones del documento 3. El sistema va presentando información o controles de ayuda dependiendo de la ubicación del usuario dentro del documento en el panel de tareas

Tabla 3-9 Descripción del caso de uso "Visualización de información sensitiva al contexto".

Caso de Uso	Envío directo de documentos vía e-mail
Descripción	Permite al actor enviar directamente un documento generado por el sistema a una o varias direcciones de correo electrónico, el asunto y la introducción son llenadas de forma automática por el sistema y si la base de datos contiene la información correcta de las direcciones de e-mail, las direcciones también son ingresadas por el sistema.
Flujo de Sucesos	
Camino Básico	<ol style="list-style-type: none"> 1. El usuario genera un documento 2. El usuario busca dentro del documento el nombre de la persona a la que desea enviar el documento y se ubica sobre la misma 3. El sistema muestra en el panel de tareas un control que permite enviar el documento a la persona sobre la cual se encuentra el usuario en el documento 4. El sistema presenta la información de tema, introducción y dirección de correo a ser utilizados para el envío 5. El usuario envía el documento si está de acuerdo con la información, caso contrario la puede editar antes del envío

Tabla 3-10 Descripción del caso de uso "Envío directo de documentos vía e-mail".

Caso de Uso	Edición del contexto de los documentos
Descripción	Permite al actor editar el texto de un documento generado por el sistema, así como también cambiar el formato y configuraciones de papel y documento para poder imprimirlo o enviarlo por correo electrónico. El actor también podrá configurar permisos de edición de tal forma que ciertas secciones del documento no puedan ser editadas a menos que se realice un proceso de autenticación.
Flujo de Sucesos	
Camino Básico	<ol style="list-style-type: none"> 1. El usuario solicita configurar los permisos de edición de los documentos

	<ol style="list-style-type: none"> 2. El sistema presenta una pantalla que permite determinar cuales documentos tienen permiso de edición y cuales no 3. El usuario solicita la generación de un documento 4. El sistema genera el documento 5. El sistema revisa la configuración de permisos de edición para el documento generado y los aplica 6. El usuario modifica el documento si tiene permiso
--	---

Tabla 3-11 Descripción del caso de uso "Edición del contexto de los documentos".

Caso de Uso	Edición del texto común (repetitivo)
Descripción	Permite al actor editar el texto base de los documentos que genera el sistema, los cambios que se realicen en este texto se verán reflejados en los documentos generados, en este caso la edición de este texto sería análogo a editar el texto de una plantilla de Word (base para la creación de un documento).
Flujo de Sucesos	
Camino Básico	<ol style="list-style-type: none"> 1. El usuario solicita la edición del texto común de los documentos 2. El sistema abre el archivo que contiene todo el texto para ser editado (como un documento de Word) 3. El usuario modifica el texto 4. El sistema guarda los cambios y cierra el archivo modificado

Tabla 3-12 Descripción del caso de uso "Edición del texto común".

Caso de Uso	Elaboración Automática de Reportes
Función	Permitir generar reportes con información de los PPTs o PTs
Descripción	El actor podrá generar documentos a manera de reportes con información de los Proyectos de Titulación que han sido presentados en la Carrera o en otras Carreras de Electrónica. El actor podrá utilizar filtros para seleccionar la información que desea visualizar; tal como, nombre del director, carrera en la que ha sido presentado el proyecto, estado actual del proyecto, título (completo o parcial). El reporte podrá ser editado e impreso con facilidad.

Tabla 3-13 Descripción del caso de uso "Elaboración Automática de Reportes".

Caso de Uso	Utilización de filtros de información
Descripción	Permite al actor utilizar algunos parámetros de búsqueda de información para presentar reportes más resumidos y que incluyan únicamente la información necesaria.
Flujo de Sucesos	
Camino Básico	<ol style="list-style-type: none"> 1. El usuario solicita la elaboración de un reporte 2. El sistema muestra en el Panel de Tareas determinados controles que permitan poner filtros de información 3. El usuario llena los datos que contendrá el filtro 4. El usuario solicita la información que coincida con el filtro 5. El sistema obtiene los datos y presenta la información

Tabla 3-14 Descripción del caso de uso "Utilización de filtros de información".

Caso de Uso	Configuración de direcciones
Descripción	Permite al actor configurar las direcciones de los servidores Web donde se encuentran alojados los servicios Web que proporcionan la información de los proyectos de titulación de otras carreras.
Flujo de Sucesos	
Camino Básico	<ol style="list-style-type: none"> 1. El usuario solicita configurar las direcciones de los servidores Web 2. El sistema presenta una pantalla que muestra los espacios para ingresar las direcciones 3. El usuario solicita probar si se establece la conexión con el servidor Web 4. El sistema intenta realizar una conexión y muestra mensajes de éxito o fracaso 5. El sistema almacena la información

Tabla 3-15 Descripción del caso de uso "Configuración de direcciones".

3.2 DISEÑO

El diseño de la Solución Personalizada de Office, a la que de aquí en adelante se hará referencia como SPO, consta de cuatro aspectos fundamentales: Arquitectura, Fuentes de Datos, Esquemas XML y Capas Lógicas; es importante conocer los detalles de cada uno de ellos para entender de mejor manera el funcionamiento de la solución, por esta razón las características más importantes serán presentadas a lo largo de esta Sección.

3.2.1 ARQUITECTURA

Aquí se comienza con una pequeña descripción de los elementos principales que componen la SPO, también se habla de los modelos de Interconexión Física y Lógica entre los equipos, y finalmente se presenta un esquema detallado de la Arquitectura Completa de la solución.

3.2.1.1 Elementos Principales e Interconexión Física

A continuación se dará una breve introducción a los cuatro elementos fundamentales (estarán instalados en cada uno de los equipos) de los que estará compuesta la SPO [ver Ilustración 3-3]:

- Un Formulario de InfoPath
- Un Documento Inteligente
- Una Base de Datos
- Un Servicio Web XML

El elemento principal de la solución será el Documento Inteligente, el mismo que tendrá implementadas las funciones necesarias para cubrir todos los requerimientos descritos en la Sección 3.1.3.

El Formulario de InfoPath servirá de ayuda, como un adicional que podrán utilizar las secretarías, para el proceso de recopilación de la información relativa a las observaciones, comentarios o sugerencias realizadas a cada uno de los PPTs, en las sesiones de Subcomisión Académica de las Carreras de Electrónica.

Tanto el Documento Inteligente como el Formulario de InfoPath, serán los únicos elementos de interacción directa con el usuario, y entre sus características principales se pueden mencionar:

- El Documento Inteligente, accederá a la información contenida en la base de datos local de dos maneras distintas, dependiendo de las funciones invocadas por el usuario:

- **Conexión Directa:** Se accederá directamente a la base de datos local, utilizando para la conexión una cuenta de usuario con permisos de lectura y escritura.
 - **Conexión Indirecta:** Esta conexión será utilizada únicamente cuando se vaya a generar algún reporte, no se accederá directamente a la base de datos local, sino que se lo hará a través de un Servicio Web XML, el mismo que a su vez será quien tenga una conexión directa a la base de datos local, utilizando una cuenta de usuario con permiso de lectura.
- El Documento Inteligente accederá a la información contenida en bases de datos remotas (para la generación de reportes), únicamente utilizando Servicios Web XML.
 - El Documento Inteligente manejará los modelos de objetos de Microsoft Office Word 2003 y de Microsoft Office Outlook 2003, para la generación y envío de los documentos respectivamente.
 - El Formulario de InfoPath accederá a la base de datos local a través de una conexión directa con permisos de lectura y escritura, y no tendrá acceso a bases de datos remotas.

En cuanto a la interconexión física, en la Ilustración 3-3 se muestra un esquema bastante simple y general; muestra únicamente una nube en vista de que existen muchas posibilidades para interconectar los tres equipos, algunas de las opciones serían:

- Red WLAN de Infraestructura, con un punto de acceso
- Red WLAN ad-hoc
- Red LAN cableada, con un concentrador
- Equipos independientes (no están unidos por ninguna red LAN), se comunicarán por Internet (requiere direcciones IP públicas para que los Servicios Web sean accesibles)

En la nube de interconexión física, al indicar que puede ser una Intranet o el Internet, no quiere decir que sean excluyentes, ya que se podrán utilizar en combinación.

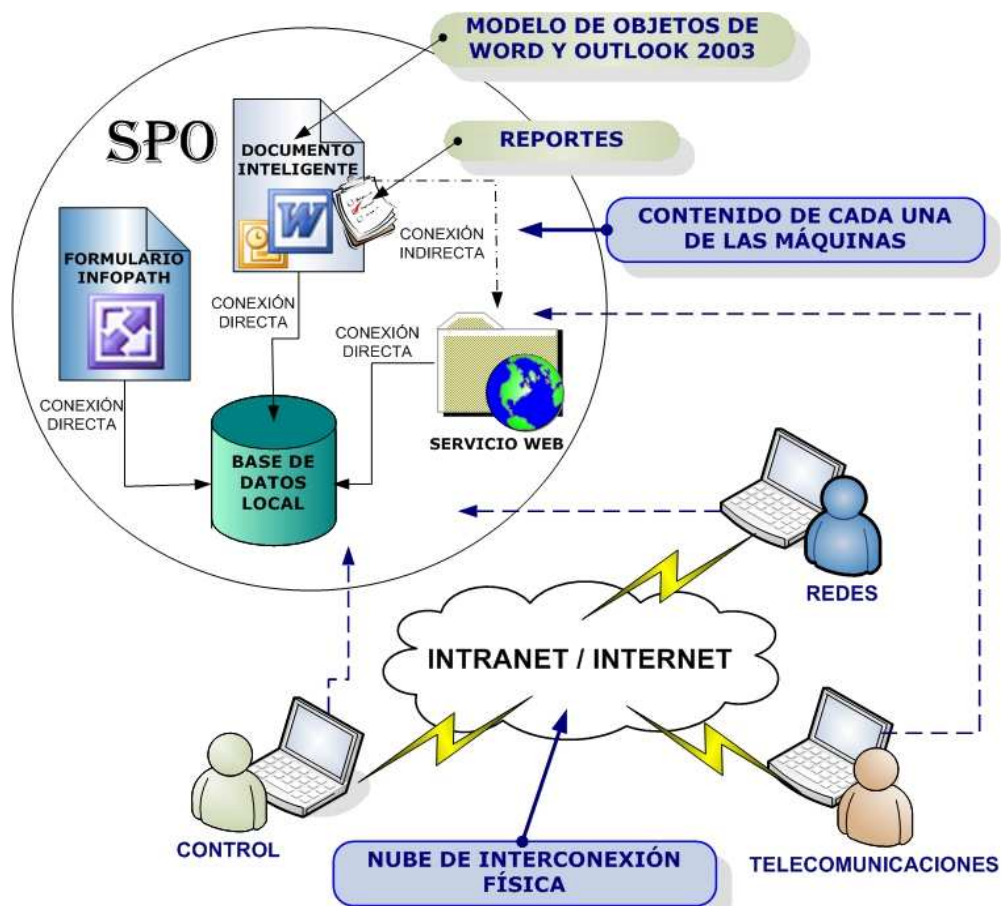


Ilustración 3-3 Elementos fundamentales de la SPO.

3.2.1.2 Interconexión Lógica

Dentro de los cuatro elementos fundamentales de la SPO, ya se mencionó la existencia de un Servicio Web XML; este elemento se ha considerado para el diseño, en vista de que permitirá aprovechar el soporte que ofrecen tanto la versión de Office 2003 como la tecnología .NET, para la utilización de este tipo de servicios como mecanismo de programación distribuido.

Y justamente la Interconexión Lógica, es la manera en la que una SPO (instalada en una determinada Carrera) estará conectada a otras SPOs (instaladas en las otras Carreras) para trabajar en un ambiente distribuido, sin importar el medio físico que se utilice para la transferencia de información.

Al utilizar Servicios Web XML, se realizará la transferencia de información en formato XML, empaquetado en SOAP y utilizando HTTP como protocolo de transporte; si bien es cierto que toda esta envoltura aumentaría el tráfico en una Intranet, también daría la posibilidad de enviar información libremente por Internet, haciendo a la SPO más funcional.

En la Ilustración 3-4 se pueden apreciar algunas de las opciones de Interconexión Lógica entre las SPOs instaladas en tres equipos diferentes. En la Sección anterior se mencionó la existencia de una conexión indirecta a la base de datos local a través de un Servicio Web XML, ésta es la que en esta figura se denomina “Conexión Local”.

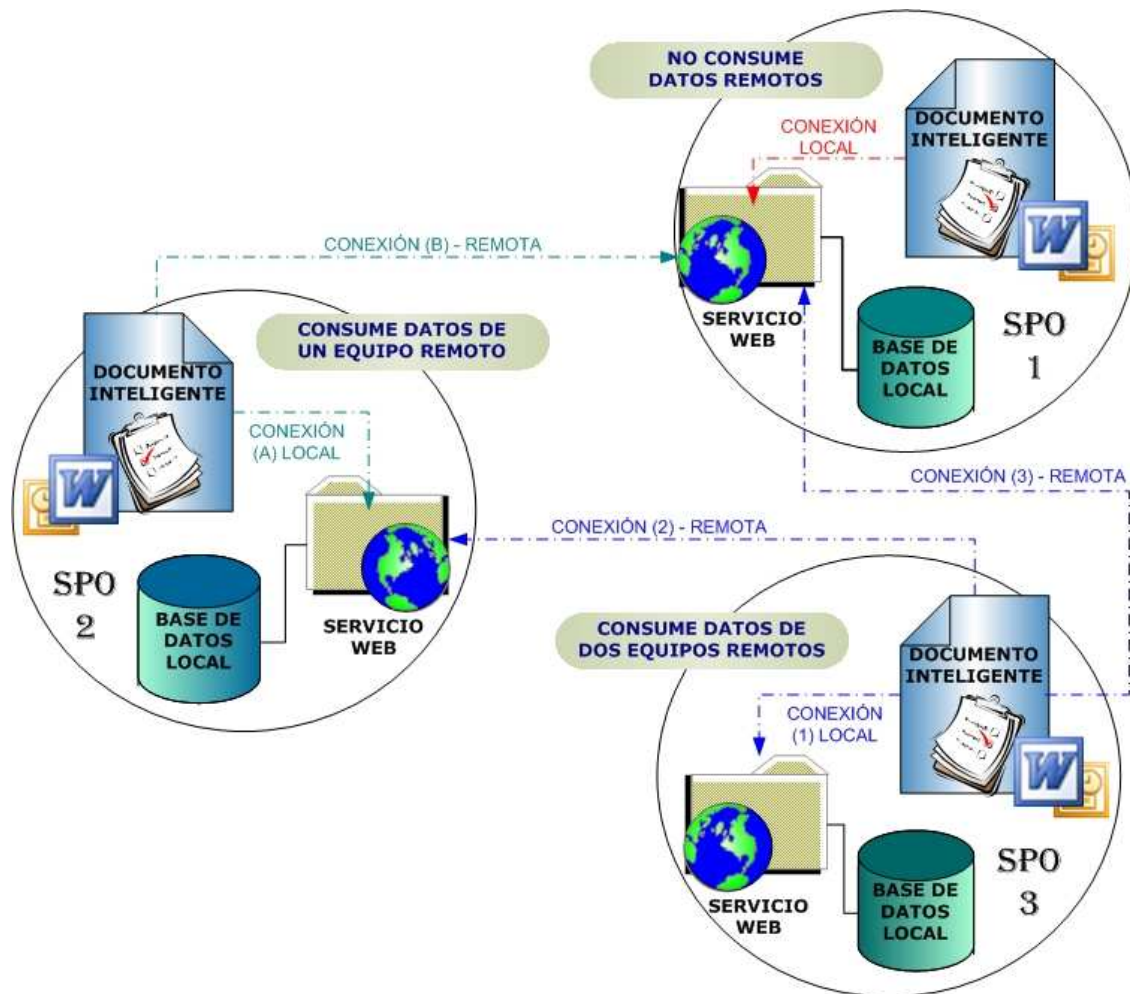


Ilustración 3-4 Ejemplos de interconexión lógica.

Al instalar la SPO en diferentes equipos, cada uno de ellos tendrá alojado un Servicio Web XML; en el momento de generar un reporte, el usuario podrá seleccionar las carreras de donde desea obtener la información, y aquí es donde entrarán en funcionamiento los Servicios Web instalados en cada computador.

Esto quiere decir, que para la generación de reportes, el Documento Inteligente no estará limitado únicamente a consumir el Servicio Web Local, sino que también podrá consumir Servicios Web Remotos, para lo cual las posibilidades de Interconexión Lógica entre las tres Carreras de Electrónica serían:

- Consumo de un Servicio Web Remoto
- Consumo de dos Servicios Web Remotos

En cualquiera de los casos, siempre existirá la opción de que el Documento Inteligente consuma o no el Servicio Web Local (si no se requieren datos locales, no se consume el Servicio Web Local).

3.2.1.3 Esquema Detallado de la Arquitectura

Tanto el Documento Inteligente, como el Formulario de InfoPath, necesitan de varios archivos (elementos secundarios), para su correcto funcionamiento; en el caso de InfoPath, la existencia de estos archivos será completamente transparente para el usuario; en el caso del Documento Inteligente existirá cierto grado de flexibilidad para que el usuario interactúe con algunos de estos archivos.

En la Ilustración 3-5 se muestra el esquema de la Arquitectura Completa de la SPO, considerando todos los elementos principales y secundarios que la componen. El esquema se ha realizado considerando un caso en el cual la SPO tiene una Interconexión Lógica utilizando dos Servicios Web XML Remotos; la interconexión física por medio de la cual se conecta con uno de ellos es una Intranet, mientras que al otro se conecta a través de Internet.

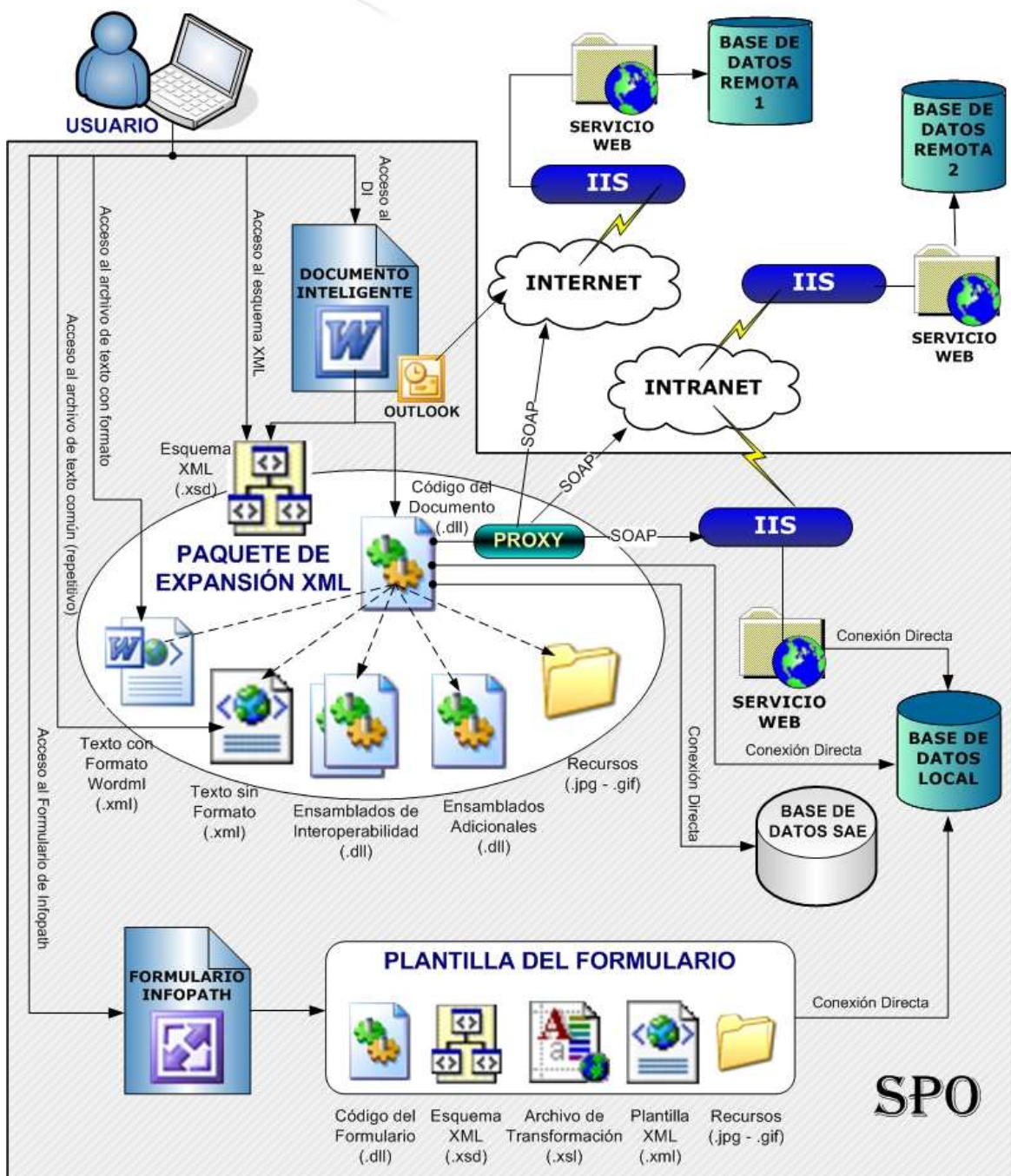


Ilustración 3-5 Arquitectura detallada de la SPO.

3.2.1.3.1 Plantilla del Formulario

Los Formularios de InfoPath se generan a partir de una Plantilla de Formulario, la misma que está compuesta por varios elementos [Tabla 2-2], el usuario sólo tendrá a su disposición el archivo **.xsn**, que ocultará la existencia de todos los demás.

Los elementos más importantes que componen la Plantilla del Formulario son:

- **Código .NET (.dll)**: Permite establecer la conexión directa a la base de datos local, y manipular la información contenida en todo el formulario.
- **Esquema XML (.xsd)**: Fundamental para determinar cada uno de los elementos XML que componen el formulario; muestra el camino a seguir para llegar al contenido de una etiqueta determinada.
- **Archivo de transformación XML (.xsl)**: Contiene toda la información (formato HTML) para presentar la interfaz de usuario del formulario.

3.2.1.3.2 Paquete de Expansión XML

Este es el elemento más complejo de toda la SPO, cualquier documento de Word, puede convertirse en Documento Inteligente tan solo con asociarlo a este elemento, los archivos que lo componen son:

- **Esquema XML (.xsd)**: Este archivo puede formar o no parte del Archivo de Manifiesto XML que define al Paquete de Expansión; es fundamental para etiquetar el contenido de cada documento que se auto generará; entre el Esquema XML y el Paquete de Expansión, siempre debe existir una relación (por el espacio de nombres). Al no formar parte del Archivo de Manifiesto, el Esquema XML deberá ser instalado por separado, pero siempre deberá estar asociado con el Documento Inteligente.
- **Texto con Formato WordML (.xml)**: Archivo XML que utilizará el esquema *WordProcessingML*, con la gran ventaja de que al ser el lenguaje XML propio de Word, se pueden almacenar párrafos de texto con el mismo formato que se desea que sean presentados (el usuario podrá editarlo).
- **Texto sin Formato (.xml)**: Este archivo, a diferencia del anterior no se basa en ningún esquema específico, es el que contendrá todo el texto común repetitivo de cada uno de los documentos que se auto generarán (el usuario podrá editarlo).
- **Ensamblados de Interoperabilidad (.dll)**: Adicionales a los PIAs de Office (no pueden estar en el Paquete de Expansión); estos ensamblados de interoperabilidad al no ser primarios, deberán ser instalados con la solución para no tener problemas de compatibilidad; permiten la interacción con componentes COM especiales que utilizará la aplicación.

- **Recursos (.jpg, gif, .bmp):** Son archivos gráficos que se utilizarán para ser mostrados en el panel de tareas del Documento Inteligente.
- **Código .NET (.dll):** Se refiere al ensamblado principal, deberá interactuar con todos los elementos que componen el Paquete de Expansión, y el Esquema XML, aquí es donde se manejarán los modelos de objetos de Word y Outlook; permitirá establecer las conexiones a la base de datos local, y a la base de datos del SAE, y adicionalmente contendrá un Proxy que permitirá tener comunicación con los Servicios Web XML.
- **Ensamblados Adicionales (.dll):** Son ensamblados requeridos por el ensamblado principal del Documento Inteligente, entre estos está el ensamblado SmartDocumentWrapper.dll, que permitirá encapsular la implementación de la interfaz *ISmartDocument*, indispensable para la ejecución de la lógica inteligente (ayuda al contexto, identificación de etiquetas, etc.) de la solución.

3.2.2 FUENTES DE DATOS

Para el diseño de las Fuentes de Datos, al ser la parte medular de la SPO, se ha tenido presente que deben ser, funcionales y bien estructuradas, para que la solución tenga un correcto flujo de información; esta Sección estará dedicada a describir las características de cada una de ellas (Base de Datos y Archivos de Origen de Datos).

3.2.2.1 Base de Datos

La Base de Datos diseñada para la SPO, recibirá el nombre de SPODB, las metodologías utilizadas en el diseño de la misma fueron las siguientes:

- Elaboración de diagramas de flujo de datos para determinar las tablas y sus relaciones
- Creación de un diccionario de datos para determinar los campos que deben contener los registros de cada una de las tablas

Una reseña sobre la nomenclatura que se utiliza en estas metodologías se puede encontrar en el ANEXO E o en la Referencia 6.

En la Sección 3.1.3 se realizó una narración verbal de los requerimientos del sistema, esto fue utilizado para diagramar gran parte de la lógica subyacente que debe contener la SPO; los diagramas realizados no incluyen la parte de la “lógica de la tecnología inteligente de los documentos” (ayuda relativa al contexto, etiquetado de documentos, identificación de etiquetas, etc.), en vista de que se harían muy extensos y confusos de entender.

En esta Sección los diagramas serán utilizados en el diseño de la SPODB, en vista de que sacan a relucir muy claramente los almacenes de datos⁶² y las relaciones entre ellos. Más adelante estos diagramas serán utilizados para el diseño de la arquitectura del *software*.

La primera tarea para la elaboración de los diagramas de flujo de datos es la creación de un Diagrama de Contexto⁶³, en donde el objetivo principal es representar las Entidades Externas, las Entradas de Información, y las Salidas de la Información Procesada que deberá tener la solución.

En la Ilustración 3-6 se presenta el Diagrama de Contexto de la SPO, en donde se puede apreciar que las Entidades Externas que estarán interactuando con la SPO son: Estudiante, Profesor, Secretaria, Subcomisión Académica, Secretaría General, Sistema de Recopilación de Información Universitaria, y SPO Remota.

⁶² En los diagramas de flujo de datos los almacenes de datos, representan cualquier cosa en la que se pueda guardar información; pueden ser almacenes físicos como archivadores, documentos impresos, etc., o pueden ser almacenes digitales como tablas de una base de datos, documentos electrónicos, directorios de archivos, etc. [Ref. 6].

⁶³ El Diagrama de Contexto es el nivel mas alto en un diagrama de flujo de datos, contiene únicamente un proceso (se le asigna el número cero) que representa el sistema completo, muestra todas las entidades externas, así como los flujos de datos principales que entran y salen de él. [Ref. 6].

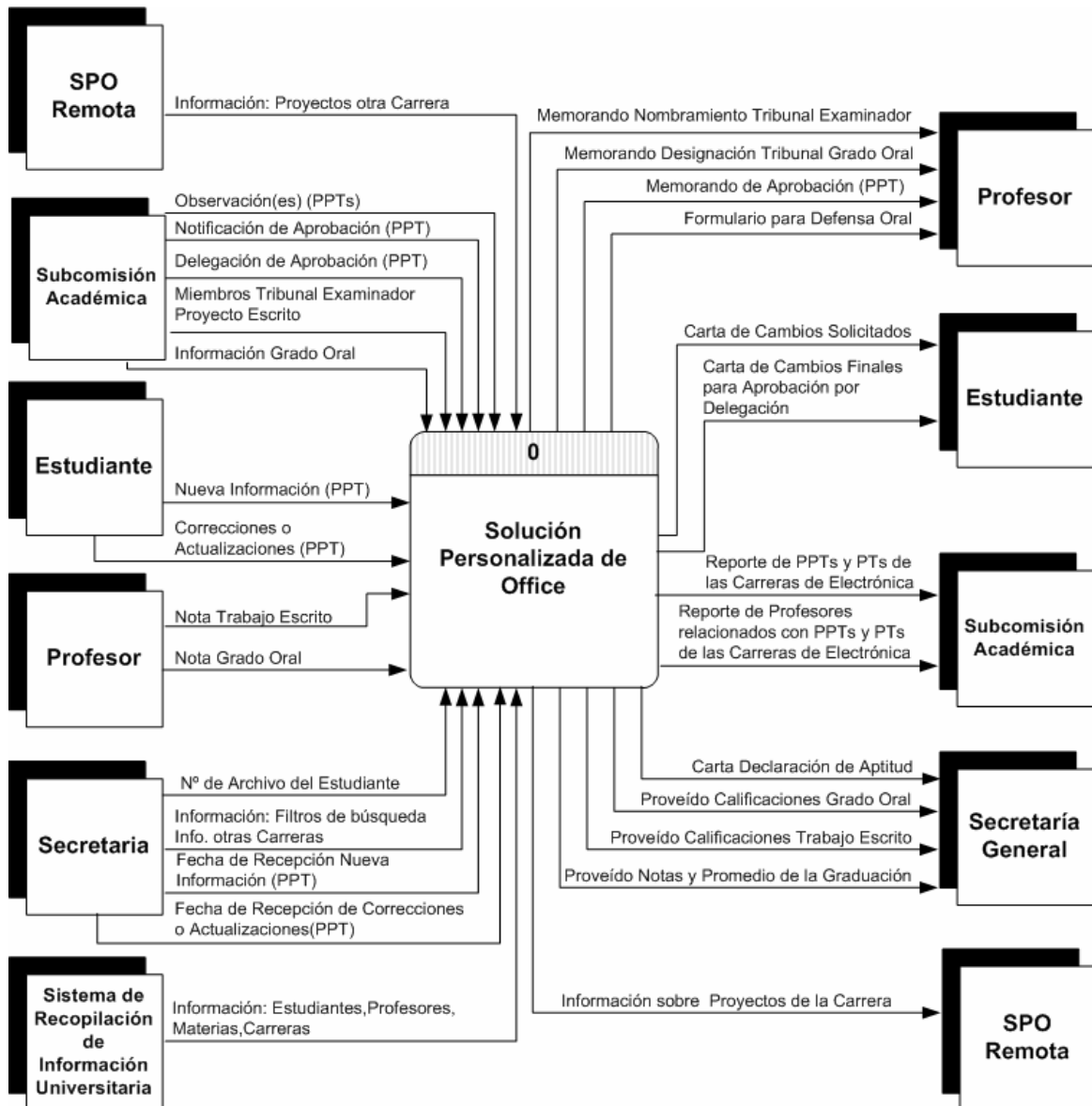


Ilustración 3-6 Diagrama de Contexto de la SPO.

Algunas consideraciones importantes sobre las Entidades Externas son las siguientes:

- La entidad: “Sistema de Recopilación de Información Universitaria”, es la entidad que representa al sistema de Administración Estudiantil; ésta es la entidad que proporcionará la información relacionada con profesores, estudiantes, materias y carreras, de la Base de Datos SAE.
- A pesar de que en el Diagrama de Contexto, tanto en la entrada como en la salida, se presenta una sola entidad “Profesor”, hay que considerar que esta entidad en la práctica aparecerá varias veces, para representar a los distintos profesores que estarán relacionados con un Proyecto de

Titulación. Lo mismo ocurre con la entidad “SPO Remota”, para representar a cada una de las Carreras de las que se consume o a las que se envía información. En la Ilustración 3-7 se presenta un ejemplo de lo descrito (no son todas las posibilidades).

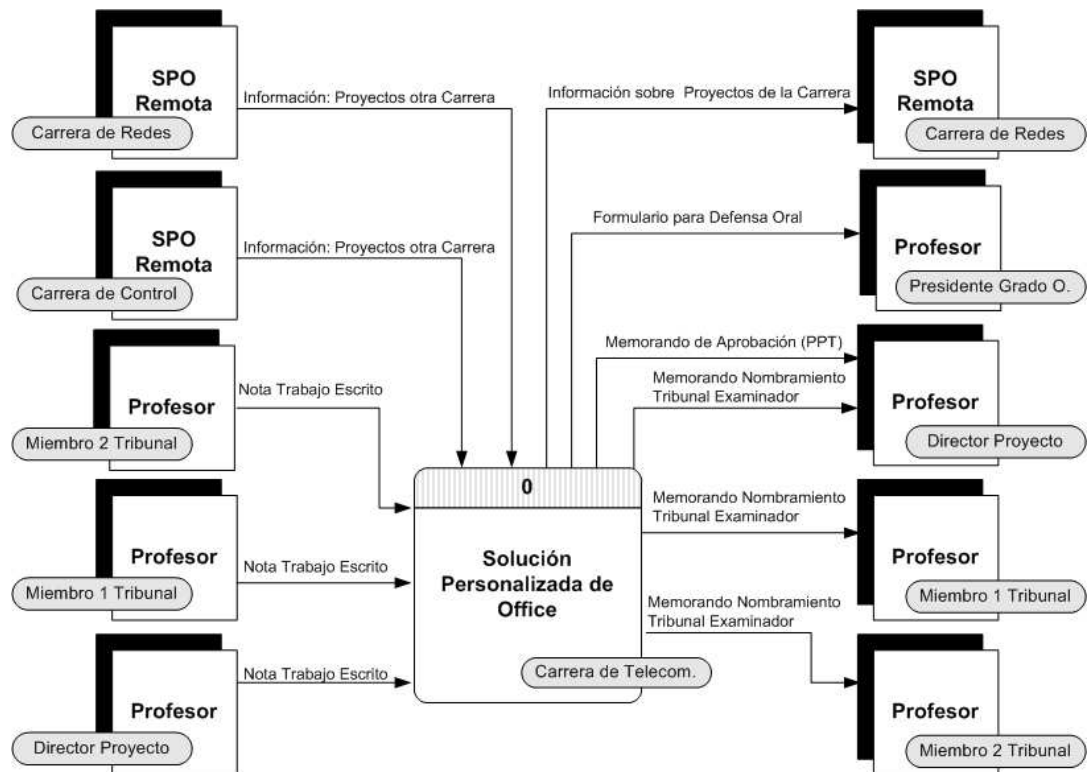


Ilustración 3-7 Ejemplo de la presencia de varias Entidades Externas del mismo tipo.

- La entidad “Estudiante”, a pesar de ser mostrada con un nombre en singular, puede representar a dos estudiantes (para proyectos no individuales). A diferencia de las entidades “Profesor” o “SPO Remota”, la entidad “Estudiante” no aparece en múltiples ocasiones. En la Ilustración 3-7 se pueden apreciar varias Entidades del mismo tipo porque la información contenida en cada uno de los flujos de datos asociados es diferente, lo que no ocurre para la entidad “Estudiante” ya que en caso de que represente a dos estudiantes, hay que considerar que los estudiantes presentan la información en conjunto. En la Ilustración 3-8 se puede ver una muestra gráfica de este particular.

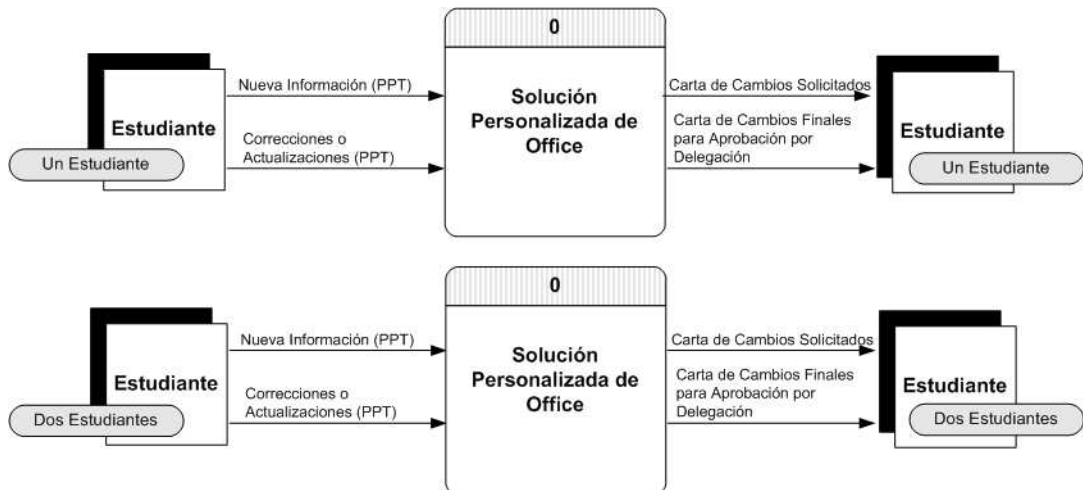


Ilustración 3-8 Consideraciones para la Entidad Estudiante.

- La entidad: “SPO Remota”, contiene en su interior una estructura idéntica a la de la SPO a la que proporciona (como entidad de entrada), o de la que recibe (como entidad de salida) información. Para ilustrar de mejor manera este particular, a continuación se presenta un gráfico que permitirá visualizar de mejor manera lo señalado (relacionar con el Diagrama de Contexto de la Ilustración 3-6).

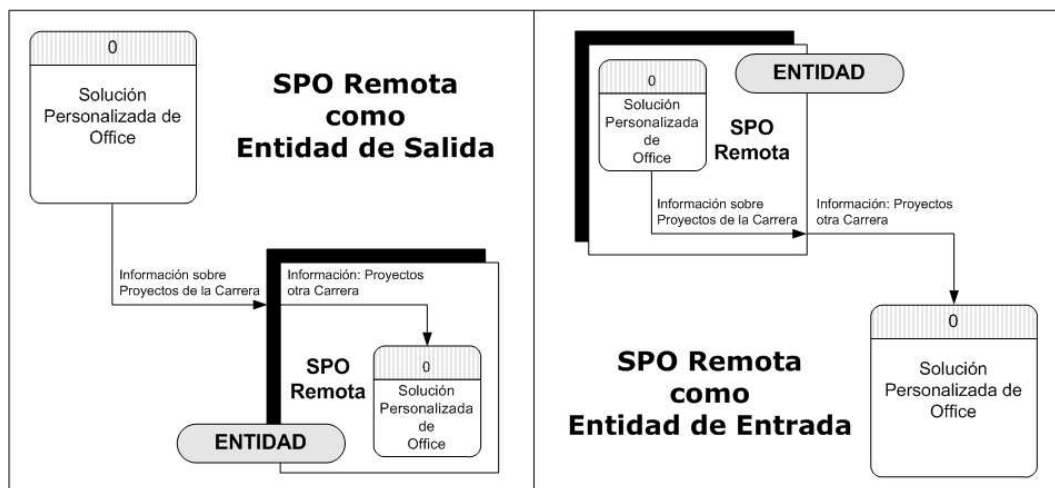


Ilustración 3-9 Entidad Externa SPO Remota.

3.2.2.1.1 Tablas y Campos

Para determinar las tablas y campos de la SPODB, se realizó un Diagrama de Flujo de Datos Lógico (diagrama de nivel 0 o también llamado Diagrama 0)⁶⁴, el

⁶⁴ El Diagrama 0 es la explosión del diagrama de contexto (es el efecto que se obtendría usando una lupa) y puede incluir varios procesos enumerados con un entero, muestra además almacenes de datos y todas las entidades externas y los flujos de datos incluidos en el diagrama de contexto.

mismo que se presentará en esta sección dividido en cuatro segmentos para facilitar su lectura.

En el Diagrama 0 se pueden apreciar varios almacenes de datos los mismos que teóricamente pueden ser de varios tipos (archivos de texto, tablas de una base de datos, etc.); en este caso específico, todos representarán una tabla equivalente de la SPODB, y los flujos de datos asociados servirán para la elaboración del Diccionario de Datos, que es el que permitirá determinar los campos que deben contener cada uno de los registros de dichas tablas.

El Diccionario de Datos será presentado en varias tablas en las cuales se podrán ver los flujos de datos con las entradas del diccionario asociadas. Para comprender mejor la simbología tanto de los DFDs como de los elementos del Diccionario de Datos se recomienda revisar el **Anexo E**.

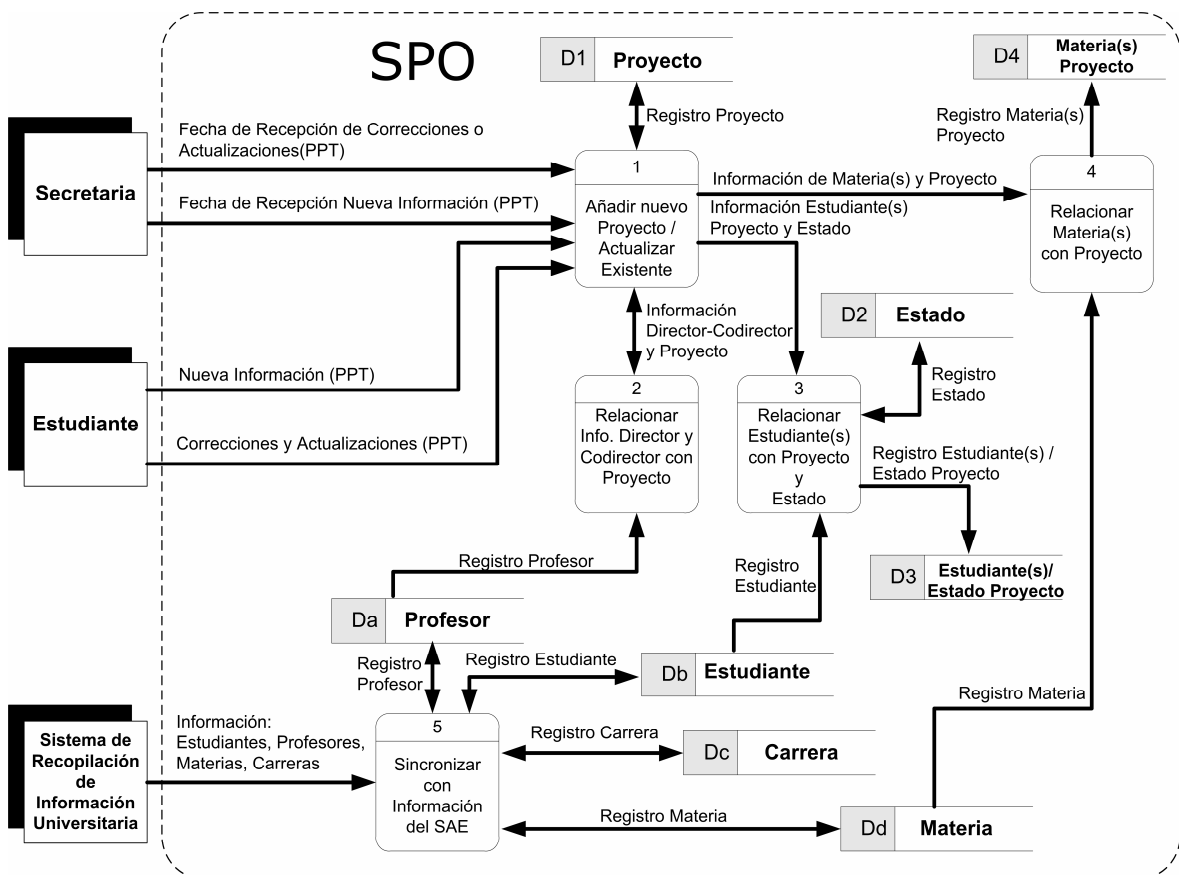


Ilustración 3-10 Diagrama 0 de la SPO (segmento 1/4).

Entradas del Diccionario de Datos (segmento 1)	
Fecha de Recepción de Correcciones o Actualizaciones (PPT)	= Fecha de recepción de correcciones/actualizaciones
Fecha de Recepción Nueva Información (PPT)	= Fecha de recepción de nueva información
Nueva Información (PPT)	= Título del proyecto + Nombre del director del proyecto + (Nombre del codirector del proyecto) + Nombre del estudiante 1 + (Nombre del estudiante 2) + {Materias} *Áreas Técnicas relacionadas*
Correcciones y Actualizaciones (PPT)	= (Título del proyecto) + (Nombre del director del proyecto) + (Nombre del codirector del proyecto) + (Nombre del estudiante 1) + (Nombre del estudiante 2) + ({Materias}) *Áreas técnicas relacionadas*
Información de Materia(s) y Proyecto	= @ Identificador del proyecto + {Nombre Materia} *Áreas técnicas relacionadas*
Registro Materia	= @ Identificador de materia + Nombre Materia
Registro Materia(s) Proyecto	= @ Identificador del proyecto + {@ Identificador de materia} *relacionadas*
Información Estudiante(s) Proyecto y Estado	= @ Identificador del proyecto + Nombre de estudiante 1 + (Nombre de estudiante 2) + Nombre del estado
Registro Estado	= @ Identificador del estado + Nombre del estado
Registro Estudiante	= @ Identificador estudiante + *número único* Nombre completo + (Número de cédula) + (Número de libreta militar) + (Fecha de nacimiento) + (Dirección del domicilio) + (Teléfono de contacto principal) + (Teléfono de contacto secundario) + (Dirección de correo electrónico) + (Estado civil) + (Sexo) + (Dirección del archivo de imagen) *foto*
Registro Estudiante(s) / Estado Proyecto	= @ Identificador del proyecto + {@ Identificador estudiante} + *estudiante 1/(2)* @ Identificador del estado
Información Director-Codirector y Proyecto	= @ Identificador del proyecto + Nombre del director del proyecto + (Nombre del codirector del proyecto)
Registro Carrera	= @ Identificador de la carrera + Nombre de la carrera + Escuela a la que pertenece + Alias de la carrera + @ Identificador profesor + *coordinador carrera* @ Identificador profesor + *director de escuela* Nombre de la secretaria + (Teléfono principal) + (Teléfono secundario) + (Dirección página Web principal) + (Dirección página Web secundaria) + (Dirección del archivo de imagen) *foto*

Registro Profesor	= @ Identificador profesor + Nombre profesor + (Número de cédula) + (Fecha de nacimiento) + (Dirección del domicilio) + (Teléfono de contacto) + (Dirección de correo electrónico) + (Título Académico) + (Tipo de contrato) + (Sexo) + (Dirección del archivo de imagen) *foto*
Registro Proyecto	= @ Identificador del proyecto + Título del proyecto + @ Identificador profesor + *director proyecto* (@ Identificador profesor) + *codirector proyecto* Fecha de recepción de nueva información + (Fecha de recepción de correcciones/actualizaciones)

Tabla 3-16 Diccionario de Datos para la SPO (1/3).

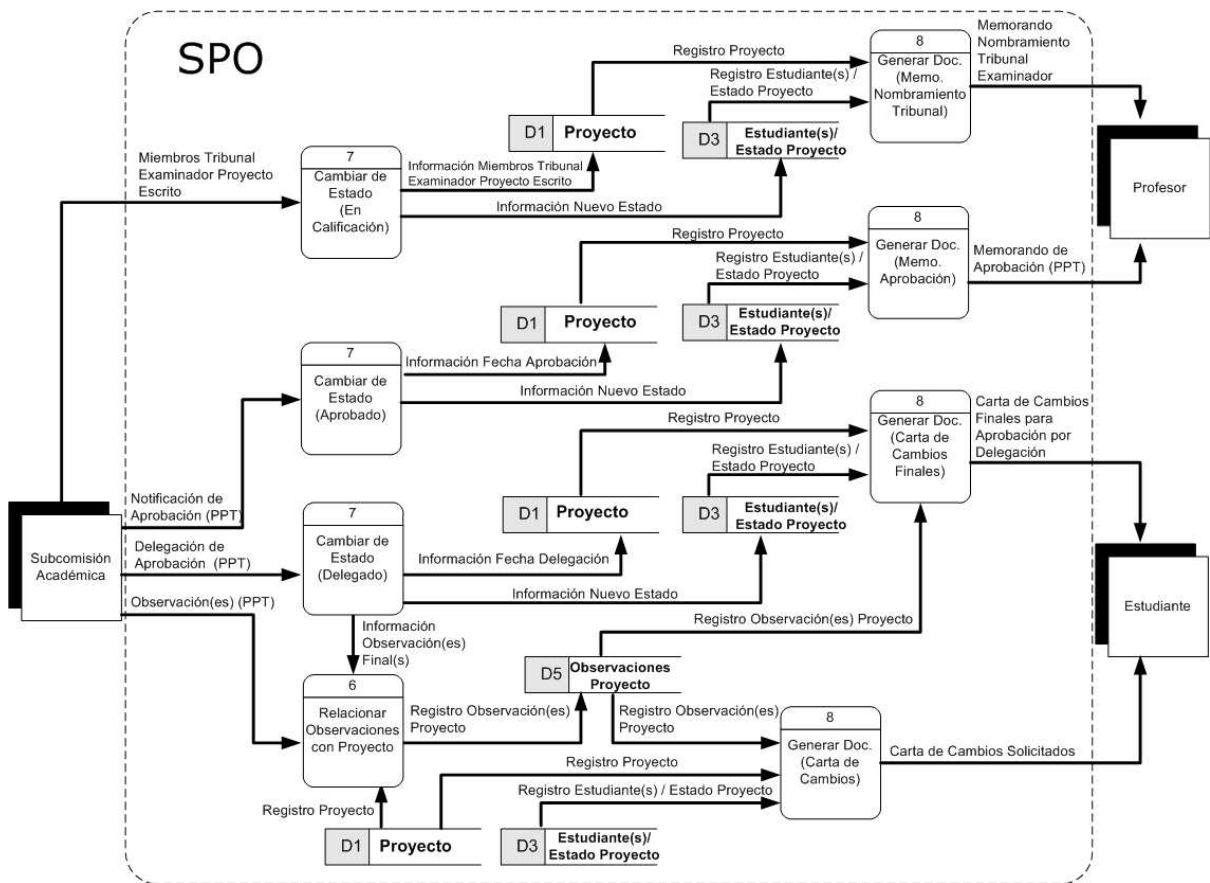


Ilustración 3-11 Diagrama 0 para la SPO (segmento 2/4).

Entradas del Diccionario de Datos (segmento 2)

Miembros Tribunal Examinador Proyecto Escrito	= Nombre miembro(1) tribunal trabajo escrito + Nombre miembro(2) tribunal trabajo escrito + Fecha designación del tribunal
Información Miembros Tribunal Examinador Proyecto Escrito	= @ Identificador del proyecto + @ Identificador profesor + *tribunal1 trabajo escrito*

	=	@ Identificador profesor + *tribunal2 trabajo escrito* Fecha designación del tribunal
Notificación de Aprobación (PPT)	=	Fecha de la aprobación
Información Fecha Aprobación	=	@ Identificador del proyecto + [Delegado No Delegado]+ *Indicador de delegación* Fecha de la aprobación
Delegación de Aprobación (PPT)	=	Fecha de la delegación
Información Fecha Delegación	=	@ Identificador del proyecto + [Delegado No Delegado]+ *Indicador de delegación* Fecha de la delegación
Observación(es) (PPT)	=	{Observaciones sobre el PPT} + Fecha de la sesión de Subcomisión Académica
Información Observación(es) Final(es)	=	{Observaciones finales sobre el PPT} + Fecha de la sesión de Subcomisión Académica
Registro Observación(es) Proyecto	=	@ Identificador del proyecto + {Observaciones sobre el PPT} + [Final No Final] + *tipo de observación* Fecha de la sesión de Subcomisión Académica
Información Nuevo Estado	=	@ Identificador del proyecto + @ Identificador del estado
Registro Proyecto	=	@ Identificador del proyecto + Título del proyecto + @ Identificador profesor + *director proyecto* (@ Identificador profesor) + *codirector proyecto* Fecha de recepción de nueva información + (Fecha de recepción de correcciones/actualizaciones) (Fecha de aprobación) + ([Delegado No Delegado])+ *Indicador de delegación* (Fecha de la delegación) + (Fecha designación de tribunal) + (@ Identificador profesor) + *tribunal1 trabajo escrito* (@ Identificador profesor) *tribunal2 trabajo escrito*

Tabla 3-17 Diccionario de Datos para la SPO (2/3).

Entradas del Diccionario de Datos (segmento 3)		
Número de Archivo del Estudiante	=	Número de archivo *Num. Folio Estudiante*
Nota Grado Oral	=	Nota promedio del grado oral
Información Nota Grado Oral y Nuevo Estado	=	@ Identificador del proyecto + @ Identificador estudiante + @ Identificador del estado + Nota promedio del grado oral
Información Grado Oral	=	Nombre del presidente del grado oral + Nombre del estudiante + Nombre del miembro1 tribunal grado oral + Nombre del miembro2 tribunal grado oral + Fecha grado oral + Hora grado oral
Información Grado Oral y Nuevo Estado	=	@ Identificador del proyecto + @ Identificador estudiante + *estudiante 1* (@ Identificador estudiante) + *estudiante 2* @ Identificador profesor+ *presidente grado oral* @ Identificador profesor + *director* @ Identificador profesor + *tribunal grado oral 1* @ Identificador profesor + *tribunal grado oral 2* @ Identificador del estado + Fecha grado oral

Nota Trabajo Escrito	= Nota trabajo escrito -director + Nota trabajo escrito -miembro1 tribunal + Nota trabajo escrito -miembro2 tribunal
Información Nota Trabajo Escrito y Nuevo Estado	= @ Identificador del proyecto + Nota director trabajo escrito + Nota miembro1 tribunal trabajo escrito + Nota miembro2 tribunal trabajo escrito
Registro Estudiante(s) / Estado Proyecto	= @ Identificador del proyecto + {@ Identificador estudiante} + *estudiante 1/(2)* @ Identificador del estado (Nota director trabajo escrito) + (Nota miembro1 tribunal trabajo escrito) + (Nota miembro2 tribunal trabajo escrito) + (@ Identificador profesor)+ *presidente grado oral* (@ Identificador profesor)+ *tribunal grado oral 1* (@ Identificador profesor)+ *tribunal grado oral 2* (Fecha grado oral) + (Nota promedio grado oral) + (Número de archivo)

Tabla 3-18 Diccionario de Datos para la SPO (3/3).

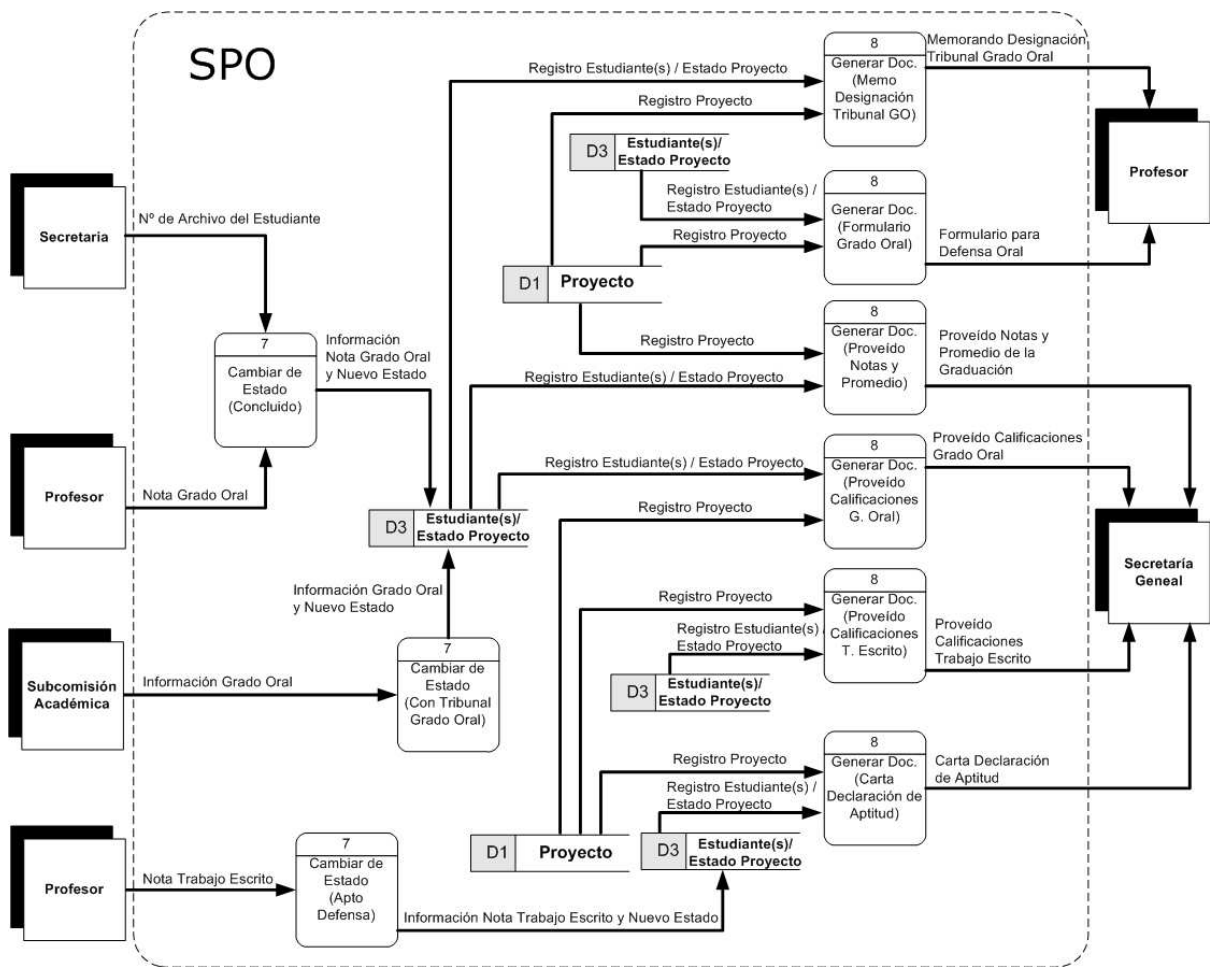


Ilustración 3-12 Diagrama 0 de la SPO (segmento 3/4)

En la Tabla 3-16 se presentaron varias entradas del Diccionario de Datos asociadas a los flujos de datos “Registro Proyecto”, y “Registro Estudiante(s) /Estado Proyecto”, se podría decir que las entradas que se presentan en esta tabla, son las básicas asociadas a dichos flujos de datos.

Tanto en la Tabla 3-17 como en la Tabla 3-18 se agregan entradas adicionales a las denominadas entradas básicas, las mismas que tienen un calificativo de opcional (encerradas entre paréntesis '()') en vista de que son datos que se van obteniendo según el cambio de estado del proyecto (ej. nunca estarán presentes en un registro nuevo); estas entradas adicionales son el resultado del análisis de los flujos de datos y procesos incluidos en los segmentos 2 y 3 del Diagrama 0 de la SPO [Ilustración 3-11 y Ilustración 3-12].

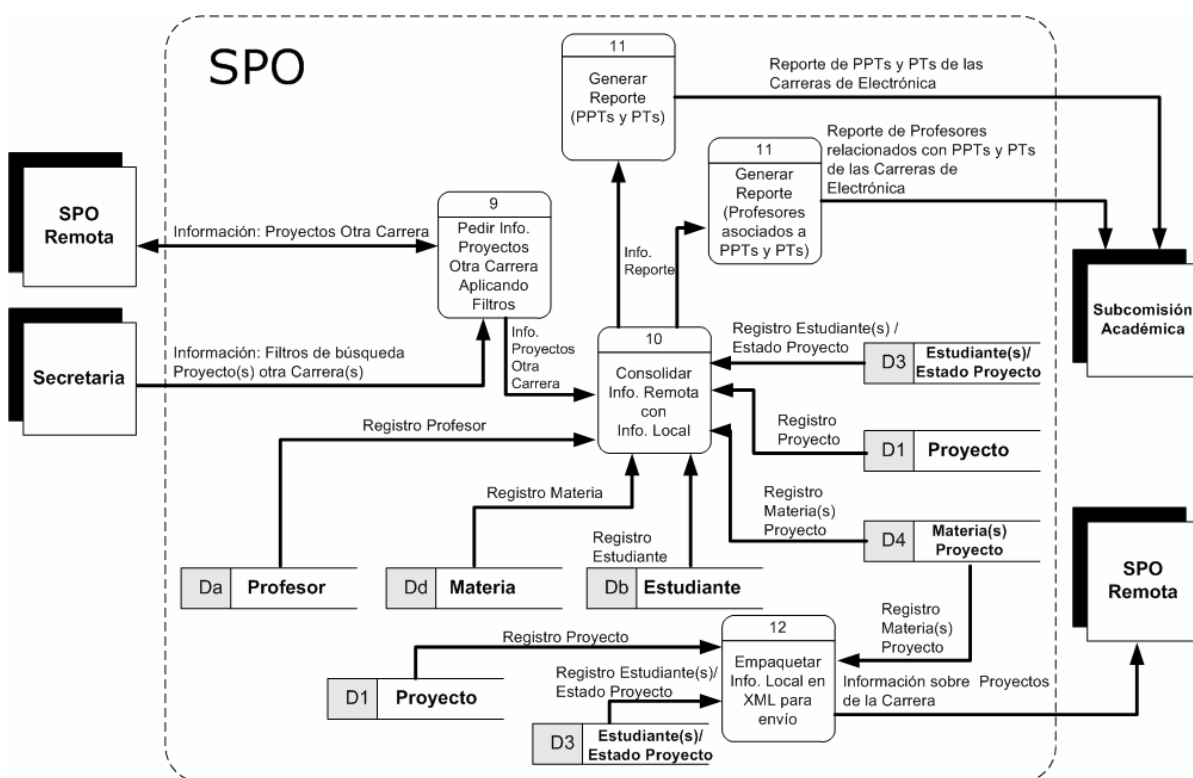


Ilustración 3-13 Diagrama 0 para la SPO (segmento 4/4)

El cuarto segmento el Diagrama 0 para la SPO, se presenta en la Ilustración 3-13, pero a razón de que no aporta ninguna información adicional para el diseño de la

Base de Datos (tópico de esta Sección), no se presentaran las entradas del Diccionario de Datos correspondientes a los flujos de datos de este segmento⁶⁵.

Como se puede observar en la Ilustración 3-12, todos los datos relacionados con las notas así como los relacionados con el Grado Oral del Estudiante, no fluyen directamente al almacén de datos **Proyecto**, sino mas bien al almacén de datos **Estudiante(s)/Estado Proyecto**.

El justificativo para realizar el diseño con esta lógica, es que si bien es cierto que: se aprueba, califica, se hacen observaciones, o se designa tribunal calificador para un Proyecto de Titulación, hay que tener muy presente, que las notas se ponen al estudiante, así como también se designan presidente y tribunal calificador para el grado oral del estudiante, con lo cual se puede concluir que existen datos que pertenecen al proyecto (tema, director, fecha de aprobación, observaciones, etc.), pero también existen otros datos que a pesar de tener mucha relación con el proyecto, pertenecen al estudiante (notas del proyecto escrito, nota promedio de la defensa oral, etc.).

Esta aclaración tal vez permita entender mejor el porqué del nombre **Estudiante(s)/Estado Proyecto** para uno de los almacenes de datos. A simple vista se podría decir que un proyecto debería tener asociado un estado, por lo cual el estado es un dato que pertenece al proyecto. Sin embargo, esto no es del todo cierto, ya que hay que tener en consideración que los proyectos pertenecen a los estudiantes, y el estudiante tiene un proyecto en un determinado estado.

Este particular se puede observar con claridad en los Proyectos de Titulación presentados por dos estudiantes; por ejemplo, **A** y **B** presentan un plan de proyecto; el plan es aprobado, por lo tanto **A** y **B** tienen un proyecto en estado aprobado; luego de un determinado tiempo **A** y **B** están listos para defender el proyecto (grado oral independiente); **B** tiene que viajar y estará fuera del país por seis meses; **A** no puede esperar y decide pedir que nombren el presidente y

⁶⁵ El cuarto segmento al igual que los otros segmentos que componen el Diagrama 0 para la SPO con sus flujos de datos y procesos, serán utilizados más adelante para el diseño de la arquitectura del *software*.

tribunal para su grado oral; **A** defiende el proyecto y tiene un proyecto concluido, mientras que **B** a pesar de estar hablando del mismo proyecto, lo tiene en un estado anterior, por lo tanto el estado del proyecto es un dato que pertenece al estudiante.

Si bien es cierto que en la generalidad de los casos, cuando existen proyectos de dos estudiantes prácticamente los únicos datos que cambian entre ellos son: las notas del grado oral, la hora de la defensa oral y el número de archivo del estudiante, siempre existirá la posibilidad de que varíen otros datos; como por ejemplo, la fecha de la defensa, el presidente que preside el grado oral, las notas del trabajo escrito (no muy frecuente), etc., por esta razón fue muy importante analizar con detenimiento en la lógica de la solución cuales datos pertenecen o fluyen con relación al estudiante y cuales lo hacen con relación al proyecto, y el resultado se ve reflejado tanto en el Diagrama 0 como en las entradas del Diccionario de Datos.

Una vez presentados todos los segmentos del Diagrama 0 para la SPO, se pueden identificar claramente nueve almacenes de datos: **Proyecto, Profesor, Estudiante, Carrera, Materia, Estado, Materia(s) Proyecto, Observación(es) Proyecto, y Estudiante(s)/Estado Proyecto.**

Cada uno de estos almacenes de datos estarán incluidos en la SPODB, de los cuales **Profesor, Estudiante, Carrera y Materia** serán compatibles con los almacenes de datos de la Base de Datos SAE, los factores por los cuales no se utilizarán directamente los almacenes de datos del SAE son los siguientes:

- En la secretaría de cada una de las Carreras de Electrónica, existe una copia autorizada de la Base de Datos SAE, la misma que es reemplazada semestralmente (información por períodos académicos), y que es con la cual podría interactuar la SPO. El problema fundamental que se presenta, es que todos aquellos estudiantes que concluyan su Proyecto de Titulación, al semestre siguiente de su graduación ya no estarán registrados en la mencionada copia del SAE (su información se encuentra únicamente en la base de datos SAE principal) lo que dificultaría la

posibilidad de llevar una historia completa sobre los proyectos de titulación (uno de los objetivos de la SPO).

- La copia de la Base de Datos SAE no puede ser manipulada en su estructura por ninguna persona no autorizada, razón por la cual no es posible agregar procesos almacenados o campos adicionales que sean requeridos.
- La Base de Datos SAE posee mucha información que no es relevante para la SPO.

Es importante hacer énfasis que el Diagrama 0 [Ilustración 3-11 a Ilustración 3-13], no incluye la “lógica de la tecnología inteligente”, de ser así, los procesos relacionados con la generación de documentos (proceso 8) y con la generación de reportes (proceso 11), también deberían incluir un flujo de datos que los asocie a cada uno de los almacenes de datos: **Profesor**, **Estudiante**, **Carrera** y **Materia**, en vista de que ésta es la parte lógica de la SPO encargada de proporcionar la ayuda relativa al contexto de los documentos que se auto generarán.

Básicamente, esta ayuda dependerá de la información personal tanto de los profesores como de los estudiantes, así como también de información propia de cada una de las carreras y materias, de ahí la importancia de todas las entradas del Diccionario de Datos asociadas con estos almacenes de datos.

En la Tabla 3-19 se muestran las entradas del Diccionario de Datos asociadas a los flujos de datos relacionados con cada uno de los almacenes de datos.

ALMACEN DE DATOS	ENTRADAS DEL DICCIONARIO ASOCIADAS
D1 – Proyecto	@ Identificador del proyecto Título del proyecto @ Identificador profesor *director proyecto* (@ Identificador profesor) *codirector proyecto* Fecha de recepción de nueva información (Fecha de recepción de correcciones/actualizaciones) (Fecha de aprobación) ([Delegado No Delegado]) * Indicador de delegación * (Fecha de la delegación) (Fecha designación de tribunal) (@ Identificador profesor) *tribunal1 trabajo escrito* (@ Identificador profesor) *tribunal2 trabajo escrito*
D2 – Estado	@ Identificador del estado Nombre del estado

D3 – Estudiante (s) / Estado Proyecto	@ Identificador del proyecto {@ Identificador estudiante} *estudiante 1/(2)* @ Identificador del estado (Nota director trabajo escrito) (Nota miembro1 tribunal trabajo escrito) (Nota miembro2 tribunal trabajo escrito) (@ Identificador profesor) *presidente grado oral* (@ Identificador profesor) *tribunal grado oral 1* (@ Identificador profesor) *tribunal grado oral 2* (Fecha grado oral) (Nota promedio grado oral) (Número de archivo)
D4 – Materia (s) Proyecto	@ Identificador del proyecto {@ Identificador de materia} *relacionadas al proyecto*
D5 – Observación (es) Proyecto	@ Identificador del proyecto {Observaciones sobre el PPT} [Final No Final] *tipo de observación* Fecha de la sesión de Subcomisión Académica
Da – Profesor	@ Identificador profesor Nombre profesor (Número de cédula) (Fecha de nacimiento) (Dirección del domicilio) (Teléfono de contacto) (Dirección de correo electrónico) (Título Académico) (Tipo de contrato) (Sexo) (Dirección del archivo de imagen) *foto*
Db – Estudiante	@ Identificador estudiante *número único* Nombre completo (Número de cédula) (Número de libreta militar) (Fecha de nacimiento) (Dirección del domicilio) (Teléfono de contacto principal) (Teléfono de contacto secundario) (Dirección de correo electrónico) (Estado civil) (Sexo) (Dirección del archivo de imagen) *foto*
Dc – Carrera	@ Identificador de la carrera Nombre de la carrera Escuela a la que pertenece Alias de la carrera @ Identificador profesor *coordinador carrera* @ Identificador profesor *director de escuela* Nombre de la secretaria (Teléfono principal) (Teléfono secundario) (Dirección página Web principal) (Dirección página Web secundaria) (Dirección del archivo de imagen) *foto*
Dd – Materia	@ Identificador de materia Nombre Materia

Tabla 3-19 Almacenes de Datos y Entradas del Diccionario de Datos asociadas.

Los nombres que se darán a los almacenes de datos presentados en la Tabla 3-19, ya vistos como parte de la SPODB, y una pequeña descripción del contenido de cada uno de ellos se muestra en la Tabla 3-20.

ALMACEN DE DATOS	TABLA SPODB	DESCRIPCIÓN
Proyecto	T_Proyecto	Contiene información relativa al proyecto como: título, director, etc.
Estado	T_Estado_Proyecto	Contiene información de los posibles estados de un proyecto.
Estudiante(s)/ Estado Proyecto	T_Proyecto_Estudiante_Estado	Contiene la información de la relación entre los estudiantes con los proyectos y el estado de los mismos.
Materia(s) Proyecto	T_Proyecto_Materia	Contiene información de la relación de las materias con los proyectos.
Observación(es) Proyecto	T_Proyecto_Observación	Contiene la información de las observaciones realizadas a los proyectos.
Profesor	T_Profesor	Contiene información personal de los profesores.
Estudiante	T_Estudiante	Contiene información personal de los estudiantes.
Carrera	T_Carrera	Contiene información general de la carrera.
Materia	T_Materia	Contiene información general de las materias.

Tabla 3-20 Almacenes de datos y sus Tablas equivalentes en la SPODB.

Una vez conocidas las tablas que conformarán la SPODB, el siguiente paso es determinar los campos de cada una de ellas, y para ello se utilizará la información de la Tabla 3-19.

T_PROYECTO		
ENTRADAS DEL DICCIONARIO ASOCIADAS	CAMPOS	
@ Identificador del proyecto	pry_cod	Texto
Título del proyecto	pry_titulo	Texto
@ Identificador profesor *director proyecto*	pry_cod_pro_dir	Texto
(@ Identificador profesor) *codirector proyecto*	pry_cod_pro_codir	Texto / NULL
Fecha de recepción de nueva información	pry_fec_pres	Fecha
(Fecha recepción de correcciones/actualizaciones)	pry_fec_pres_cbos	Fecha / NULL
(Fecha de aprobación)	pry_fec_aprob	Fecha / NULL
([Delegado] No Delegado) *Indicador delegación*	pry_tipo_aprob	[Delegado No Delegado]
(Fecha de la delegación)	pry_fec_delega	Fecha / NULL
(Fecha designación de tribunal)	pry_fec_des_tribe	Fecha / NULL
(@ Identificador profesor) *tribunal1 trabajo escrito*	pry_cod_pro_tribe1	Texto / NULL
(@ Identificador profesor) *tribunal2 trabajo escrito*	pry_cod_pro_tribe2	Texto / NULL

T_CARRERA		
@ Identificador de la carrera	car_cod	Texto
Nombre de la carrera	car_nom	Texto
Escuela a la que pertenece	car_escuela	Texto
Alias de la carrera	car_alias	Texto
@ Identificador profesor *coordinador carrera*	car_codcord	Texto
@ Identificador profesor *director de escuela*	car_coddirescu	Texto
Nombre de la secretaria	car_nomsecre	Texto
(Teléfono principal)	car_fono1	Texto / NULL
(Teléfono secundario)	car_fono2	Texto / NULL
(Dirección página Web principal)	car_webpage1	Texto / NULL
(Dirección página Web secundaria)	car_webpage2	Texto / NULL
(Dirección del archivo de imagen) *foto*	car_pathfoto	Texto / NULL
T_MATERIA		
@ Identificador de materia	mat_cod	Texto
Nombre Materia	mat_nom	Texto

Tabla 3-21 Entradas del Diccionario de Datos y sus equivalentes en las Tablas de la SPODB.

Como se puede apreciar en la Tabla 3-21, las entradas del Diccionario de Datos también permiten determinar los campos que pueden contener un valor NULL, todos aquellos que tengan el calificativo de opcional podrán hacerlo, mientras que los demás no.

3.2.2.1.2 Diagramas Entidad – Relación

Uno de los métodos más utilizados para el diseño de bases de datos, son los diagramas de Entidad-Relación; en este caso, estos diagramas no son utilizados con la finalidad de diseñar la base de datos, sino más bien como una ayuda para explicar visualmente los resultados obtenidos con los diagramas de flujo y el diccionario de datos.

En un principio se van a explicar las relaciones que existen entre las siguientes entidades: Proyecto, Estudiante, Estado del Proyecto, Observación y Materia, para lo cual se utilizará la Ilustración 3-14, de donde se puede interpretar:

- Un proyecto puede tener muchas observaciones, relación 1:M
- Un proyecto puede tener muchas materias relacionadas, y a su vez muchos proyectos pueden estar relacionados con la misma materia, relación M:M

- Un proyecto puede ser presentado por varios estudiantes (se considera el caso de los proyectos de dos estudiantes), y a su vez un estudiante puede presentar varios proyectos (se considera el caso de los proyectos negados o abandonados por los estudiantes), relación M:M
- Varios estudiantes pueden tener proyectos que se encuentran en un mismo estado, y a su vez un estudiante puede tener varios proyectos que se encuentran en diferentes estados (ej. estudiante con un proyecto negado, y un proyecto en estado de aprobación), relación M:M

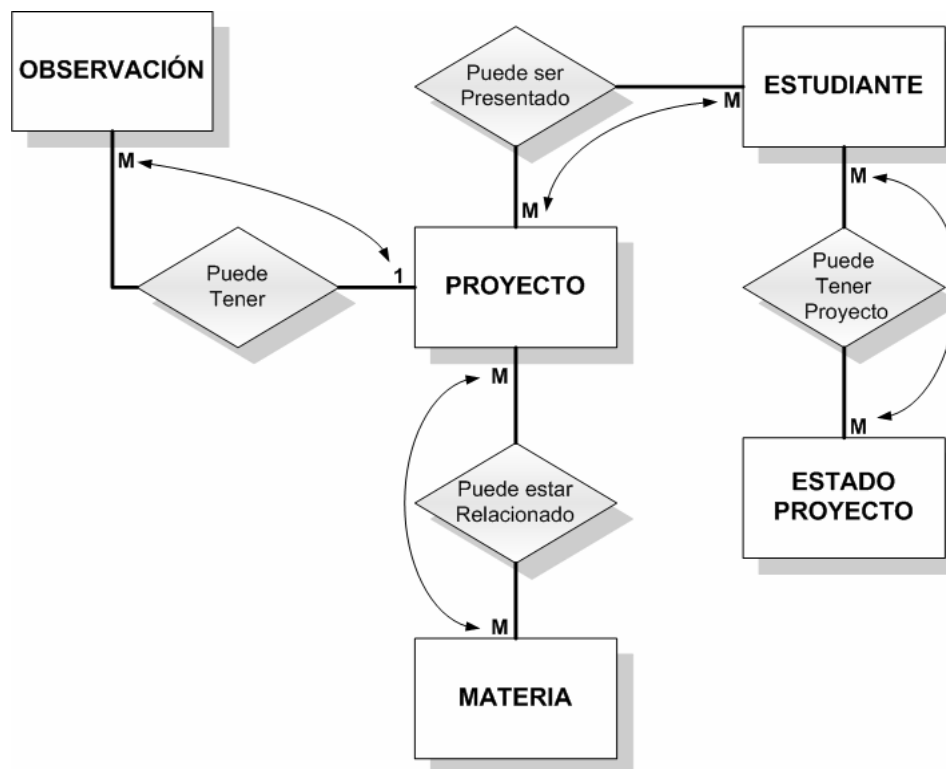


Ilustración 3-14 Entidad Relación para la SPODB (1/3).

Si en la Ilustración 3-14, se reemplazan las tablas de la SPODB, se podrá apreciar que la Tabla **T_PROYECTO_ESTUDIANTE_ESTADO**, maneja las dos relaciones Proyecto → Estudiante, y Estudiante → Estado del Proyecto, como se muestra en la Ilustración 3-15.

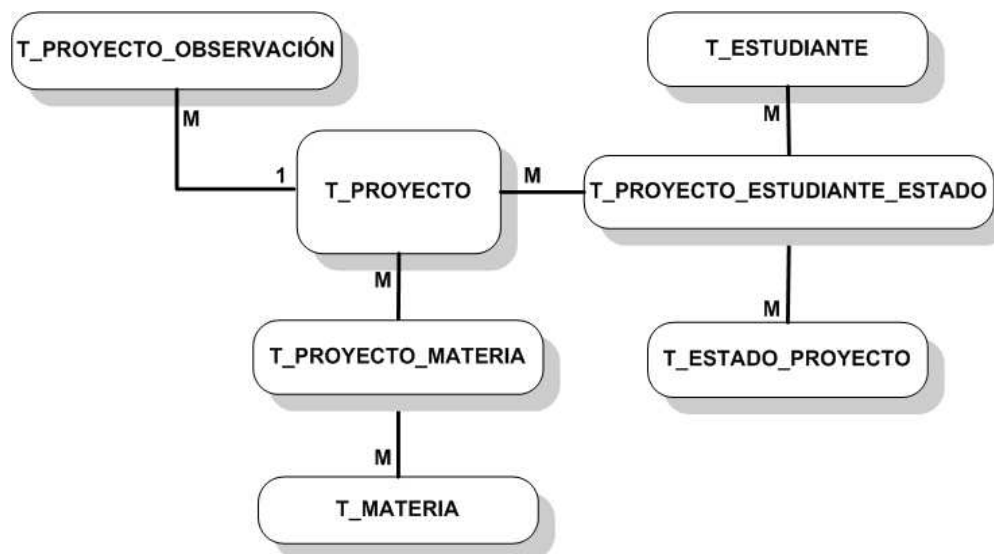


Ilustración 3-15 Entidad Relación para la SPODB (2/3).

Para explicar la relación existente entre las entidades: Proyecto, Estudiante y Profesor, se utilizará la Ilustración 3-16.

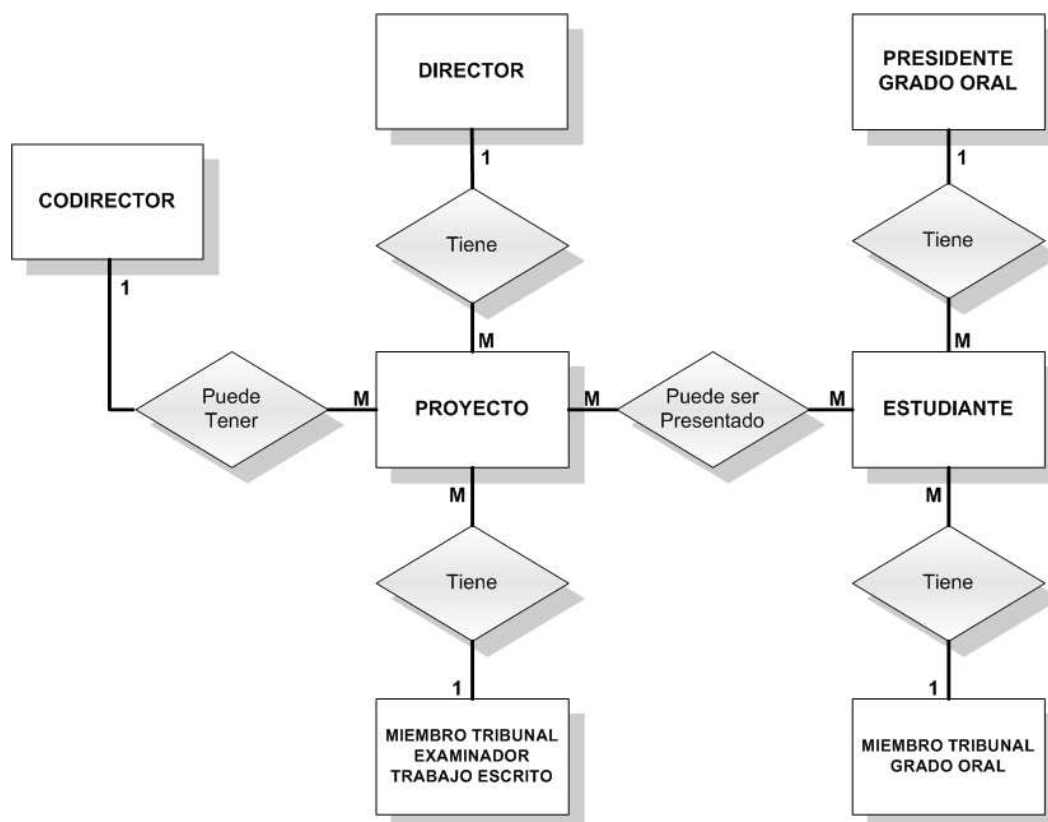


Ilustración 3-16 Entidad Relación para la SPODB (3/3).

De la Ilustración 3-16 se puede interpretar:

- Varios proyectos pueden tener el mismo director, relación M:1
- Varios proyectos pueden tener el mismo codirector (opcional), relación M:1
- Varios proyectos pueden tener el mismo miembro del tribunal examinador del trabajo escrito
- Distintos estudiantes pueden tener el mismo presidente en el grado oral
- Distintos estudiantes pueden tener el mismo miembro de tribunal en el grado oral

Hay que tener presente que tanto para el trabajo escrito como para el grado oral, el tribunal está compuesto por dos miembros (profesores) adicionales al director del proyecto, razón por la cual a pesar de que en la Ilustración 3-16 se muestre una sola entidad que representa a estos profesores, en la práctica serán dos, esto se puede apreciar mejor en la Ilustración 3-17 que muestra el diagrama final (tablas, campos y relaciones) de la SPODB, elaborado en MS SQL Server 2000 (DBMS que se utilizará para la SPO).

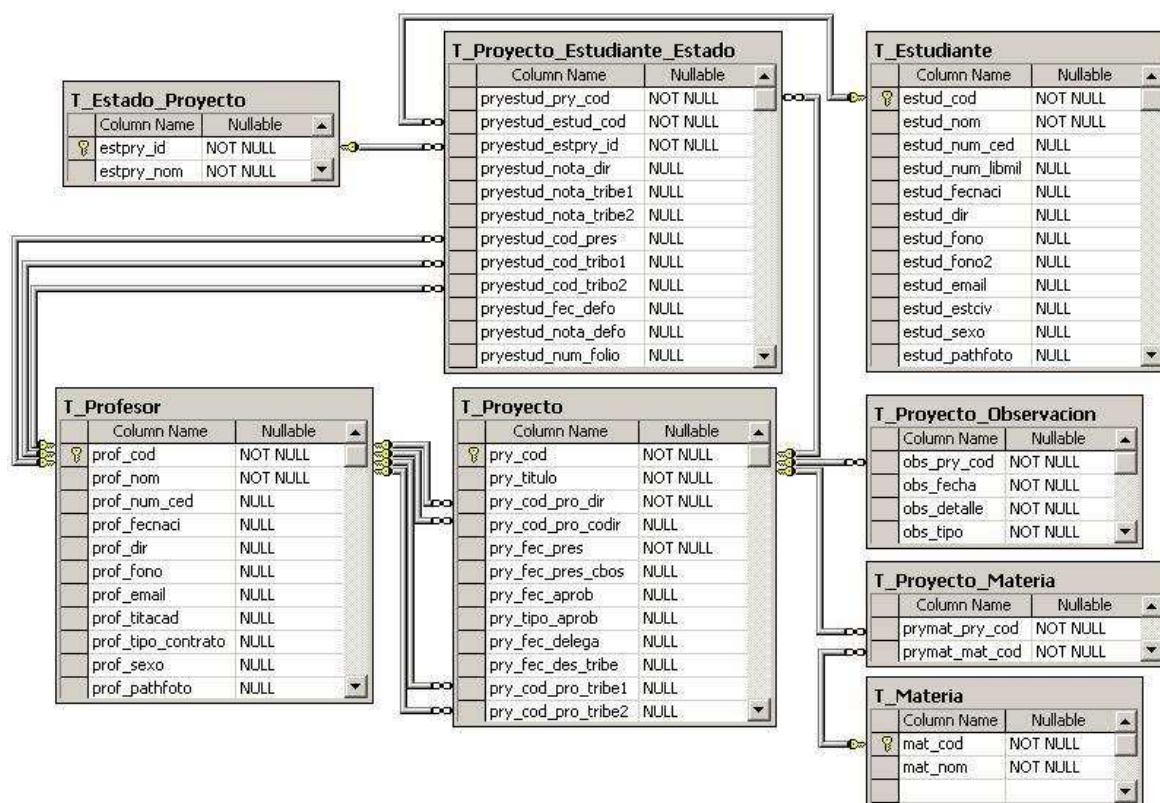


Ilustración 3-17 Diagrama de la SPODB.

Si se analizan con detenimiento la Ilustración 3-17 y la Tabla 3-21, se pueden sacar las siguientes conclusiones:

- Todos los campos cuyas equivalencias en las entradas del Diccionario de Datos tienen el calificador de identificador (están precedidos del signo @), son un punto de relación entre las tablas.
- Todas aquellas tablas que poseen campos cuyas equivalencias en las entradas del Diccionario de Datos tienen un calificador de iteración⁶⁶ (están encerrados entre llaves '{ }'), serán un puente de una relación de muchos a muchos cuando el elemento iterado sea un identificador (ej. {@Identificador estudiantes}), o tendrán asociada una relación de uno a mucho si el elemento iterado no es un identificador (ej. {Observaciones sobre el PPT}).

La SPODB, como se puede apreciar en la Tabla 3-21 también contendrá una tabla llamada 'T_Carrera', la misma que no estará relacionada con ninguna de las otras tablas, a pesar de que en la base de datos SAE la tabla que representa a las carreras, está relacionada de alguna manera con las otras tablas (profesores, estudiantes, materias); dentro de la SPO no es necesario mantener estas relaciones debido a que la información que se importará a la SPODB será exclusivamente la de la carrera donde esté implementada la solución.

Esto quiere decir que, por ejemplo, en el computador de Electrónica y Redes, la SPODB contendrá la información exclusiva de la carrera de Redes, lo mismo ocurrirá en Telecomunicaciones y Control, cuando alguna carrera requiera información de cualquiera de las otras, la SPO a la que se le solicita dicha información, será la encargada de proporcionar todos los datos necesarios, incluyendo los relativos a estudiantes, materias y profesores.

La tabla 'T_Carrera', con todos sus campos y los tipos de datos de cada uno de ellos se muestra en la Ilustración 3-18.

⁶⁶ Los elementos con un calificador de iteración, indican la existencia de varios registros con datos comunes, en donde el elemento distintivo entre los registros es justamente aquel que tiene este calificador; por ejemplo, si existen dos registros A (cod_pry=2 y cod_estud=3) y B (cod_pry=2 y cod_estud =4) el elemento de iteración será cod_estud.

T_Carrera				
	Column Name	Data Type	Length	Allow Nulls
	car_cod	char	3	
	car_nom	char	60	✓
	car_escuela	varchar	30	✓
	car_alias	char	10	✓
	car_codcord	char	11	✓
	car_coddirescu	char	11	✓
	car_nomsecre	nvarchar	60	✓
	car_fono1	nvarchar	15	✓
	car_fono2	nvarchar	15	✓
	car_webpage1	nvarchar	60	✓
	car_webpage2	nvarchar	60	✓
	car_pathfoto	nvarchar	60	✓

Ilustración 3-18 Tabla 'T_Carrera' de la SPODB.

Para concluir la parte del diseño de la SPODB, sólo falta mencionar que en la "lógica inteligente de la SPO", existe un proceso que permite proteger ciertas secciones de los documentos que se auto generan (en función de las etiquetas XML incluidas en cada uno de los mismos).

Este proceso requiere almacenar la información de un permiso de edición que el usuario configurará para cada uno de los documentos, se decidió incluir esta información en la SPODB, para lo cual se agregó una tabla con el nombre de 'T_Configuracion', la cual tendrá un campo para almacenar un nombre de usuario otro para almacenar una contraseña y un campo de un bit por cada documento, que servirá para saber si el usuario tiene permisos de edición sobre el mismo, tal como se muestra en la Ilustración 3-19.

T_Configuracion				
	Nombre de columna	Tipo de datos	Longitud	bitir valores r
	cfg_nomUsuario	nvarchar	50	✓
	cfg_Password	nvarchar	50	✓
	cfg_protectAprobacion	bit	1	✓
	cfg_protectAptitud	bit	1	✓
	cfg_protectFrmOral	bit	1	✓
	cfg_protectProvCalif	bit	1	✓
	cfg_protectProvGraduacion	bit	1	✓
	cfg_protectProvOral	bit	1	✓
	cfg_protectTribEscrito	bit	1	✓
	cfg_protectTribOral	bit	1	✓

Ilustración 3-19 Tabla 'T_Configuracion' de la SPODB.

3.2.2.2 Archivos de Datos

En vista de que la SPO tiene como función principal la autogeneración de varios documentos, es una realidad que existirá texto repetitivo en cada uno de ellos, la información mas variable (nombre de profesores, estudiantes, título de proyectos, etc.) se obtendrá de la SPODB, y el texto común o información relativamente constante (base para la creación de los documentos) se obtendrá de un archivo de datos. De igual forma, el texto de la ayuda relativa al contexto que se presentará en el Documento inteligente, se obtendrá de varios archivos de datos.

Todos los archivos de datos que se utilizarán en la SPO estarán en formato XML, y serán de dos tipos: archivos de datos XML únicamente con información de contenido (se los llamará archivos de texto sin formato), y archivos de datos XML con información de formato y contenido (se los llamará archivos de texto con formato).

3.2.2.2.1 Archivos de Texto sin Formato

Aprovechando la ventaja del soporte para los archivos en formato XML que tienen las aplicaciones de Office, se decidió utilizar un archivo de este tipo como la fuente de datos para el texto común repetitivo de cada uno de los documentos.

El diseño de este archivo es bastante simple, y a continuación se presenta una pequeña muestra de su contenido.

```
<?xml version="1.0"?>
<Parrafos>
  <Aprob_P1>
    Pongo en su conocimiento que</Aprob_P1>
  <Aprob_P1a>
    por delegación de</Aprob_P1a>
  <Aprob_P1c>
    , a su Coordinador</Aprob_P1c>
</Parrafos>
```

Ejemplo 1 Muestra del contenido XML del Archivo de Texto sin Formato.

Como se puede apreciar en el Ejemplo 1, el archivo de datos XML a ser utilizado, no necesita tener asociado ningún esquema XML específico, contiene un

elemento raíz (<Párrafos></Párrafos>), el cual a su vez contiene elementos con los párrafos o secciones de cada uno de los documentos.

La nomenclatura que se decidió utilizar para dar el nombre a las etiquetas de este archivo consiste en: las iniciales del documento al que pertenece el texto (ej. Aprob), el número del párrafo (ej. P1) y la sección del mismo (ej. d), con lo cual la etiqueta **<Aprob_P1d>, en sesión realizada el</Aprob_P1d>**, estará indicando que el texto del párrafo uno, sección d, del documento de Aprobación del Plan de Proyecto de Titulación será: “, **en sesión realizada el**”.

Acceder a esta información desde código .NET es una tarea sencilla, si se utilizan los objetos y métodos para el soporte XML integrados en la librería de clases del *Framework* .NET. Un ejemplo de código en Visual Basic .NET que permitiría acceder al contenido de la etiqueta <Aprob_P1d></Aprob_P1d> se muestra a continuación:

```
Dim doc As System.Xml.XmlDocument = New XmlDocument
Dim nodo As XmlNode
Dim s_texto As String
doc.Load("c:\Párrafos.xml")
nodo = doc.SelectSingleNode("//Aprob_P1d")
s_texto = nodo.InnerText
```

Ejemplo 2. Código .NET que permite el manejo del contenido de una etiqueta XML.

Para el Ejemplo 2 se considera que el archivo se denomina Párrafos.xml y que estará en el directorio raíz ‘ c:\ ’, la estructura jerárquica bastante simple del archivo de datos XML que se utilizará, permite que la dirección en XPATH (“//Aprob_P1d”) que se proporciona para seleccionar un nodo, sea bastante fácil de manejar y recordar. La dirección XPATH mostrada en este ejemplo sería equivalente a tener “Párrafos/Aprob_P1d” siguiendo la muestra mostrada en el Ejemplo 1.

Trabajar con archivos de datos XML es más práctico y funcional que trabajar con archivos de texto delimitados por comas o por tabulaciones. El contenido de cada uno de los nodos del archivo de datos XML también podrá ser editado fácilmente por los usuarios de la SPO, gracias a la ayuda que presenta el Word 2003 para la

visualización de este tipo de archivos. La forma en la que se verá el Archivo de Texto sin Formato para los usuarios de la SPO usando Word 2003, se muestra en la Ilustración 3-20.

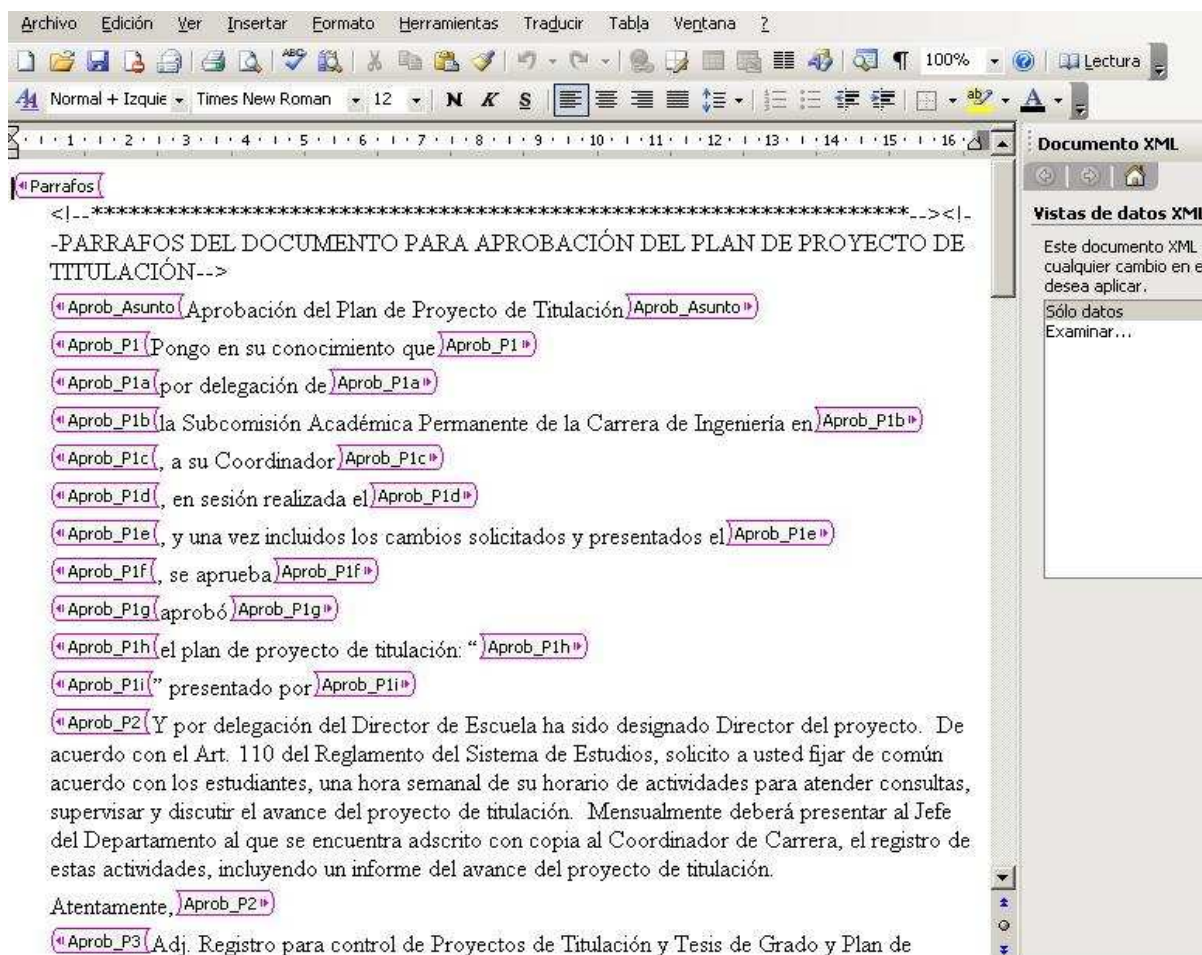


Ilustración 3-20 Archivo de datos XML que se está editando en Word 2003.

Como se puede apreciar, el contenido de cada una de las etiquetas es texto simple que el usuario podrá entender, buscar y modificar, sin tener que preocuparse de darle un formato (como tendría que hacerlo con una plantilla de Word), en vista de que en la SPO tanto la concatenación entre los segmentos de texto como el formato de los mismos se realizará mediante código y utilizando el modelo de objetos de Word.

3.2.2.2.2 Archivos de Texto con Formato

Utilizar Word 2003 permite emplear archivos de datos XML asociados al esquema *WordProcessingML*; estos archivos no sólo contendrán el texto a ser mostrado, sino también el formato que se debe aplicar sobre el mismo.

El inconveniente que se presenta al utilizar esta técnica es que para que Word pueda entender el formato que debe aplicar, se debe leer el contenido completo del archivo de datos XML (incluyendo las declaraciones de los esquemas XML), y no se puede ir seleccionando nodo por nodo como en el caso anterior (se perdería la información de formato).

Si se quisiera utilizar este tipo de archivos para insertar texto con formato directamente en los documentos que se auto generarán en lugar de dar el formato por código, habría que crear muchos archivos (uno para cada párrafo o segmento de texto), lo cual no es funcional.

Por lo tanto, estos archivos serán utilizados solamente para almacenar el texto de ayuda que aparecerá en el panel de tareas personalizado que tendrá el Documento Inteligente. A continuación se presenta una pequeña fracción de un archivo de datos XML auto generado por Word, que utiliza el esquema *WordProssesingML*.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<?mso-application progid="Word.Document"?>
<w:wordDocument xmlns:
w="http://schemas.microsoft.com/office/word/2003/wordml"
>
    <w:t>Secretaría</w:t>
    <w:rPr>
        <w:rFonts w:ascii="Arial" w:cs="Arial"/>
        <w:color w:val="000080"/>
        <w:highlight w:val="yellow"/>
    </w:rPr>
</w:wordDocument>
```

Ejemplo 3 Muestra del contenido XML de un Archivo de Texto con Formato.

Word tiene la capacidad de entender completamente la muestra del archivo que se presenta en el Ejemplo 3, pero para un desarrollador utilizar este tipo de codificación complicaría el trabajo en lugar de facilitarlo; sin embargo, como se mencionó en la Sección 2.2.1, para utilizar las ventajas ofrecidas por este tipo de archivos en la programación de aplicaciones, no es indispensable conocer la función que cumple cada uno de sus elementos, o en este caso la información

que proporciona cada una de sus etiquetas, sino más bien hay que saber como generarlos automáticamente [ver Sección **¡Error! No se encuentra el origen de la referencia.**].

Un ejemplo de código que se puede utilizar para insertar el contenido de un archivo de datos XML que utiliza el esquema *WordProssesingML* en un documento de Word, es el siguiente:

```
'El objeto Application y Document pertenecen al modelo de objetos
'de Word, son necesarios para la utilización del objeto Selection
'que determina el punto de inserción del texto, que en este caso
'será XML.
'La clase XmlDocument es propia de la tecnología .NET
'-----
Dim app As New Microsoft.Office.Interop.Word.Application
Dim doc As New Microsoft.Office.Interop.Word.Document
Dim xmlDoc As New System.Xml.XmlDocument

'El objeto xmlDoc contendrá todo el contenido del archivo "Texto
con Formato.xml", que contiene etiquetas XML bajo el esquema WordML
xmlDoc.Load("C:/Texto con Formato.xml")
app = doc.Application

'El método InsertXML es propio del modelo de objetos de Word 2003
'permite la inserción del contenido completo de un archivo XML
app.Selection.Paragraphs(1).Range.InsertXML(xmlDoc.InnerXml)
```

Ejemplo 4 Código .NET que permite insertar el contenido completo de un archivo XML en un documento de Word 2003.

En la Sección 2.2.2 se mencionó, la existencia de nuevos elementos al modelo de objetos de Word 2003 los mismos que permiten el trabajo con datos XML, y la función '*InsertXML*' que se muestra en el Ejemplo 4, es uno de estos elementos, esta función se puede utilizar con objetos '*Range*' o '*Selection*' del modelo de objetos de Word y permite insertar el contenido XML especificado, en el intervalo o la selección determinados.

Para insertar el contenido de un Archivo con Formato en el Panel de Tareas del Documento Inteligente, que es en realidad donde se utilizarán este tipo de archivos, se utiliza la misma lógica del Ejemplo 4, con la diferencia de que no se utilizará el modelo de objetos de Word sino una clase especializada, como se verá en la Sección 3.2.4.1.

3.2.3 ESQUEMAS XML DE LA SPO

Como se puede ver en la Ilustración 3-5 (arquitectura de la SPO), esta solución personalizada requiere de dos esquemas XML, uno para el Documento Inteligente, y otro para el Formulario de InfoPath. En esta sección se presentarán las consideraciones de diseño que se utilizaron para estos dos esquemas.

3.2.3.1 Esquema XML para el Documento Inteligente

Lo más importante a considerar para el diseño del esquema XML del Documento Inteligente, es que absolutamente todos los párrafos de interés y en general el texto importante de cada uno de los documentos que se auto generarán, debe estar etiquetado.

Lo primero que se realizó fue analizar la estructura de cada uno de los documentos (encabezados, párrafos, secciones para la firma del remitente, etc.), con lo cual se pudo determinar las partes comunes para todos ellos.

Lógicamente, existirán etiquetas en un documento que no existirán en los demás, las cuales deberán tener atributos que indiquen que son opcionales, otras deberán ser utilizadas en orden secuencial, y también existirán algunas que deberán ser utilizadas por lo menos una vez en cada documento.

Se definieron varios tipos complejos para tener una estructura jerárquica adecuada de las etiquetas, y una enumeración que será utilizada como el tipo de dato de un atributo para el elemento raíz del esquema que permitirá identificar el tipo de documento etiquetado.

El elemento raíz del esquema recibirá el nombre 'Documento', por otro lado el espacio de nombres se denominará: **EPN.wgvc/DI/Documentos/Electronica**, el mismo que será utilizado en repetidas ocasiones al momento de programar, razón por la cual es muy importante tenerlo bien identificado.

Los elementos tendrán el calificador *qualified*, lo que significa que cuando se vaya utilizar cualquiera de ellos desde el código .NET, habrá que hacer una referencia explícita del espacio de nombres, esto permitirá detectar más fácilmente cualquier error que se pueda cometer al hacer uso de este esquema (en caso de error de programación se provocarán excepciones fáciles de interpretar).

Para los atributos se decidió utilizar el calificador *unqualified*, en razón de que una vez que esté bien definido un elemento, es menos probable cometer errores asignándole un valor a sus atributos.

A continuación en el Ejemplo 5 se presenta una muestra donde se puede observar como fueron definidos los tipos de datos complejos propios de este esquema, en este caso es el tipo 'T_Profesor' con todos sus atributos.

```

<!--Tipo de dato que representa un profesor con los atributos que lo definen-->
<xsd:complexType name="T_PROFESOR">
  <xsd:all><!--mínimo 1 máximo 1-->
    <xsd:element name="Profesor">
      <xsd:complexType>
        <xsd:simpleContent>
          <xsd:extension base="xsd:string">
            <xsd:attribute name="codigo" use="required" form="unqualified"/>
            <xsd:attribute name="dedicacion" use="optional" form="unqualified"/>
            <xsd:attribute name="fecnac" use="optional" form="unqualified"/>
            <xsd:attribute name="cedula" use="optional" form="unqualified"/>
            <xsd:attribute name="titacad" use="optional" form="unqualified"/>
            <xsd:attribute name="direccion" use="optional" form="unqualified"/>
            <xsd:attribute name="email" use="optional" form="unqualified"/>
            <xsd:attribute name="telefono1" use="optional" form="unqualified"/>
            <xsd:attribute name="telefono2" use="optional" form="unqualified"/>
            <xsd:attribute name="nombre" use="optional" form="unqualified"/>
            <xsd:attribute name="path_foto" use="optional" form="unqualified"/>
          </xsd:extension>
        </xsd:simpleContent>
      </xsd:complexType>
    </xsd:element>
  </xsd:all>
</xsd:complexType>

```

Ejemplo 5 Muestra del XML utilizado para definir el Esquema para el Documento Inteligente.

Al visualizar el esquema XML diseñado para el Documento Inteligente en el visor XML del Visual Studio .NET, luego de que se ha comprobado que esté bien formado (sin errores lógicos ni gramaticales), se pueden apreciar gráficamente los tipos complejos, elementos, y atributos definidos tal como se muestran a continuación [Ilustración 3-21 y Ilustración 3-22].

Elemento	Nombre	Nombre	Tipo
T_PROFESOR	E	Profesor	(Profesor)
	A	codigo	string
	A	dedicacion	string
	A	fecnac	string
	A	cedula	string
	A	titacad	string
	A	direccion	string
	A	email	string
	A	telefono1	string
	A	telefono2	string
	A	nombre	string
	A	path_foto	string
	A	rol_proyecto	integer
	T_ESTUDIANTE	E	Estudiante
A		codigo	string
A		sexo	string
A		fecnac	string
A		cedula	string
A		lmilitar	string
A		lugarTrabajo	string
A		estciv	string
A		direccion	string
A		email	string
A		telefono1	string
A		telefono2	string
A		nombre	string
A		path_foto	string
T_CARRERA	E	Carrera	(Carrera)
	A	nombre	string
	A	nombre_coord	string
	A	nombre_direct	string
	A	nombre_secre	string
	A	nombre_secre	string
	A	num_telefono	string
	A	path_foto	string
	A	pag_web_2	string
	A	num_telefono	string
	A	pag_web	string

Ilustración 3-21 Elementos complejos para el Esquema del Documento Inteligente (1/2).

En la Ilustración 3-21 se muestran los elementos complejos elementales (no utilizan otros elementos complejos) que se definieron para este esquema, los mismos que como se puede apreciar tienen mucha relación con tres tablas de la SPODB que llevan su mismo nombre.

Elemento	Nombre	Nombre	Tipo
T_PARRAFO_ESPECIAL	E	UnaFirma	T_PROFESOR
	E	VariasFirmas	T_TRIBUNAL
	A		
T_SECCIONES_DOCUMENTO	E	Reporte	(Reporte)
	E	Campus	string
	E	Carrera	(Carrera)
	E	TituloDocumento	string
	E	Etiqueta	T_ETIQUETA
	E	Parrafo1	string
	E	Estudiantes	T_ESTUDIANTE
	E	Parrafo2	string
	E	Parrafo3	string
	E	Firmas	T_PARRAFO_ESPECIAL
	E	Secretaria	string
A	tipo	D_TYPO	
T_ETIQUETA	E	Etiqueta1	T_ETQ_DATOS_EPN
	E	Etiqueta2	T_ETQ_DATOS_PROYECT
	A		
T_ETQ_DATOS_EPN	E	Memorando	string
	E	Para	T_TRIBUNAL
	E	De	T_PROFESOR
	E	Asunto	string
	E	Fecha	string
T_ETQ_DATOS_PROYECTO	E	Tema	string
	E	Director	T_PROFESOR
	E	Fecha	string
	E	Nom_Estudiante	T_ESTUDIANTE
T_TRIBUNAL	E	Director	T_PROFESOR
	E	MiembroTribunal1	T_PROFESOR
	E	MiembroTribunal2	T_PROFESOR
	E	Presidente	T_PROFESOR
	A		

Ilustración 3-22 Elementos complejos para el Esquema del Documento Inteligente (2/2).

En la Ilustración 3-22 se muestran los tipos de datos complejos que contienen elementos cuyos tipos de datos corresponden a otros tipos de datos complejos definidos para este esquema; el elemento complejo principal es 'T_SECCIONES_DOCUMENTO', el cual tiene un único atributo que permitirá identificar el tipo de documento que está etiquetando y para el cual se ha definido la siguiente enumeración:

□ S _T D_TYPO		string
F	enumeration	D_TYPO_APROBACION
F	enumeration	D_TYPO_DESIGNACION_TRIBUNAL
F	enumeration	D_TYPO_DECLARACION_APTITUD
F	enumeration	D_TYPO_DEFENSA_ORAL
F	enumeration	D_TYPO_FORMULARIO_GRADO_ORAL
F	enumeration	D_TYPO_PROV_ACTA
F	enumeration	D_TYPO_PROV_DEFENSA
F	enumeration	D_TYPE_PROV_CALIFICACIONES
F	enumeration	D_TYPE_CONFIGURACION

Ilustración 3-23 Enumeración definida para el Esquema del Documento Inteligente.

La jerarquía de los elementos definida por este esquema, se obtiene al seguir la secuencia de utilización de los tipos complejos; por ejemplo, una de las posibles jerarquías de un elemento profesor es la siguiente: Documento → Etiqueta → Etiqueta1 → Para → Director → Profesor, la misma que se puede comprobar fácilmente en la Ilustración 3-22 buscando los elementos mencionados en los tipos de datos complejos según el siguiente orden: T_SECCIONES_DOCUMENTO → T_ETIQUETA → T_ETQ_DATOS_EPN → T_TRIBUNAL → T_PROFESOR.

Para tener una guía permanente en el momento de escribir código, se elaboró un árbol jerárquico, el mismo que permite determinar más rápidamente el camino o dirección XPATH para llegar a un determinado elemento, fundamental para asignar u obtener su valor. Este árbol jerárquico se muestra en la Ilustración 3-24.

De igual manera, al utilizar este esquema en Word 2003, se pueden apreciar visualmente algunas de las validaciones que se aplicaron a cada uno de los elementos, esto se presenta en la Ilustración 3-25.

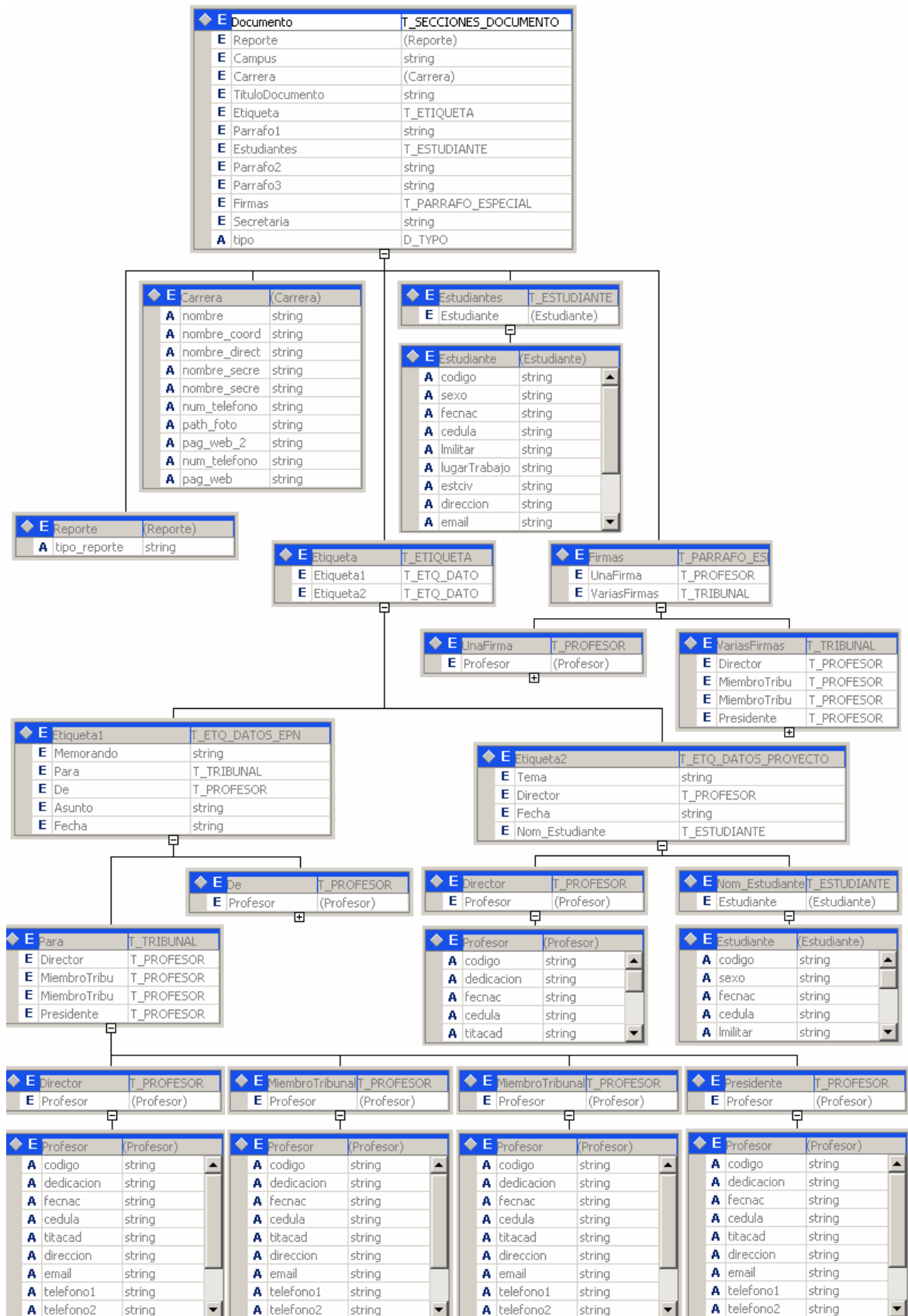


Ilustración 3-24 Árbol jerárquico del Esquema XML para el Documento Inteligente.

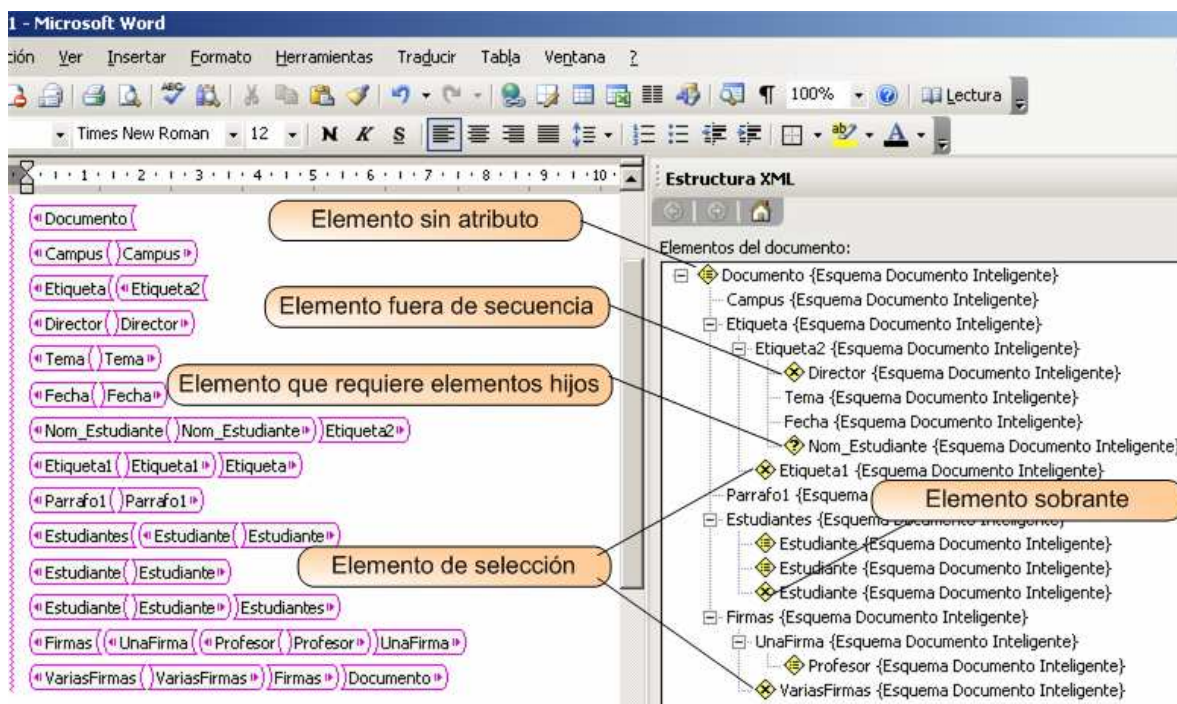


Ilustración 3-25 Ayudas de error que presenta Word cuando se etiqueta mal un documento.

En la Ilustración 3-25 se observa un documento de Word etiquetado incumpliendo varias de las reglas impuestas por el esquema diseñado para el Documento Inteligente:



- **Elemento sin atributo:** Tanto el elemento raíz '*Documento*' como los elementos 'Profesor' y 'Estudiante', contienen atributos no opcionales; en el caso del elemento raíz es requerido el atributo 'tipo', y en el caso de los otros dos el atributo 'código', estos siempre deben tener un valor.



- **Elemento fuera de secuencia:** Los elementos 'Etiqueta1' y 'Etiqueta2', deben tener todos sus elementos hijos en secuencia, el error del ejemplo se da porque el elemento 'Tema' debe estar antes del elemento 'Director'.



- **Elemento que requiere elementos hijos:** Existen varios elementos tales como: 'Director', 'MiembroTribunal', 'Presidente', 'Para', 'De', 'Nom_Estudiante', que una vez que se los utiliza, también deben incluirse a todos sus elementos hijos (definidos por tipos complejos).



- **Elemento de selección:** los elementos 'Etiqueta' y 'Firmas', deben utilizar solo uno de sus elementos hijos; por ejemplo, el elemento 'Etiqueta' puede utilizar el elemento 'Etiqueta1' o el 'Etiqueta2', pero no ambos.



- **Elemento sobrante:** La gran mayoría de elementos fueron definidos con el atributo 'maxOccurs', asignado con un valor; en el caso del elemento 'Estudiante', este valor es dos, por esta razón al haber tres elementos 'Estudiante' se produce un error.

En general, las ayudas que presenta Word cuando se utiliza mal algún esquema no son lo suficientemente explícitas, y como se puede ver se utilizan los mismos íconos para representar errores diferentes, de todas formas son una guía para saber que se ha cometido algún error.

3.2.3.2 Esquema XML para el Formulario de InfoPath

Para el caso de InfoPath, el diseño del esquema XML es un tanto diferente, un ejemplo de la forma en la que se definen los tipos en este esquema se muestra a continuación:

```
<xsd:complexType>
  <xsd:sequence>
    <xsd:element ref="my:PROYECTOS" />
    <xsd:element ref="my:fechaSesion" />
  </xsd:sequence>
</xsd:complexType>
```

Ejemplo 6 Ejemplo de definición de Tipos Complejos para el Esquema XML de InfoPath.

Si el mismo elemento del Ejemplo 6 estuviera definido en el esquema para el Documento Inteligente, se vería de la siguiente manera:

```
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="PROYECTOS" type="PROYECTOS"/>
    <xsd:element name="fechaSesion" type="xsd:date"/>
  </xsd:sequence>
</xsd:complexType>
```

Ejemplo 7 Definición del Tipo Complejo del Ejemplo 6 en el formato del Esquema XML diseñado para el Documento Inteligente.

Para cualquiera de los dos ejemplos, 'PROYECTOS' es otro tipo complejo definido en el esquema, la diferencia radica en que para el esquema de InfoPath la mayoría de elementos definidos no tienen especificado un tipo de datos sino que son referidos; es decir, que un elemento complejo hace referencia a otro o a otros más elementales (más bajos en jerarquía), lo cual es muy útil para el trabajo con múltiples esquemas.

La forma predeterminada de trabajar con datos externos que utiliza InfoPath, es mediante la utilización de *Datasets* con tipo⁶⁷ o Conjuntos de Datos con Tipo de ADO .NET, esto implica que existirá un esquema XML para cada origen de datos diferente. En el caso de la SPO el origen de datos para el Formulario de InfoPath será la SPODB, por lo tanto existirá un Conjunto de Datos con Tipo para la SPODB.

Para evitar confusiones entre los elementos del esquema del formulario y los elementos del esquema para el Conjunto de Datos de la SPODB, se debe utilizar el nombre de cada uno de los elementos acompañado de un prefijo, y esto es justamente lo que se logra definiendo los tipos de datos complejos como referidos.

Para el caso del Ejemplo 6, el acceso al tipo de datos complejo 'PROYECTOS', debe hacerse utilizando 'my:PROYECTOS'; las cadenas de XPATH que se utilicen en el código del formulario de InfoPath para seleccionar un elemento o un grupo de elementos siempre deben incluir los prefijos.

El código en Visual Basic .NET que se debe utilizar para obtener el valor contenido en el elemento 'fechaSesion' (también definido en el Ejemplo 6), se muestra en el Ejemplo 8, para el cual se ha considerado que la clase en donde estará el código especificado incluye una referencia al ensamblado de interoperabilidad para InfoPath Microsoft.Office.Interop.InfoPath.SemiTrust.

⁶⁷ Un *Dataset* de ADO.NET es una representación de un conjunto de datos (tablas, campos, etc.) mediante XML; se dice que un *Dataset* tiene tipo si toda su información se rige a las condiciones de un esquema XML.

```

Dim XDocument As XDocument
Dim nd_fecha As IXMLDOMNode
Dim fecha_Sesion As Date
nd_fecha =thisXDocument.DOM. _
selectSingleNode("my:SESION/my:fechaSesion")
fecha_Sesion = nd_fecha.nodeValue

```

Ejemplo 8 Código para tener acceso al valor un elemento definido para el Esquema de InfoPath.

La cadena XPATH ("my:SESION/my:fechaSesion"), utilizada en el Ejemplo 8, saca a relucir el elemento 'SESION', que justamente será el elemento raíz definido para el esquema XML del Formulario de InfoPath, se escogió este nombre en vista de que este formulario estará diseñado para recopilar los apuntes relevantes a los Proyectos de Titulación durante una sesión de Subcomisión Académica.

InfoPath ofrece bastante ayuda para el trabajo con XML, y en este caso cuando se comete algún error en el diseño del esquema del Formulario de InfoPath su corrección es bastante simple e incluso se lo hace visualmente (*clic* derecho para agregar o quitar elementos o atributos), lo que no ocurre con el esquema del Documento Inteligente, en el cual incluso hay que recurrir al Registro de Windows para eliminar las entradas del registro correspondientes a dicho esquema, para que cualquier cambio en el mismo tenga efecto.

En el esquema definido para el Formulario de InfoPath existirán dos elementos: 'PROYECTOS' y 'Observaciones', con el atributo 'maxOccurs="unbounded"', lo que significa que podrán utilizarse una cantidad de veces indeterminadas dentro de un mismo formulario, esto se debe a que no es posible determinar cuantos proyectos se puedan tratar en una sesión, ni cuantas observaciones se pueden agregar a cada uno de ellos.

No se realizó ningún árbol jerárquico de ayuda para los elementos XML definidos en el esquema del Formulario de InfoPath, ya que los elementos tanto de este esquema, como los del esquema del *Dataset* con Tipo (se generan automáticamente con ayudas visuales) que se utilizará son fáciles de ubicar desde la vista de diseño de InfoPath, gracias a la ayuda visual ofrecida por el panel de tareas Origen de Datos, como se puede apreciar en la Ilustración 3-27.

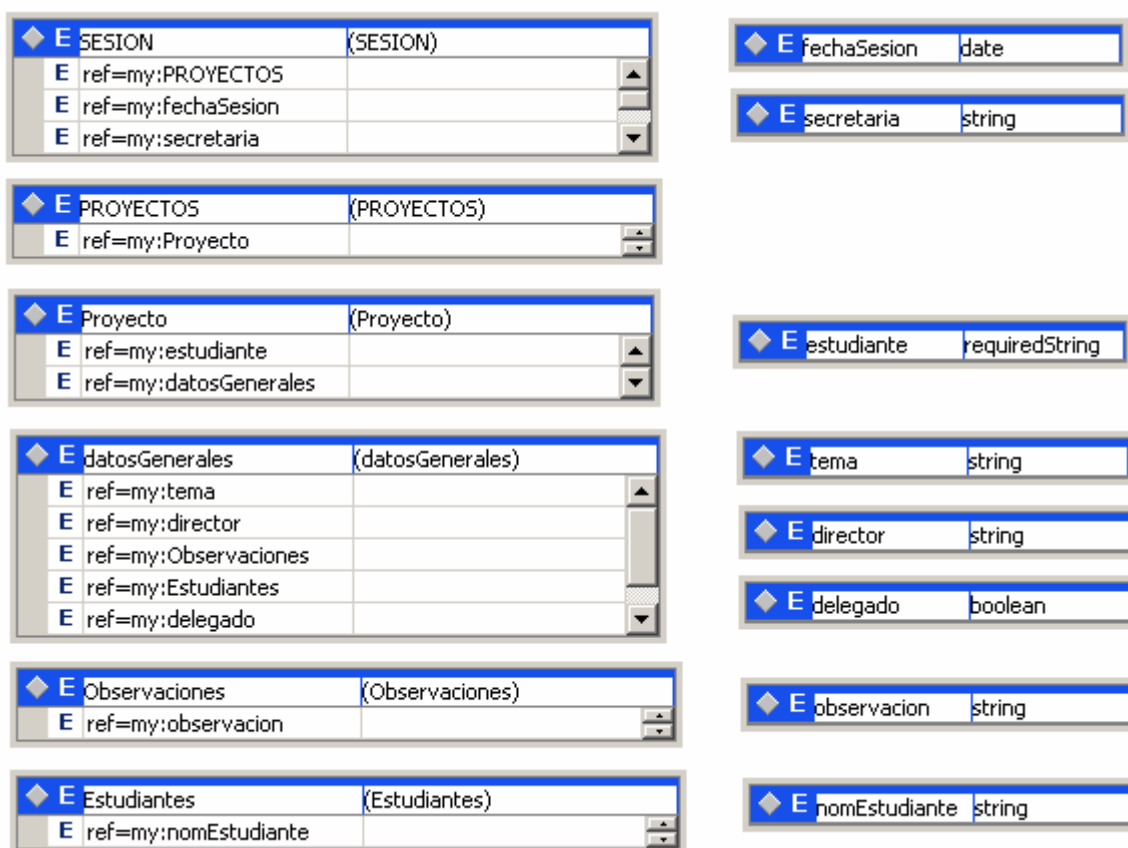


Ilustración 3-26 Diagrama del Esquema XML diseñado para el Formulario de Infopath.

Esquemas XML que se utilizarán en el Formulario de InfoPath	
<pre> SESION ├── PROYECTOS │ ├── Proyecto │ │ ├── estudiante* │ │ └── datosGenerales │ │ ├── tema │ │ ├── director │ │ └── Observaciones │ │ ├── observacion │ │ └── Estudiantes │ │ ├── nomEstudiante │ │ └── delegado │ └── fechaSesion └── secretaria </pre>	<pre> myFields ├── dataFields │ ├── d:T_Proyecto │ │ ├── :pry_cod │ │ ├── :pry_titulo │ │ ├── :pry_cod_pro_dir │ │ └── :pry_fec_pres │ ├── T_Profesor │ │ ├── :pro_codprf │ │ └── :pro_nompro │ ├── T_Proyecto_Estudiante │ │ ├── :pryestud_estud_cod │ │ └── :pryestud_pry_cod │ └── T_Estudiante │ ├── :estud_cod │ └── :estud_nombre </pre>
<p align="center">Esquema principal para el Formulario de InfoPath</p>	<p align="center">Esquema para el acceso a datos del <i>Dataset</i> con Tipo</p>

Ilustración 3-27 Esquemas XML que utilizará el Formulario de InfoPath.

3.2.4 CAPAS LÓGICAS

Para el diseño de la arquitectura de software de la SPO, se utilizó el patrón de diseño de tres capas, para lo cual se dividen los elementos de la aplicación en paquetes⁶⁸ de Interfaz de Usuario, Lógica de Negocio y Acceso a Datos y Servicios.

Cada una de las capas lógicas estarán conformadas por paquetes principales y secundarios (anidados en un paquete principal); para la determinación de estos paquetes se utilizó como Referencia el Diagrama 0 de la SPO [Ilustración 3-10 a Ilustración 3-13].

En la Ilustración 3-28 se presenta un diagrama básico de las tres capas lógicas de la SPO, en donde se puede apreciar que el Documento Inteligente, el Formulario de InfoPath y el Servicio Web, son subsistemas independientes.

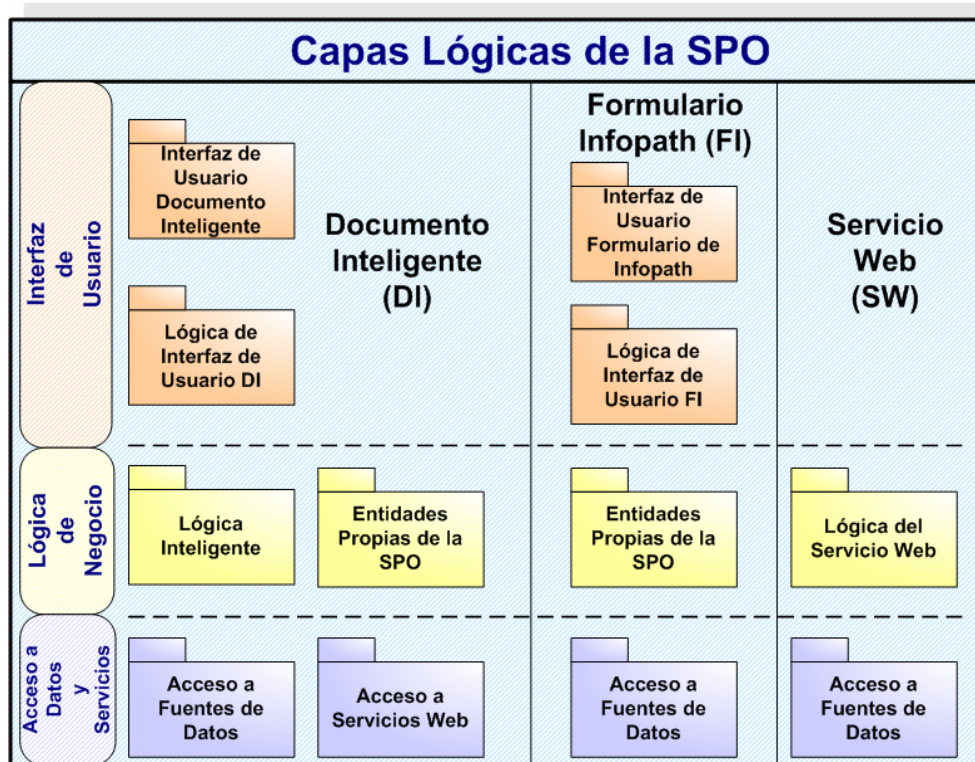


Ilustración 3-28 Capas Lógicas de la SPO.

⁶⁸ Un paquete es una agrupación de elementos; es el elemento básico de organización de un modelo de UML; un paquete puede anidarse en otro [Ref. 12].

3.2.4.1 Interfaz de Usuario

El usuario de la SPO tendrá como interfaz un Documento Inteligente (interfaz personalizada de Word 2003) y un Formulario de InfoPath; por definición el Servicio Web no tiene interfaz gráfica de usuario, lo cual se puede apreciar claramente en la Ilustración 3-28.

La Ilustración 3-29 muestra los paquetes principales y secundarios que componen la capa lógica de Interfaz de Usuario de la SPO.

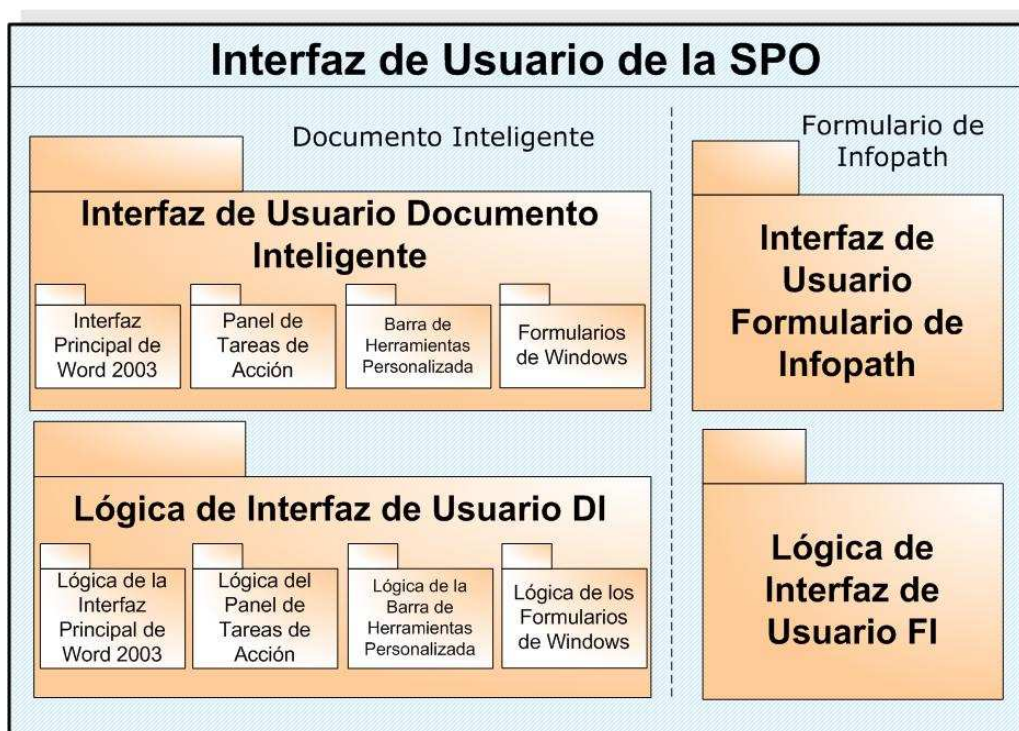


Ilustración 3-29 Capa Lógica de Interfaz de Usuario de la SPO.

3.2.4.2 Lógica de Negocio

Para el caso de la Lógica de Negocio de la SPO, se consideró la existencia de tres paquetes: Lógica Inteligente, Entidades Propias de la SPO, y Lógica del Servicio Web, de los cuales el segundo será común para dos de los tres subsistemas de la SPO: el Documento Inteligente y el Formulario de InfoPath, tal como se muestra en la Ilustración 3-30.

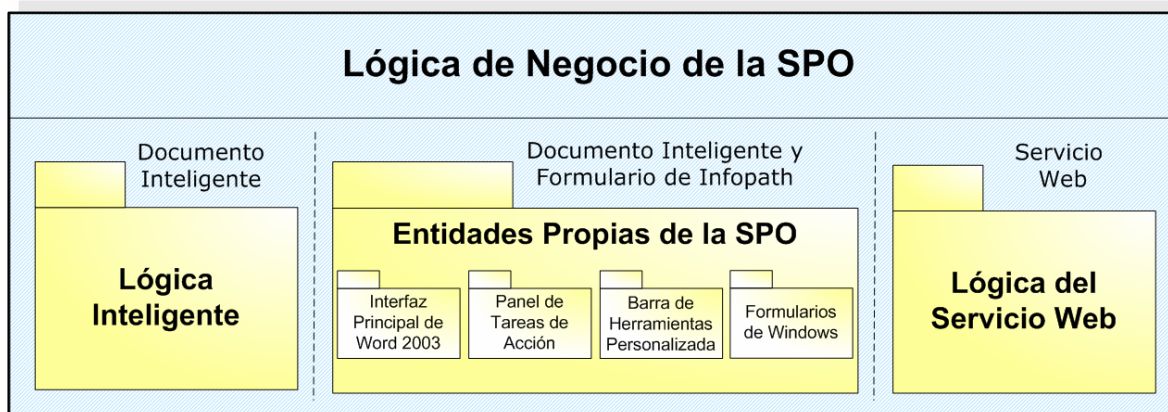


Ilustración 3-30 Capa Lógica de Negocio de la SPO.

3.2.4.3 Acceso a Datos y Servicios

Para el diseño de la capa de Acceso a Datos y Servicios, lo primero a considerar fueron las fuentes de información de la SPO, esto permitió determinar que esta capa constará de dos paquetes principales:

- Acceso a Servicios Web
- Acceso a Fuentes de Datos

El paquete de Acceso a Servicios Web será exclusivo del Documento Inteligente, mientras que el de Acceso a Fuentes de Datos será común para todos los subsistemas, tal como se muestra en la Ilustración 3-31.

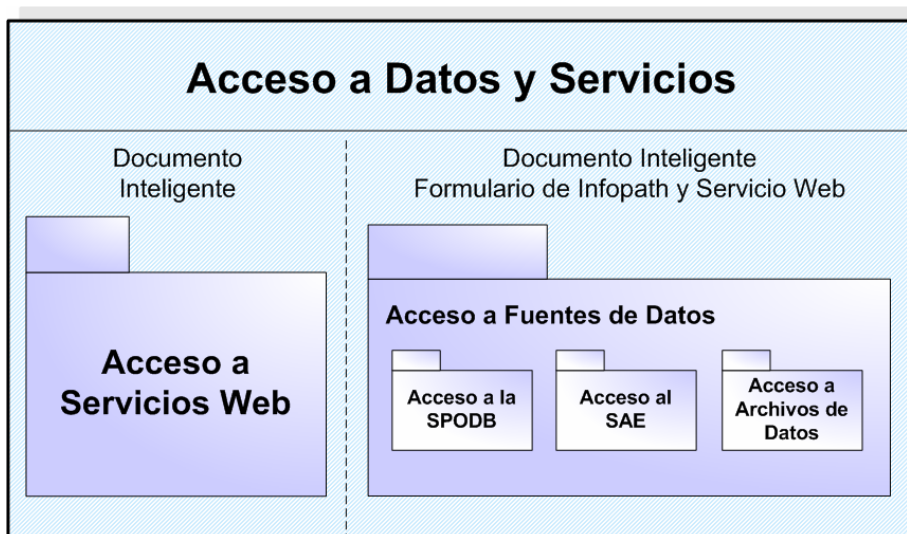


Ilustración 3-31 Capa Lógica de Acceso a Datos y Servicios.

3.3 IMPLEMENTACIÓN

En esta Sección se señalan los aspectos más importantes que fueron considerados para la implementación de cada una de las capas lógicas de la SPO, adicionalmente se especificarán los requerimientos de *software* para el equipo que alojará la solución, y varias consideraciones específicas en cuanto a la implementación del Documento Inteligente y del Formulario de Infopath.

3.3.1 IMPLEMENTACIÓN DE LAS CÁPAS LÓGICAS

Para explicar de mejor manera la lógica subyacente asociada con la implementación de la SPO, se utilizarán gráficos que muestran cada una de las clases y métodos que definen las capas lógicas [Ilustración 3-28], por cuestiones de espacio no se presentarán los parámetros de entrada ni los valores de retorno de los métodos que contienen cada una de las clases, pero se presentarán algunos ejemplos de implementación.

De igual manera se utilizarán los diagramas de flujos de datos que fueron utilizados para el diseño de la SPODB; en este caso, lo más importante para la implementación son los procesos incluidos en el Diagrama 0, razón por la cual se presentarán varios gráficos que relacionan los procesos con los paquetes que componen cada una de las capas lógicas de la SPO.

3.3.1.1 Interfaz de Usuario

En el Diagrama 0 todos aquellos procesos que están en relación directa tanto con las entradas como con las salidas de la SPO, exceptuando aquellas relacionadas con las entidades: “Sistema de Recopilación de Información Universitaria” y “SPO Remota”, deben tener un equivalente en la capa de la interfaz de usuario.

En la Ilustración 3-32 se presentan los procesos que tienen relación con los paquetes definidos para la Interfaz de Usuario de la SPO.

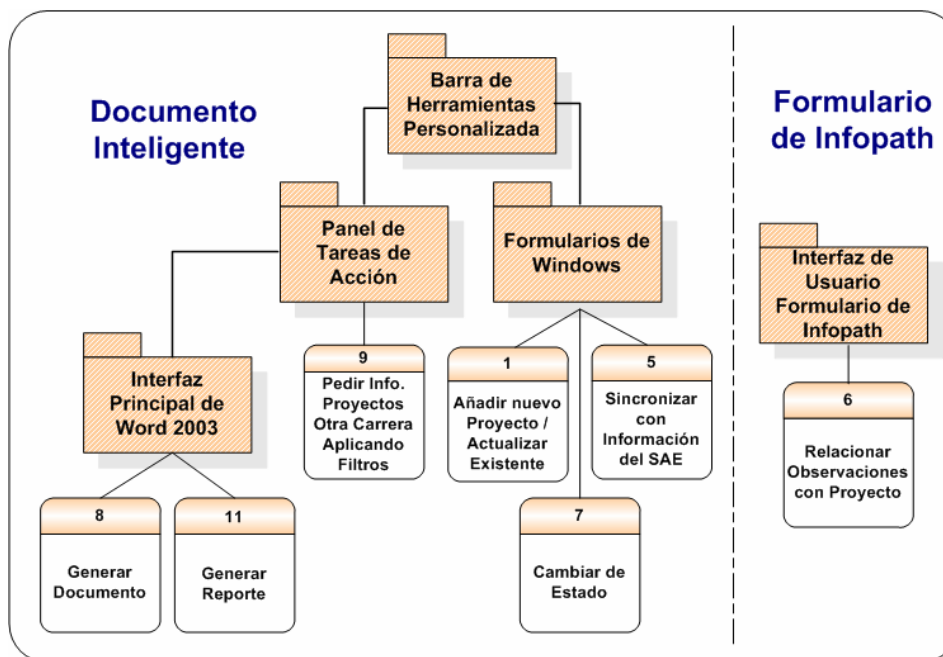


Ilustración 3-32 Relación de Procesos con la capa lógica de Interfaz de Usuario.

3.3.1.1.1 Interfaz de Usuario del Documento Inteligente

Para el caso del Documento Inteligente, como se mencionó en el Capítulo 1, la gran ventaja es que no hay que preocuparse por desarrollar algo que ya está hecho, que en este caso en particular, será la interfaz que ofrece Word 2003.

Sin embargo, hay que considerar que a pesar de estar utilizando una interfaz ya desarrollada, ésta estará personalizada de acuerdo a los requerimientos de la SPO, contendrá una Barra de Herramientas adicional, un Panel de Tareas de Acción, y permitirá al usuario acceder a Formularios de Windows (pantallas con controles) a manera de cuadros de diálogo.

En la Ilustración 3-33 se puede apreciar un esquema de las áreas principales que conforman la interfaz de Word 2003, este esquema es importante en vista de que sirve de guía para identificar de mejor manera los elementos que serán personalizados y la ubicación que tendrán dentro de la interfaz de usuario.

En el área de Barra de Herramientas se ubicará la Barra de Herramientas Personalizada, el Panel de Tareas de Acción aparecerá en el área

correspondiente y el área de la Ventana principal de Word es la hoja en blanco en donde se encontrarán los documentos (cartas, memorandos y reportes) que generará el Documento Inteligente.

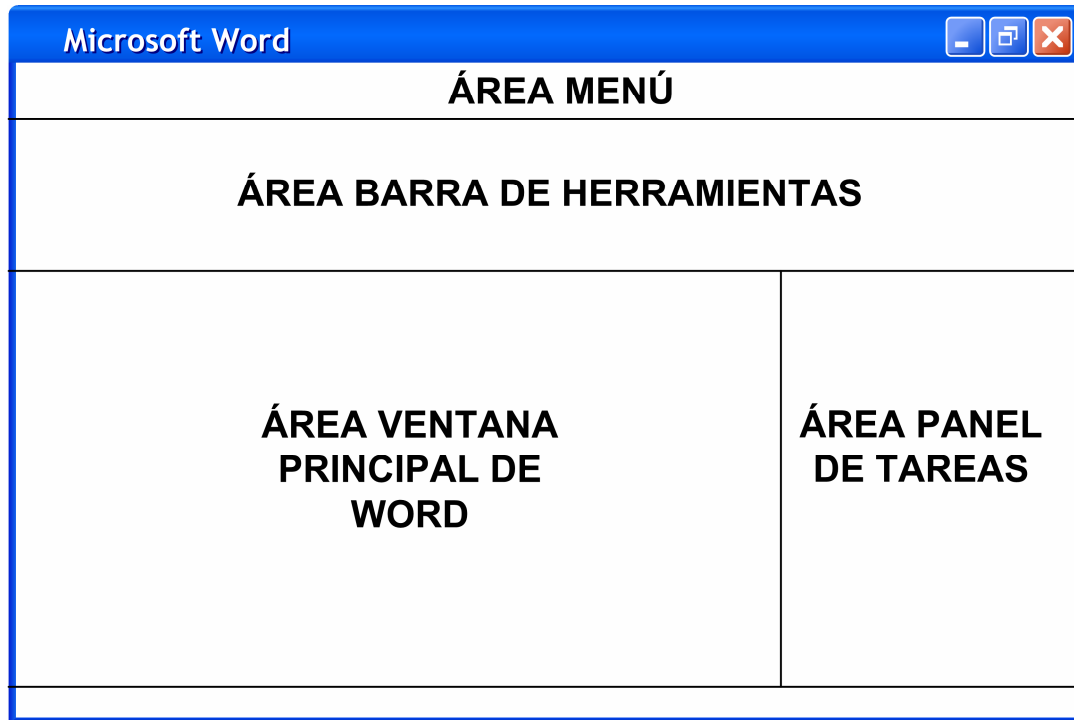


Ilustración 3-33 Interfaz de Usuario de Microsoft Word 2003.

Panel de Tareas de Acción: En el panel de tareas se manejará la interfaz de usuario relacionada con la sensibilidad al contexto de los documentos, y también será donde el usuario configure o seleccione los documentos que desea generar.

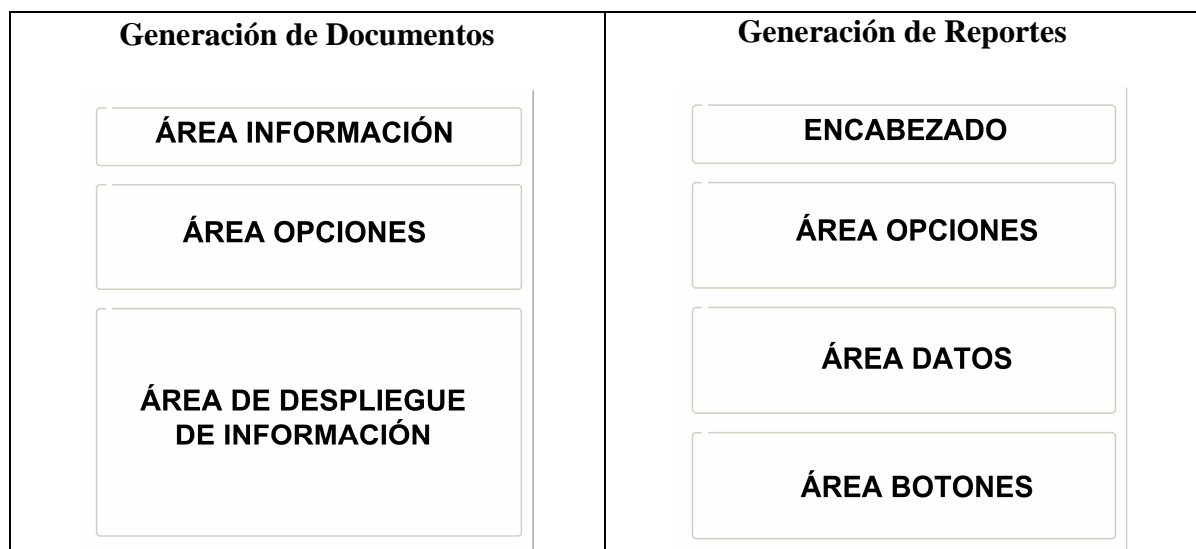


Ilustración 3-34 Interfaces del Panel de Tareas de Acción.

Para el caso de la sensibilidad al contexto no se consideró necesario realizar el diagrama de interfaz ya que únicamente existirá un área de Despliegue de Información (dependerá de la ubicación del usuario dentro del documento).

Por otro lado, para la generación de documentos es importante establecer una diferenciación entre lo que es generar una carta o memorando y lo que es generar un reporte, para ello se realizaron los diagramas de interfaz de usuario mostrados en la Ilustración 3-34 donde se pueden apreciar estas dos configuraciones.

La interfaz mostrada en el lado izquierdo de la Ilustración 3-34 será utilizada cuando el usuario solicite generar un documento, mientras que la interfaz de la derecha será utilizada cuando el usuario solicite generar un reporte.

La descripción de cada una de las áreas que conforman la interfaz de usuario para el Panel de Tareas, se muestra a continuación (Tabla 3-22).

NOMBRE DE ÁREA	DESCRIPCIÓN
Área Información	Lugar donde se va a colocar información de ayuda general para el usuario.
Área Opciones	Lugar donde se van a colocar controles de opciones para seleccionar el tipo de documento que será generado y mostrado en la ventana principal de Word.
Área de Despliegue de Información	Similar al área de Información, con la diferencia de que los datos mostrados en esta sección serán con relación a un proyecto específico.
Encabezado	Sección muy pequeña de información utilizada para colocar un título simple de guía para el usuario.
Área Datos	Es el lugar donde se colocarán los controles que permitirán establecer los filtros de información para la generación de reportes.
Área Botones	Lugar donde se ubicará el control de comando para ejecutar la generación de un reporte.

Tabla 3-22 Descripción de áreas para la interfaz de usuario del panel de tareas.

Formularios de Windows: Para todos aquellos procesos que requieren el ingreso de varios datos; tales como, añadir o actualizar un proyecto, cambiar el estado de un proyecto, etc., se decidió utilizar Formularios Windows.

Los Formularios aparecerán como cuadros de diálogo modales y deberán tener entre sí el mismo aspecto y tamaño; el diseño de la interfaz de usuario para cada formulario fue realizado teniendo en consideración las funciones que van a desempeñar en relación con los casos de uso mencionados en las Secciones 3.1.3.4 y 3.1.3.4.1.

A continuación se presentarán una serie de tablas que muestran los diagramas de interfaz de cada uno de los Formularios Windows que contendrá el Documento Inteligente.

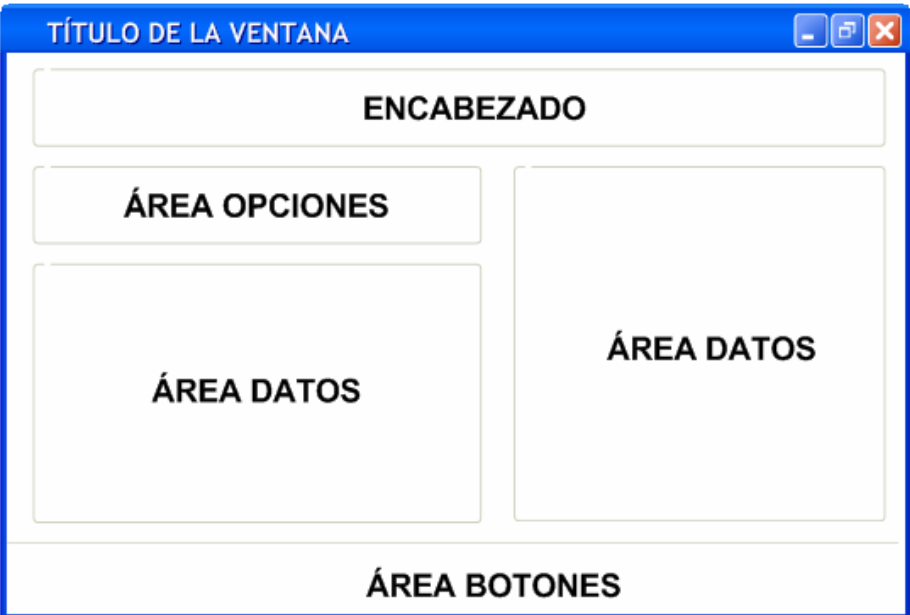
INTERFAZ 0	Ingreso y Actualización de Proyectos de Titulación
DIAGRAMA	
	
DESCRIPCIÓN	Los casos de uso asociados con esta interfaz son: “El ingreso de nuevos PPTs” y “Actualización de la información de los PPTs y PTs”; los dos casos de uso requieren los mismos datos y para establecer una distinción entre el ingreso y la actualización se ha colocado el Área de Opciones.

Tabla 3-23 Interfaz de usuario para el Ingreso y la Actualización de Proyectos.

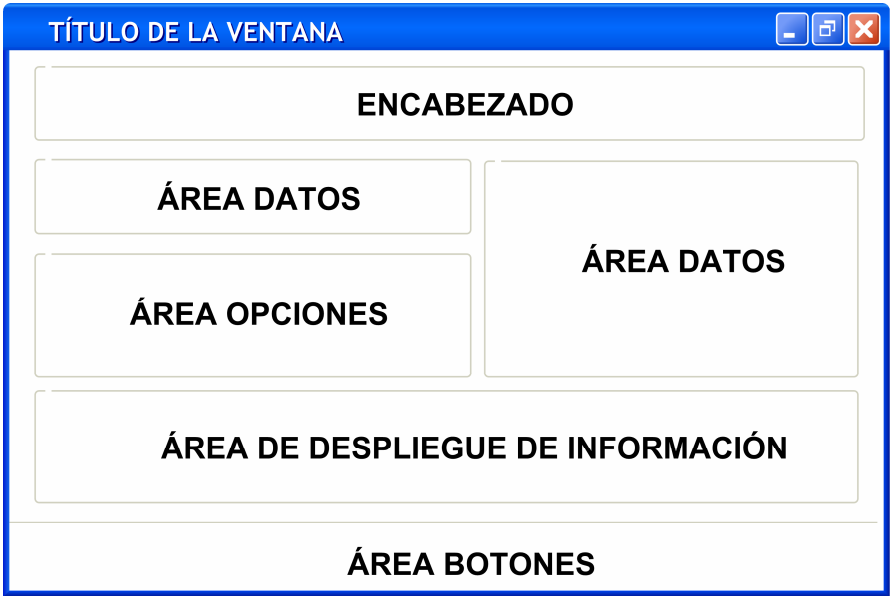
INTERFAZ 1	Cambio de Estados de Proyectos de Titulación
DIAGRAMA	
	
DESCRIPCIÓN	Los casos de uso asociados con esta interfaz son la identificación y la modificación del estado actual de un PPT o PT, el área opciones será para los estados posibles.

Tabla 3-24 Interfaz de usuario para el Cambio de Estados de un Proyecto.

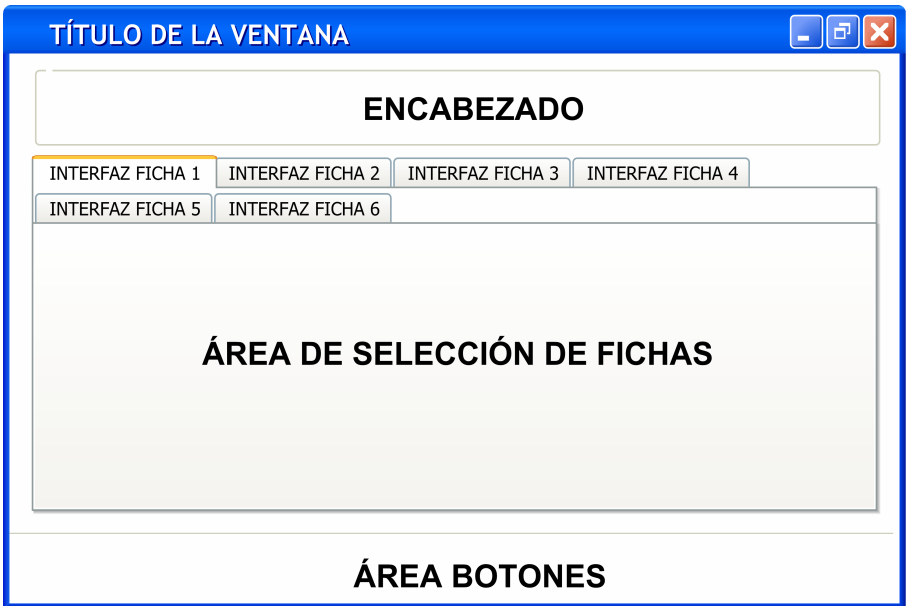
INTERFAZ 2	Configuración del Sistema
DIAGRAMA	
	
DESCRIPCIÓN	Formulario con interfaz de varias fichas; servirá para lo relativo a Sincronización con información del SAE, Edición del Texto común repetitivo y Configuración de Direcciones y contraseñas

Tabla 3-25 Interfaz de usuario para la Configuración del Sistema.

La interfaz de Configuración del Sistema (Tabla 3-25) deberá ser la menos utilizada por el usuario; es decir, que la idea fundamental es que esta pantalla contenga todas aquellas funciones que no son de uso cotidiano pero que en determinado momento deberán ser realizadas; los tres casos de uso con relación de extensión (Sección 3.1.3.4) no son de uso frecuente, por lo que su ubicación dentro de esta interfaz es la más apropiada.

Las funciones que se ejecutarán desde esta pantalla serán las siguientes:

- Sincronización con información del SAE
- Configuración de direcciones
- Configuración de permisos de edición del contexto de los documentos
- Edición del texto común repetitivo
- Cambio de nombre de usuario y contraseñas
- Edición de los datos de la Carrera

Si bien es cierto que tanto la edición de los datos de la Carrera, como el cambio de nombre de usuario y contraseña no aparecen explícitamente en el diagrama de casos de uso (Ilustración 3-2), hay que hacer notar que los datos de la Carrera son parte del texto común repetitivo de los documentos, y el cambio de nombre de usuario y contraseña es necesario para establecer permisos de edición en el contexto de los documentos.

La razón por la cual la edición de los datos de la Carrera se va a hacer en forma diferente que el resto del texto que aparece en cada uno de los documentos, es que la información de la Carrera se encontrará almacenada en la Base de Datos, mientras que el resto del texto se encontrará almacenado en Archivos de Texto sin formato (ver Sección 3.2.2).

Es importante mencionar que se decidió utilizar un formulario con interfaz de fichas ya que se consideró es práctico y funcional, esto permite simular la existencia de varias interfaces distintas (dependiendo de la opción seleccionada) bajo un mismo espacio (dentro del formulario). Las interfaces a ser utilizadas en cada una de las fichas se muestran en las Tablas 3-26 a 3-30.

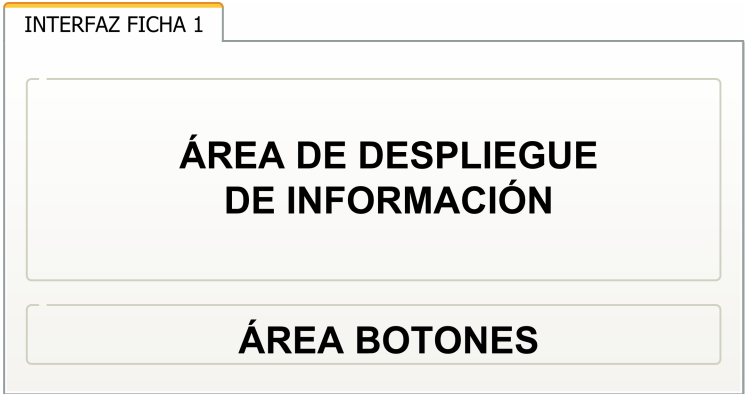
INTERFAZ 2-1	Sincronización con información del SAE
DIAGRAMA	
	
DESCRIPCIÓN	<p>La sincronización de información es un proceso fácil para el usuario y complejo para el sistema, el área de botones contendrá un solo botón de sincronización.</p> <p>El área de despliegue de información mostrará como resultado la cantidad de datos leídos del SAE y la cantidad de datos copiado a la SPODB.</p>

Tabla 3-26 Interfaz para la Sincronización con información del SAE.

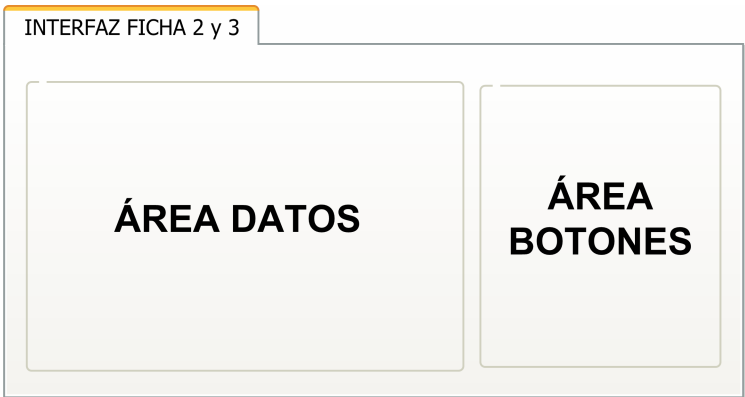
INTERFAZ 2-2	Configuración de Direcciones y Permisos de Edición de Texto
DIAGRAMA	
	
DESCRIPCIÓN	<p>Tanto la ficha para la configuración de direcciones como la ficha para configuración de permisos de edición de texto, tendrán un área de datos y una de botones.</p> <p>En el caso de las direcciones, en los datos se ingresarán las direcciones de los servidores y en el área botones estarán los controles para probar la conexión.</p> <p>En el caso de la configuración de edición del texto, en los datos se señalan los permisos para el bloqueo de los documentos, y en el área botones se encuentran los controles para poder aplicar o quitar dichos permisos.</p>

Tabla 3-27 Interfaz de Direcciones y Permisos de Edición de Texto.

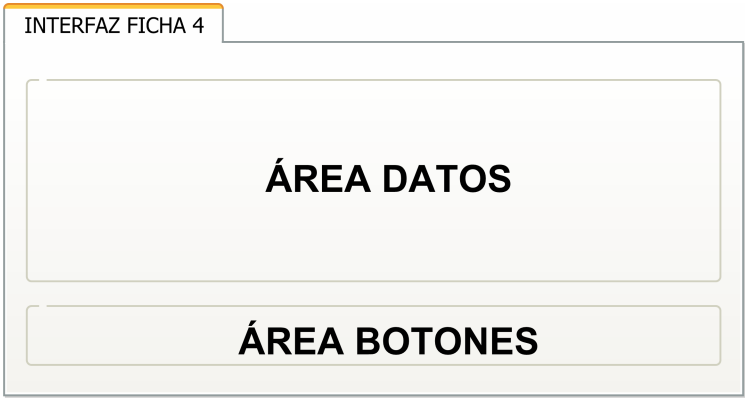
INTERFAZ 2-3	Cambio de nombre de Usuario y Contraseña
DIAGRAMA	
	
DESCRIPCIÓN	<p>Para el cambio de nombre de usuario y contraseña, en el área de datos se ubicarán controles para ingresar el nombre de usuario, la contraseña actual y la nueva contraseña.</p> <p>Se utilizará un sistema de validación instantánea que permitirá comprobar la información en el momento en el que es ingresada y en el área de botones se encontrará el comando para realizar el cambio de información.</p>

Tabla 3-28 Interfaz para Cambio de nombre de usuario y contraseñas.

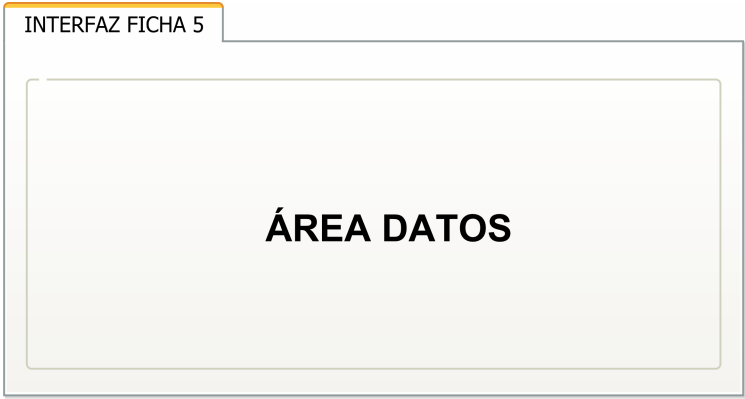
INTERFAZ 2-4	Edición de los datos de la Carrera
DIAGRAMA	
	
DESCRIPCIÓN	<p>En la ficha destinada a la edición de los datos de la Carrera, se utilizará únicamente el área de datos, en donde aparecerá toda la información almacenada en la Base de Datos y podrá ser modificada directamente, no se requieren botones de comandos de guardar y cancelar porque los cambios son almacenados automáticamente.</p>

Tabla 3-29 Interfaz de usuario para la Edición de los datos de la Carrera.

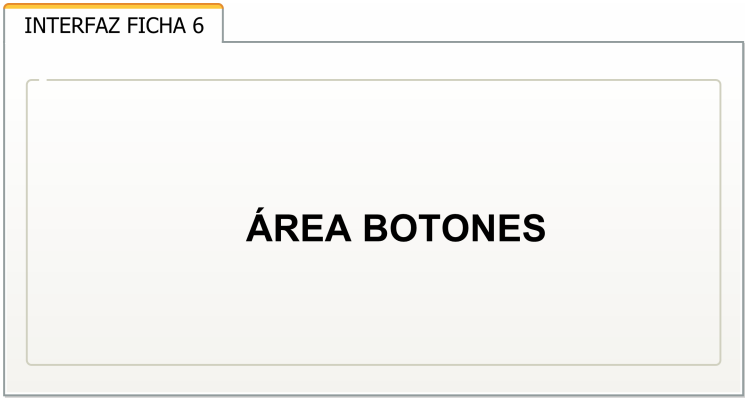
INTERFAZ 2-5	Edición del Texto Común Repetitivo
DIAGRAMA	
	
DESCRIPCIÓN	Para el proceso de edición del Texto Común de los documentos, se utilizará el Microsoft Word para abrir el archivo que contiene el texto; sin embargo, en la ficha del formulario de configuración destinada para este propósito se encontrarán los botones de comandos que permiten abrir el archivo directamente desde la aplicación.

Tabla 3-30 Interfaz de usuario para el texto común repetitivo.

La descripción de cada una de las áreas que conforman la interfaz de usuario para los Formularios Windows, se presenta a continuación (Tabla 3-31).

Encabezado:	Título que identifica la función principal para la que será utilizado el formulario.
Área Opciones:	Lugar donde se van a colocar controles de opciones que permiten variar el tipo objetos que se muestra en el área datos y en el área botones.
Área Datos:	Es el lugar donde se van a colocar diversos objetos que permitan manipular datos, cajas de texto, combos, etc.
Área Botones:	Son las opciones de comandos que pueden ser utilizadas para ejecutar diferentes tipos de acciones en función de los datos de la pantalla.
Área Despliegue de Información:	Espacio que mostrará texto informativo sobre los resultados obtenidos al ejecutar alguna de las acciones del área de botones.

Tabla 3-31 Descripción de las áreas que comprenden la interfaz de usuario de los Formularios.

Interfaz Principal de Word: Algo muy importante es que la ventana principal de Word (sección para la escritura del documento) también debe tener un diseño específico similar al de una interfaz de usuario.

Para entender mejor el diseño de la ventana principal de Word es necesario presentar como ejemplo uno de los documentos que serán generados automáticamente por el sistema, en donde se podrán identificar las distintas secciones a las que el usuario ingresará de una manera completamente transparente, es decir que pasarán desapercibidas para él, pero que internamente causarán la ejecución de una acción predeterminada.

El documento que se muestra a continuación Ilustración 3-35 corresponde a la aprobación de un Plan de Proyecto de titulación.

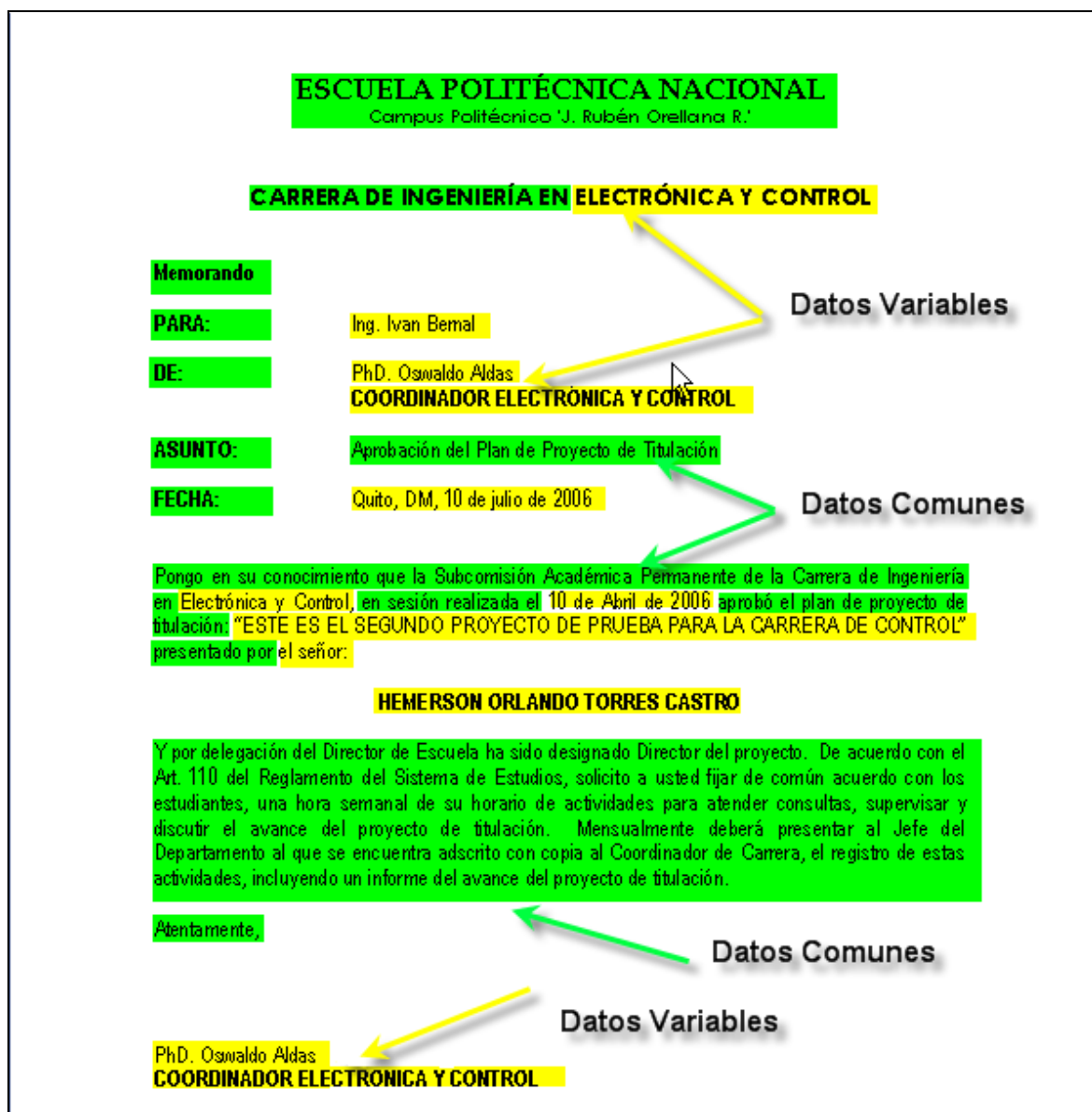


Ilustración 3-35 División en secciones de un documento a ser auto-generado.

En la Ilustración 3-35 se pueden apreciar dos tipos de Datos: Comunes y Variables, para hacer una analogía con las diferentes áreas de una interfaz de Formulario Windows, se puede decir que los Datos Comunes corresponden a un Área de Despliegue de Información, mientras que los Datos Variables corresponden a un Área de Datos o a un Área de Botones, dependiendo de que se ejecutó o no una acción en el Panel de Tareas de Acción.

En realidad del correcto diseño de la “interfaz” de la ventana principal de Word dependerá el buen funcionamiento de la lógica inteligente; por otro lado, se puede considerar que la “interfaz” de la ventana principal de Word es dinámica en vista de que depende de cada una de las diferentes secciones que contengan los documentos a ser generados.

Si bien es cierto que en esta Sección se presenta tan sólo un ejemplo de cómo se divide uno de los documentos en segmentos de datos (variables o constantes), es importante hacer énfasis en que esto se realizó para todos, y que del análisis de los segmentos se deriva el diseño del esquema XML (3.2.3.1), y la forma en la que son colocadas las etiquetas XML en cada uno de los documentos.

3.3.1.1.2 Lógica de la Interfaz de Usuario del Documento Inteligente

Como se señaló en la Sección 3.3.1.1.1 en el Documento Inteligente el usuario deberá interactuar con cuatro elementos de interfaz de usuario: La Barra de Herramientas Personalizada, el Panel de Tareas de Acción, los Formularios Windows y el Documento o Ventana Principal de Word.

Si bien es cierto que al tener los diagramas de interfaz de usuario, y el diseño de la Arquitectura de la Solución (Sección 3.2.1) se sabe la manera en la que funcionará el sistema y se tiene una idea de cómo el usuario interactuará con él, técnicamente el eje fundamental se encuentra en la parte lógica del sistema.

Para la lógica de la Interfaz de Usuario, en la Ilustración 3-29 se puede observar que la capa para el Documento Inteligente consta de cuatro paquetes secundarios

uno por cada elemento de la interfaz con los que el usuario podrá interactuar, y para los cuales es preciso señalar las siguientes consideraciones:

- Tanto los documentos como los reportes que generará la SPO, aparecerán en la Interfaz Principal Word.
- La Barra de Herramientas Personalizada será el primer elemento de contacto que tenga el usuario con la Interfaz del Documento Inteligente, permitirá abrir los Formularios Windows, así como también ayudará a mostrar ciertos controles en el Panel de Tareas de Acción.
- La interfaz de usuario para la parte de la “lógica inteligente”, al igual que para el ingreso de los parámetros que permitirán filtrar la información que se presente en un reporte, será el Panel de Tareas de Acción.
- Toda la comunicación que existirá entre los elementos que conforman los paquetes secundarios de la Interfaz de Usuario del Documento Inteligente se realizará mediante eventos, por ejemplo:
 - La Barra de Herramientas personalizada contendrá un botón que permita seleccionar el reporte que se desea generar.
 - Al presionar el botón de la Barra de Herramientas se generará un evento, el mismo que indicará al Panel de Tareas de Acción los controles que se deben presentar para permitir el ingreso de los parámetros de búsqueda de información, junto con un botón que permitirá generar el reporte.
 - Cuando se presione el botón del Panel de Tareas se generará otro evento, el cual a su vez indicará a la ventana principal de Word 2003 (hoja o documento de trabajo) el contenido del reporte que se debe presentar.

Para la implementación de la Interfaz de Usuario del Documento Inteligente, se utilizan las clases presentadas en la Ilustración 3-36, la gran mayoría de estas clases manejan el modelo de objetos de Word 2003, y están diseñadas para cumplir una tarea específica bien diferenciada, que cumpla con alguno de los requerimientos de la interfaz (manejar la Barra de Herramientas, manejar el Panel de Tareas, etc.).

Las clases: 'C_Escribir', 'C_Etiquetar', y 'C_Proteger', son aquellas que implementarán la lógica para la Interfaz Principal de Word, constituyen la capa más alta para la autogeneración de los documentos (cartas y memorandos), y se encargan de insertar el texto, dar el formato específico de cada documento (márgenes, tabulaciones, tamaño de letra, espaciamiento, etc.), colocar las etiquetas XML en los lugares adecuados y poner restricciones de edición en secciones determinadas de cada uno de los documentos.

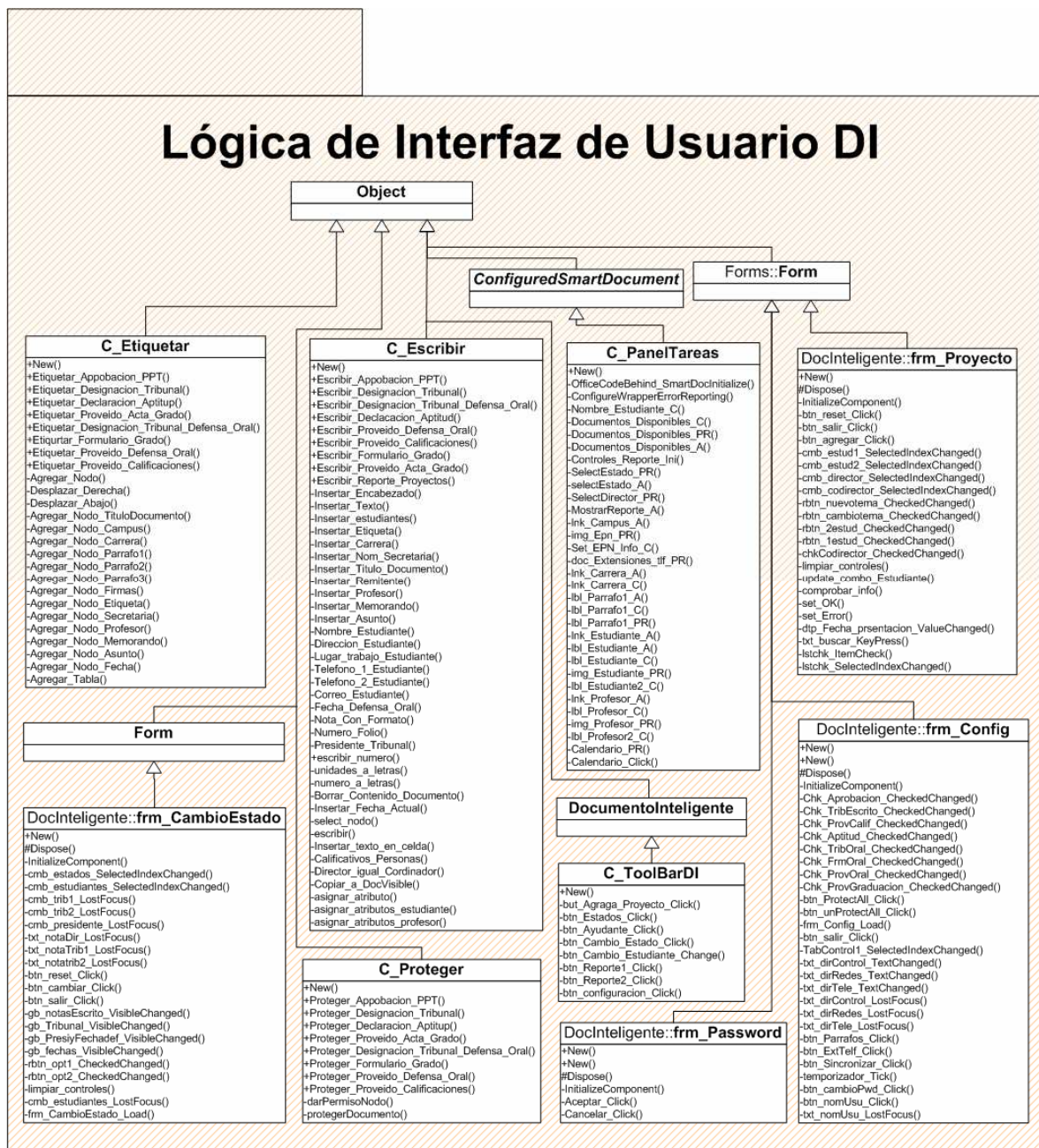


Ilustración 3-36 Clases para la Lógica de la Interfaz de Usuario del Documento Inteligente.

A continuación se presenta un ejemplo que permite ver la definición de uno de los métodos más utilizados de la clase 'C_Escribir'.

```
'Inserta el texto pasado en el parámetro "texto", en el sitio que
'indique el objeto _Selection (variable miembro de la clase
'C_Escribir) antes de la invocación de este método.
Private Sub escribir(
    ByVal texto As String,
    Optional ByVal nombre_fuente As String = "Arial Narrow", _
    Optional ByVal size As Integer = 12, _
    Optional ByVal bold As Boolean = False, _
    Optional ByVal alineacion As WdParagraphAlignment = 3, _
    Optional ByVal num_enter As Integer = 0, _
    Optional ByVal espaciado As Double = 0)
    Dim tmp As Integer
    'se establece el formato del texto a ser insertado
    _Selection.Font.Name = nombre_fuente
    _Selection.Font.Size = size
    _Selection.Font.Bold = bold
    _Selection.ParagraphFormat.Alignment = alineacion
    If espaciado > 0 Then
        _Selection.Font.Spacing = espaciado
    End If
    _Selection.TypeText(texto)
    'se establece el numero de líneas en blanco que se
    'se insertará luego del texto
    If num_enter > 0 Then
        For tmp = 1 To num_enter
            _Selection.TypeParagraph()
        Next
    End If
End Sub
```

Ejemplo 9 Código de implementación del método escribir de la clase .C_Escribir..

Lo más relevante del Ejemplo 9 es que el método utiliza varios parámetros opcionales, lo que permite tener una configuración de formato base (dependiente de los valores predeterminados), a la cual se le puede hacer varias modificaciones según los requerimientos de cada documento. El objeto '_Selection' de este ejemplo es una instancia de Microsoft.Office.Interop.Word.Selection.

La clase 'C_Panel_Tareas' está derivada de la clase 'ConfiguredSmartDocument' miembro de *SmartDocumentWrapper* [Sección 2.7.3], y es la que permitirá manejar toda la lógica del Panel de Tareas de Acción, esta derivación es indispensable ya que en este caso la clase base será la que implemente la

interfaz *ISmartDocument*, fundamental para que el Panel de Tareas de Acción reaccione en función del contenido de las etiquetas XML del documento.

Una pequeña muestra del código contenido en el método 'Controles_Reporte_Ini' de la clase 'C_Panel_Tareas', se presenta en el Ejemplo 10.

```
'Presenta los controles necesarios en el Panel de Tareas de Acción
'para ingresar los parámetros de búsqueda en la información que se
'presentará en los reportes.
Private Sub Controles_Reporte_Ini(ByVal s As String, ByVal e As
ControlArrayInitializeEventArgs)
...
..
    'inserta un control del tipo Checkbox
    Dim Control1 As SmartDocumentConfiguration.ControlRow
    Control1 = e.ControlTable.NewControlRow()
    Control1.Name = "chk_Tele"
    Control1.Caption = "TELECOMUNICACIONES"
    Control1.Type = "C_TYPE_CHECKBOX"
    'inserta un control del tipo Combobox
    'el método que controla el evento OnPreRender del control
    'es "SelectEstado_PR"
    'el método que controlará el evento OnAction del control
    'es "SelectEstado_A"
    Dim Control5 As SmartDocumentConfiguration.ControlRow
    Control5 = e.ControlTable.NewControlRow()
    Control5.Name = "cmb_Estados"
    Control5.Caption = "Seleccione el Estado"
    Control5.OnPreRender = "SelectEstado_PR"
    Control5.OnAction = "SelectEstado_A"
    Control5.Type = "C_TYPE_COMBO"
    e.ControlTable.AddControlRow(Control5)
..
...
End Sub
```

Ejemplo 10 Segmento de código del método 'Controles_Reporte_Ini' de la clase 'C_Panel_Tareas'.

Para el caso de los Formularios de Windows se definieron cuatro clases: 'frm_Proyecto', 'frm_CambioEstado', 'frm_Config', 'frm_Password', todas derivadas de la clase 'Forms::Form'; cada una de estas clases contendrá la lógica necesaria para implementar la interfaz de usuario de los procesos asociados que se presentan en la Ilustración 3-32.

La implementación de los métodos para las clase de los Formularios de Windows no requieren la utilización del modelo de objetos de Word, ni el de Office en general, toda la lógica gira entorno a la utilización de las funciones propiedades y

eventos definidas en su clase base, por lo cual no son difíciles de comprender para un programador familiarizado con la tecnología .NET.

La clase 'C_ToolBarDI' está derivada de la clase 'DocumentoInteligente', que es la clase de entrada (principal) para el Documento Inteligente, y contendrá la implementación de los controladores de eventos para los controles de la Barra de Tareas personalizada; esta clase requiere la declaración de una variable miembro con el atributo ' *WithEvents* ' por cada control de la barra de herramientas tal como se puede apreciar en el Ejemplo 11.

```
'variable miembro que debe estar declarada e inicializada:
'Private WithEvents _
'btn_Cambio_Estado As Microsoft.Office.Core.CommandBarButton
'-----
Private Sub btn_Cambio_Estado_Clic(
    ByVal e As Microsoft.Office.Core.CommandBarButton,
    ByRef a As Boolean) Handles btn_Cambio_Estado.Clic
    'crea una instancia del formulario para el cambio de estado
    f_cambiar_Estado = New frm_CambioEstado
    'muestra la instancia del formulario en forma de diálogo
    f_cambiar_Estado.ShowDialog()
End Sub
```

Ejemplo 11 Implementación del método 'btn_Cambio_Estado_Clic' de la clase 'C_ToolBarDI'.

3.3.1.1.3 Interfaz de Usuario del Formulario de InfoPath

La interfaz del Formulario de Infopath debe aprovechar las características especiales que ofrece Infopath para el diseño de interfaces, como por ejemplo los controles y secciones de repetición que en otra alternativa de formularios electrónicos ya sean Web o Windows implicarían un trabajo extra de programación.

En este caso en particular como el formulario tendrá una función enfocada a la recopilación de las observaciones que se realicen a varios proyectos de titulación en una sesión de Subcomisión Académica (ver Sección 3.2.1), el segmento de la interfaz que recopile datos de un proyecto deberá ser repetitivo n número de veces.

Esto permitirá tener la misma funcionalidad (opciones y datos) para cada uno de los proyectos que se vayan agregando al formulario. En la Ilustración 3-37 se presenta el diseño de la interfaz del Formulario de Infopath.

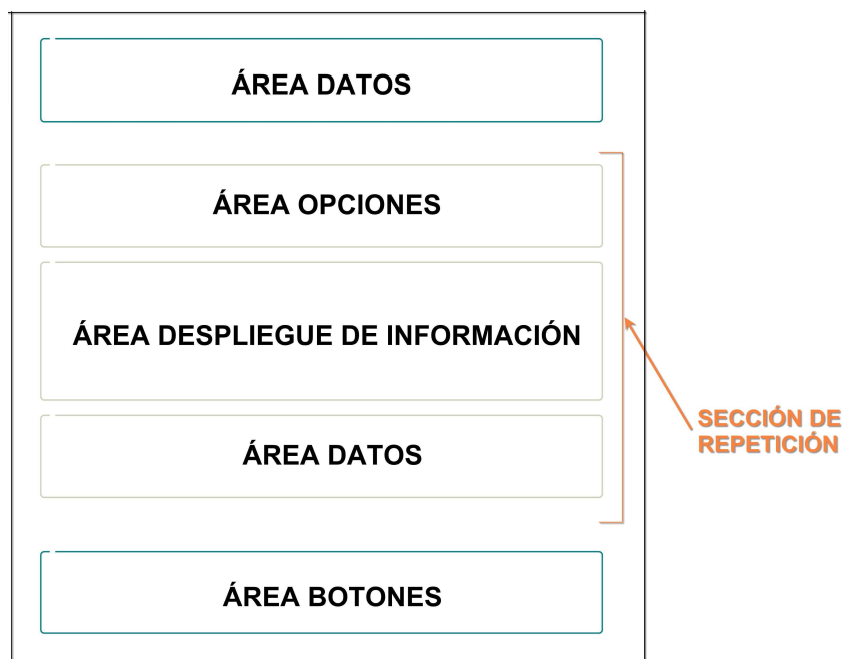


Ilustración 3-37 Diseño de Interfaz de usuario del Formulario de Infopath.

La descripción de cada una de las áreas que conforman la interfaz de usuario para el Formulario de Infopath, se muestra a continuación (Tabla 3-32).

NOMBRE DE ÁREA	DESCRIPCIÓN
Área Datos	Lugar donde se va a colocar todos aquellos controles que permitan el ingreso de la información que se recopile de una sesión.
Área Opciones	En este caso el área de opciones permitirá seleccionar el proyecto de titulación con el que se desea trabajar en la sección de repetición.
Área de Despliegue de Información	Es la sección donde se presentará la información relacionada con el proyecto seleccionado en el área de Opciones.
Área Botones	Es el lugar donde se colocarán un control que permitirá ejecutar la acción de almacenar toda la información contenida en el formulario en la Base de Datos SPODB.

Tabla 3-32 Descripción de las áreas que componen la interfaz del Formulario de Infopath.

Es importante hacer notar que en la interfaz del Formulario de Infopath no se trabaja con un tamaño o con una cantidad de controles conocidos, ya que el usuario puede agregar la cantidad de secciones de repetición que sean necesarias, y es lo que en realidad determinará el tamaño del formulario que en general tendrá un aspecto muy similar al de una página Web (por la longitud que puede llegar a tener).

También se dice que la cantidad de controles es desconocida, ya que de igual forma al añadir una sección se agregan todos los controles que contenga la misma, haciendo que la interfaz sea completamente dinámica y de mucha interacción con el usuario; las secciones al igual que pueden ser agregadas también pueden ser eliminadas e inclusive cambiadas de posición de acuerdo con los requerimientos del usuario.

Algo que se tuvo presente en el momento de diseñar la interfaz para el Formulario de Infopath, es que el usuario utilizará un formulario electrónico y lo que se debe conseguir, es que sea muy fácil de manejar.

3.3.1.1.4 Lógica de la Interfaz de Usuario del Formulario de InfoPath

Para el caso de la Lógica de la Interfaz de Usuario del Formulario de InfoPath, la clase definida es 'FormularioInfoPathSPO', presentada en la Ilustración 3-38, la cual es bastante simple en su estructura, no necesita estar derivada explícitamente de ninguna otra clase (implícitamente está derivada de '*System::Object*'), pero al igual que la mayoría de los métodos que se definen en su interior, debe estar acompañada de un prefijo como atributo, el mismo que tiene que estar en sincronía con el archivo de definición del formulario [verTabla 2-2].

La clase 'FormularioInfoPathSPO', no implementa muchos métodos debido a las facilidades que presenta el InfoPath para configurar acciones visualmente, en donde no se genera código .NET sino sentencias en XPATH que indican los caminos para llegar a los datos de origen y destino con los cuales se desea trabajar; el Ejemplo 12 muestra una de estas sentencias.

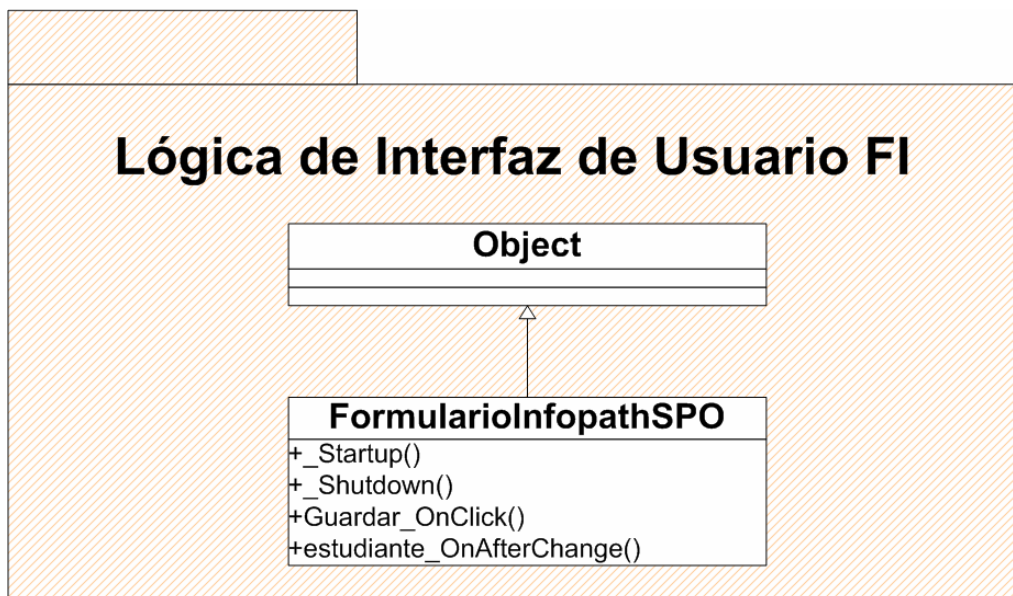


Ilustración 3-38 Clase para la Lógica de Interfaz de Usuario del Formulario de InfoPath.

```

d:T_Proyecto[@pry_titulo = xdXDocument:get- DOM () /
my:SESSION/
my:PROYECTOS/
my:Proyecto/
my:estudiante]/
d:T_Profesor/
@pro_nompro
  
```

Ejemplo 12 Sentencia en XPATH que maneja la información del Formulario de InfoPath.

En el Ejemplo 12 se utilizan tanto el esquema Principal del Formulario de InfoPath (prefijo **my:**) como el del esquema para el *Dataset* con Tipo (prefijo **d:**), aquí se puede apreciar la importancia de los prefijos como se mencionaba en la Sección 3.2.3.2.

Hay que tener presente que cada control del Formulario de InfoPath está asociado a un elemento de su esquema principal, con la sentencia XPATH del Ejemplo 12, se obtiene el valor del atributo '@pro_nompro' del elemento 'd:T_Profesor', asociado al elemento 'd:T_Proyecto' cuyo atributo '@pry_titulo' es igual al contenido del control asociado al elemento 'my:estudiante' del esquema principal [ver la jerarquía de elementos en la Ilustración 3-27].

En general, la implementación de los métodos de la clase 'FormularioInfoPathSPO' es bastante diferente a la utilizada en cualquiera de los métodos de las clases definidas para el Documento Inteligente, hay que considerar que prácticamente todo el modelo de objetos de InfoPath está diseñado para manejar datos XML, por lo tanto los métodos implementados deben tener mucha relación con los elementos definidos en los esquemas que utiliza el formulario.

A continuación se presenta una pequeña parte del código de implementación para el método 'estudiante_OnAfterChange', de la clase 'FormularioInfoPathSPO'.

```
'Prefijo del método que asocia el elemento my:estudiante,
'con el evento OnAfterChange del control asociado
<InfoPathEventHandler(
MatchPath:="/my:SESION/my:PROYECTOS/my:Proyecto/my:estudiante",
EventType:=InfoPathEventType.OnAfterChange)> _
Public Sub estudiante_OnAfterChange(ByVal e As DataDOMEvent)
'Instancia que representa a un conjunto de nodos XML
Dim ListaObservaciones As IXMLDOMNodeList
'Instancia que representa a un nodo XMML
Dim nd_Observacion As IXMLDOMNode
'variables para especificar sentencias XPATH
Dim xpath_proyectos As String
'inicialización de la variable xpath_proyectos
xpath_proyectos = "my:SESION/my:PROYECTOS/my:Proyecto"
'inicialización de la variable txt_query utilizando el contenido
'del elemento my:estudiante (parámetro e del método)
Dim txt_query As String = xpath_proyectos +
"[my:estudiante=" + e.text + "]" +
"/my:datosGenerales/my:Observaciones/my:observacion"
'inicialización del conjunto de nodos ListaObservaciones
ListaObservaciones = Me.thisXDocument.DOM.selectNodes(txt_query)
'asignación de una cadena vacía para el texto de cada uno de los
'elementos nd_Observacion asociados con el elemento my:estudiante
For Each nd_Observacion In ListaObservaciones
    nd_Observacion.text=""
Next nd_Observacion
End Sub
```

Ejemplo 13 Segmento de implementación del método 'estudiante_OnAfterChange' de la clase 'FormularioInfoPathSPO'.

En el Ejemplo 13 se puede apreciar la forma en la que se deben declarar los prefijos de los métodos para los controladores de eventos del Formulario de InfoPath; la función que cumple el código que se muestra en este ejemplo, borra el contenido del texto existente en los elementos 'my:nd_Observacion' cuando se cambia el valor del elemento 'my:estudiante'.

Solo para hacer una analogía y mostrar la gran diferencia que existe al implementar un método en InfoPath e implementar un método en un Formulario de Windows, que sería lo más similar, se pueden hacer las siguientes suposiciones:

- El elemento 'my:estudiante' es un *Combo Box* llamado 'cmb_estudiantes'
- El elemento 'my:nd_Observaciones' es un control de lista llamado 'ListaObservaciones'
- El elemento 'my:nd_Observacion' es uno de los ítems del control 'ListaObservaciones'

Con estas suposiciones una implementación análoga en un formulario de Windows, sería como la que se muestra a continuación:

```
'la sentencia Handles reemplazaría al prefijo del método
Private Sub estudiante_OnAfterChange(
    ByVal sender As Object,
    ByVal e As System.EventArgs)
    Handles cmb_estudiantes.SelectedValueChanged
'la instancia item sería equivalente a: nd_Observacion.text
Dim item as Object
For Each item In ListaObservaciones.items
    item=""
Next item
End Sub
```

Ejemplo 14 Implementación de un método utilizando controles para Formularios Windows.

Con esto lo que se quiere sacar a relucir es que en la implementación de los métodos de InfoPath, es preferible tener en mente que los controles propios de InfoPath son una representación gráfica de un elemento XML.

3.3.1.2 Lógica de Negocio

Al igual que para la Interfaz de Usuario, varios de los procesos del Diagrama 0 también tienen asociado un paquete de la Lógica de Negocio, los procesos que se relacionan con estos paquetes a diferencia de la Interfaz de Usuario no son sólo aquellos relacionados con las entradas y salidas de la SPO, sino también algunos procesos internos de la solución (ej. procesos 2, 3, 4, etc.), como se puede apreciar en la Ilustración 3-39.

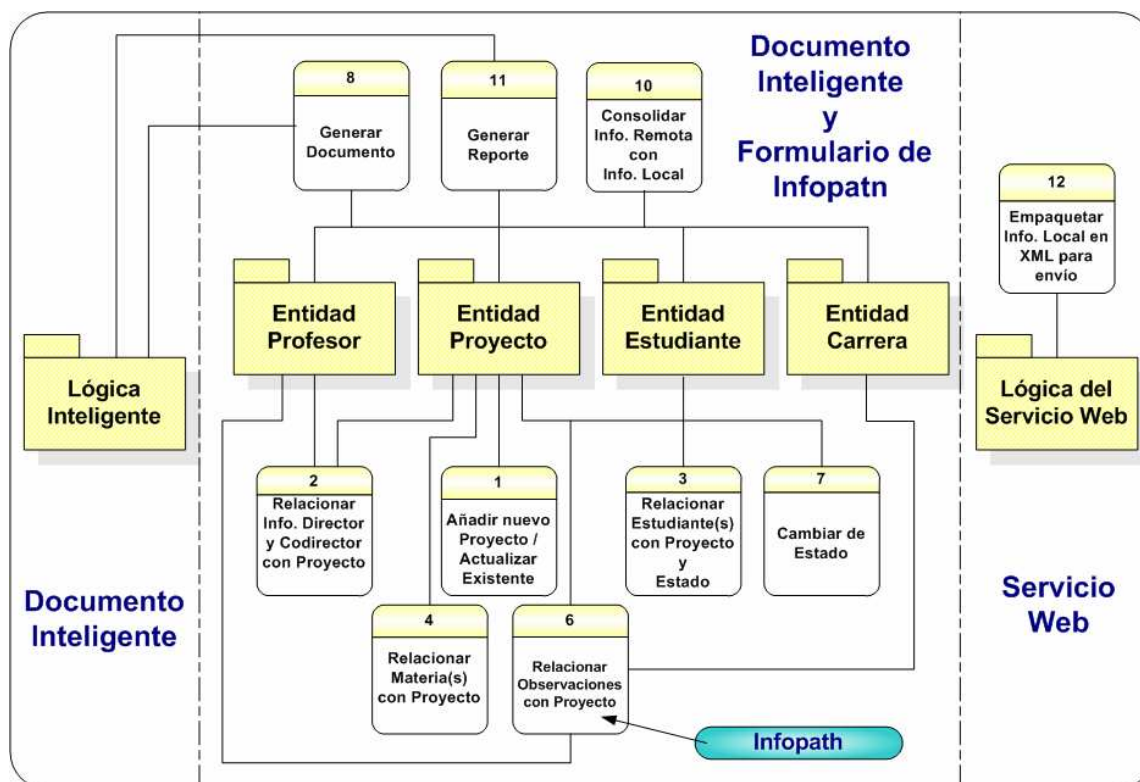


Ilustración 3-39 Relación de Procesos con la capa Lógica de Negocios.

Para el caso de la Lógica de Negocio, los paquetes asociados a los procesos serán el vínculo de comunicación entre la Lógica de la Interfaz de Usuario y los Datos o Servicios, es decir que ésta será la capa intermedia entre la información y la presentación.

3.3.1.2.1 Entidades Propias de la SPO

El paquete que se ha denominado Entidades Propias de la SPO, fue diseñado tomando en consideración las tablas principales de la SPODB ('T_Proyecto', 'T_Profesor', 'T_Estudiante' y 'T_Carrera') cada una de las cuales tendrá una clase que las represente, como se muestra en la Ilustración 3-40.

La función principal para la que fueron diseñadas estas clases, es facilitar la manipulación de los registros que se obtienen o envían a la SPODB, esta capa de código ayudó significativamente para hacer una gran variedad de cambios tanto en la capa de Interfaz de Usuario como en la capa de Acceso a Datos.

Entidades Propias de la SPO

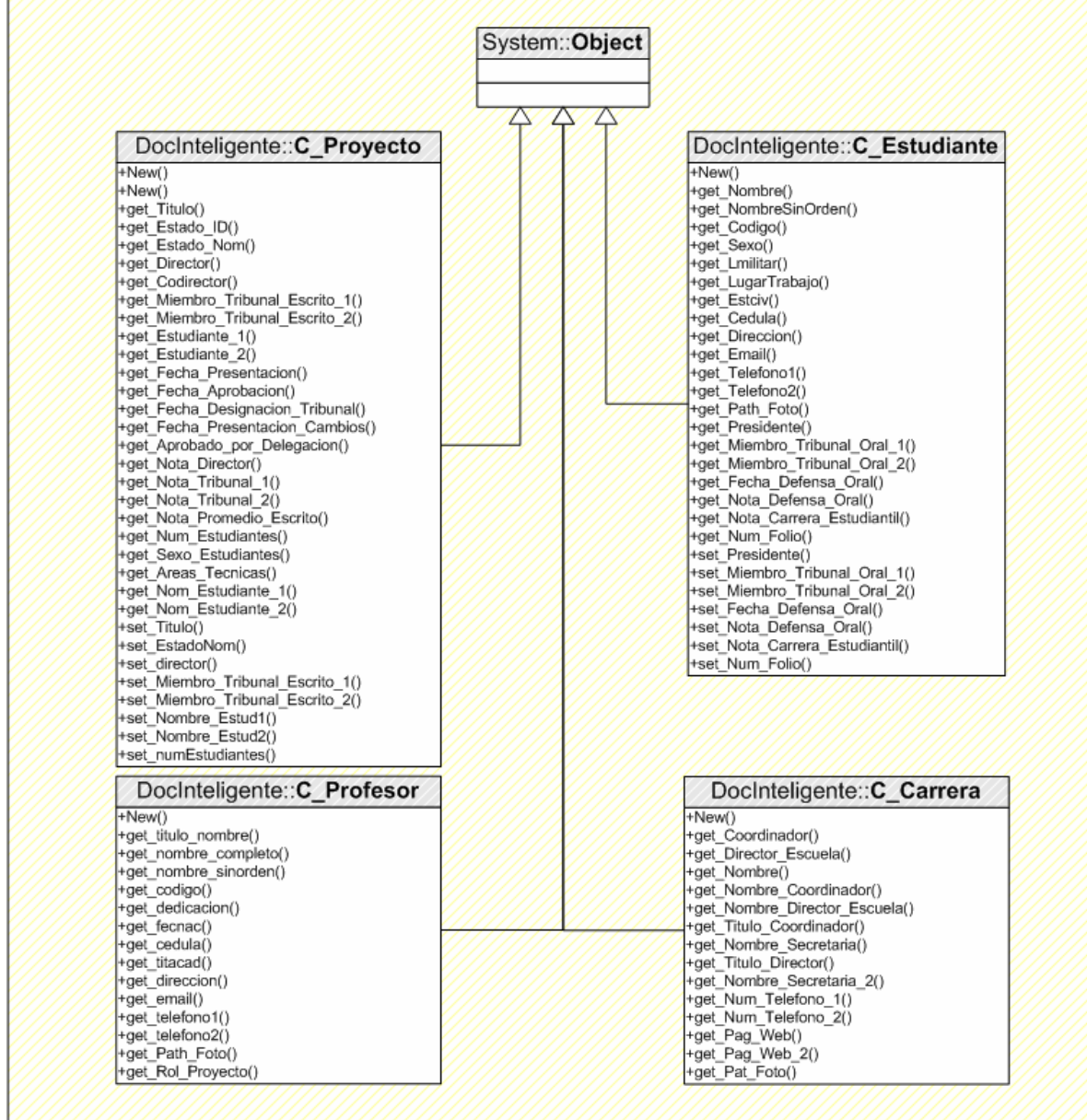


Ilustración 3-40 Clases que definen las Entidades Propias de la SPO.

Como se puede apreciar en el Ejemplo 15, la clase 'C_Proyecto' contendrá instancias de las clases 'C_Estudiante' y 'C_Profesor', esto permite que al utilizar una instancia de la clase 'C_Proyecto' desde alguna de las clases definidas para la capa de Interfaz de Usuario, implícitamente se estén utilizando tres de las clases definidas en la capa Negocio.

```

'variables miembro definidas en la clase C_proyecto(no son todas)
'Private C_Director As C_Profesor = Nothing
'Private C_Codirector As C_Profesor = Nothing
'Private C_Num_Estudiantes As Integer = 0
'Private C_Estudiante_1 As C_Estudiante = Nothing
'Private c_Proyecto As New Collection
'-----
'CONSTRUCTOR DEFINIDO PARA LA CLASE C_PROYECTO
'-----
Public Sub New(ByVal codigo_proyecto As String)
'se crea una instancia de la clase C_DatosSPODB(capa datos y servicios)
_datos = New C_DatosSPODB
'se inicializa la colección Proyecto se utiliza
' el parámetro del constructor
c_Proyecto = _datos.get_datosProyecto(codigo_proyecto)
'se inicializa la instancia con la información del director
C_Director = New C_Profesor(p_codigo:=Proyecto("dir_cod"),
rol_proyecto:=Rol_Profe.Director_Proyecto)
'se inicializa la instancia con la información del codirector
'en caso de existir
If Not Proyecto("codir_cod") = "" Then
    C_Codirector = New C_Profesor(p_codigo:=Proyecto("codir_cod"),
    rol_proyecto:=Rol_Profe.Codirector_Proyecto)
End If
'se inicializa la variable que indica el número de estudiantes
'asociados al proyecto
C_Num_Estudiantes = Proyecto("C_Num_Estudiantes")
'si el proyecto tiene asociado un estudiante se inicializa la
'instancia que manejará la información del estudiante
If C_Num_Estudiantes = 1 Then
    C_Estudiante_1 = New C_Estudiante(Proyecto("estud1_cod"))
End If
'se asignan valores a las variables miembro del estudiante
'asociadas con el proyecto
With C_Estudiante_1
    If Not Proyecto("tribdo1_cod_e1") = "" Then
        .set_Miembro_Tribunal_Oral_1(Proyecto("tribdo1_cod_e1"))
    End If
    If Not Proyecto("tribdo2_cod_e1") = "" Then
        .set_Miembro_Tribunal_Oral_2(Proyecto("tribdo2_cod_e1"))
    End If
    If Not Proyecto("presi_cod_e1") = "" Then
        .set_Presidente(Proyecto("presi_cod_e1"))
    End If
End With
End Sub
...
... 'este es un segmento de la implementación
End Sub

```

Ejemplo 15 Implementación para un constructor de la clase 'C_Proyecto'.

3.3.1.2.2 *Lógica Inteligente*

El paquete de la Lógica Inteligente [Ilustración 3-41] no fue necesario diseñarlo ni implementarlo, ya que se utilizará un paquete existente (*SmartDocumentWrapper.dll*), el mismo que se encargará de manejar todas las acciones que tengan que ver con esta parte de la SPO.

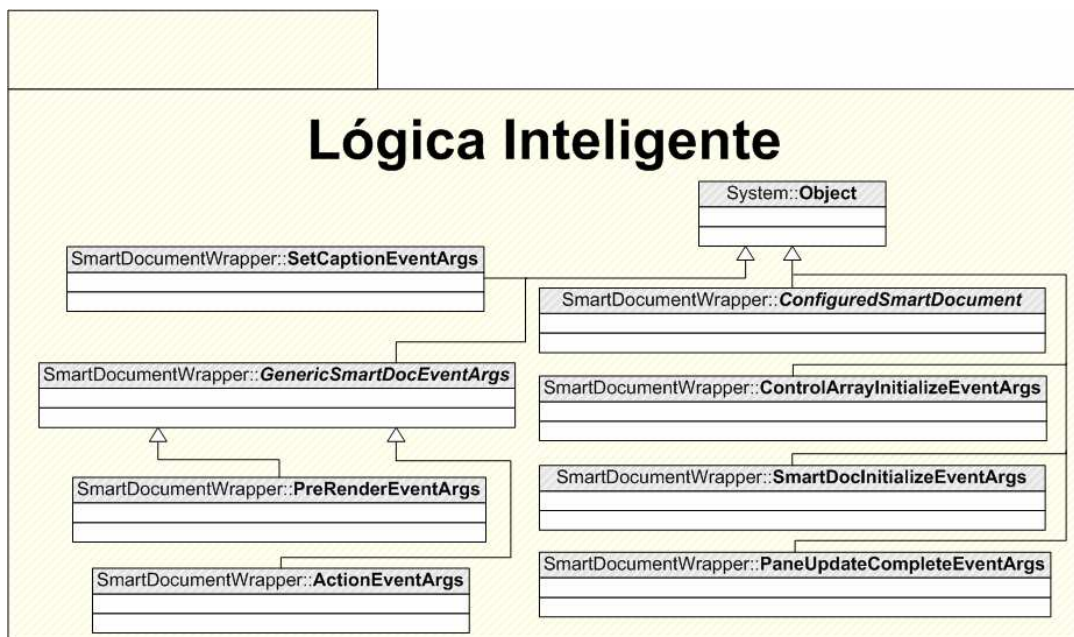


Ilustración 3-41 Clases que definen la Lógica Inteligente de la solución.

En la descripción de la capa de Interfaz de Usuario se mencionó que la clase 'C_Panel_Tareas' debe estar derivada de la clase '*ConfiguredSmartDocument*', y como se puede apreciar en la Ilustración 3-41, ésta es una de las clases que componen el paquete de la Lógica Inteligente.

En la Ilustración 2-4 se muestra un guía de los métodos de la interfaz *ISmartDocument* que deberían ser implementados para el Documento Inteligente; sin embargo, con el paquete que se utilizará como parte de la Lógica de Negocio de la SPO, no hay que preocuparse por la implementación de todos estos métodos ya que la clase '*ConfiguredSmartDocument*' se encargará de hacerlo.

Este conjunto de clases definidas en el ensamblado *SmartDocumentWrapper* es de gran ayuda para el desarrollo del Documento Inteligente de la SPO; la clase principal se denomina '*ConfiguredSmartDocument*'.

Al generar un proyecto para documentos inteligentes, como se menciona en la Sección 2.7.3, hay que limitarse a un esquema predefinido de solución, el cual tiene algunas restricciones; como por ejemplo, la clase principal 'DocumentoInteligente' sería la que debería estar derivada de '*ConfiguredSmartDocument*' y la clase 'C_ToolBarDI' no podría estar derivada de

la clase principal ya que ésta se define predeterminadamente con un atributo '*NotInheritable*'.

Por esta razón, para generar el proyecto de desarrollo del Documento Inteligente de la SPO, se prefirió utilizar el VSTO de Visual Studio .NET y se agregó manualmente la referencia al ensamblado *SmartDocumentWrapper*, lo que da más flexibilidad para la implementación y definición de las clases.

3.3.1.2.3 Lógica del Servicio Web

En general la lógica para la implementación del Servicio Web no es muy compleja, hay que tener presente que la función principal que cumplirá este servicio, es la de empaquetar la información de la SPODB en XML, para que pueda ser enviada por cualquier medio físico de transferencia de información. Las clases que implementan la Lógica del Servicio Web se presentan en la Ilustración 3-42.

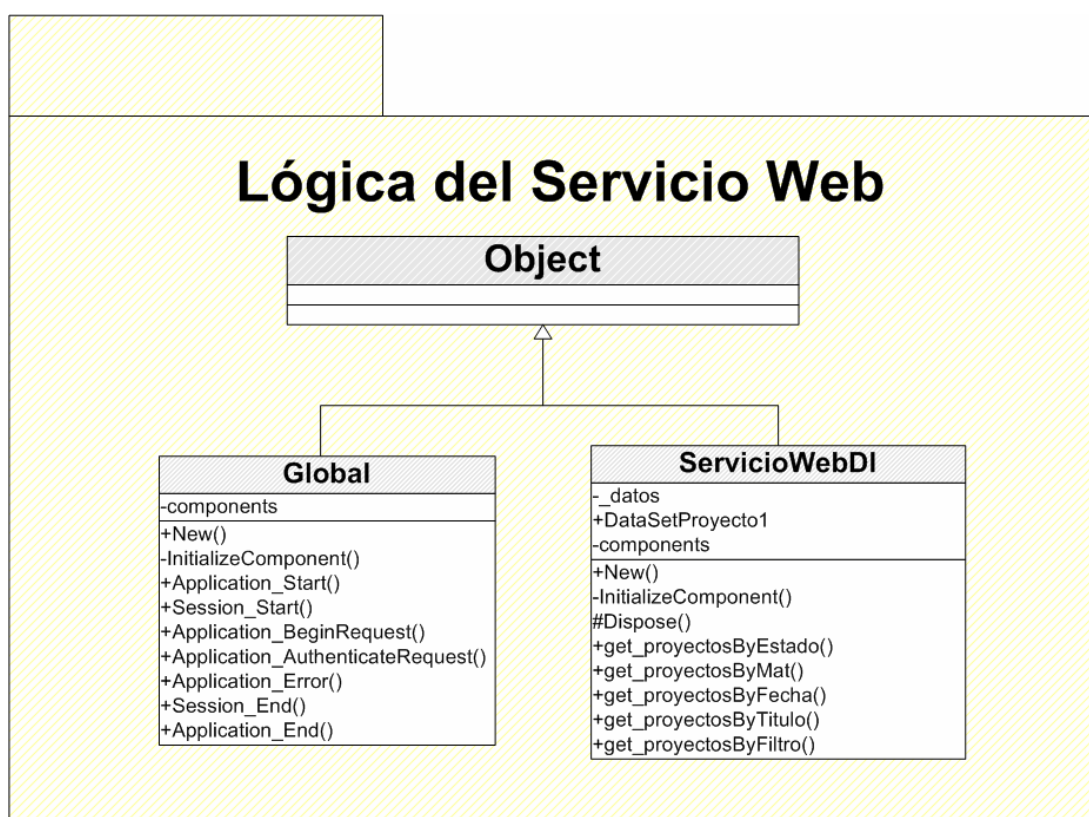


Ilustración 3-42 Clases para la Lógica del Servicio Web.

Como se puede ver en el diagrama de la Arquitectura [Ilustración 3-5], la SPO implementa un Servicio Web, y consume otros remotos, el paquete de la Lógica del Servicio Web hace referencia al Servicio Web que se implementa, ya que como se verá más adelante, los Servicios Web que se consumen están relacionados con la capa de Acceso a Datos y Servicios y no con la capa Lógica de Negocio.

En vista de que todos los métodos implementados en la clase 'ServicioWebDI' están diseñados para que retornen un conjunto de datos, al retornarse información en XML, la variable de retorno será un *Dataset*, y la única particularidad en la implementación de estos métodos es que no se debe olvidar la inclusión del atributo '<WebMethod (>)' como prefijo.

3.3.1.3 Acceso a Datos y Servicios

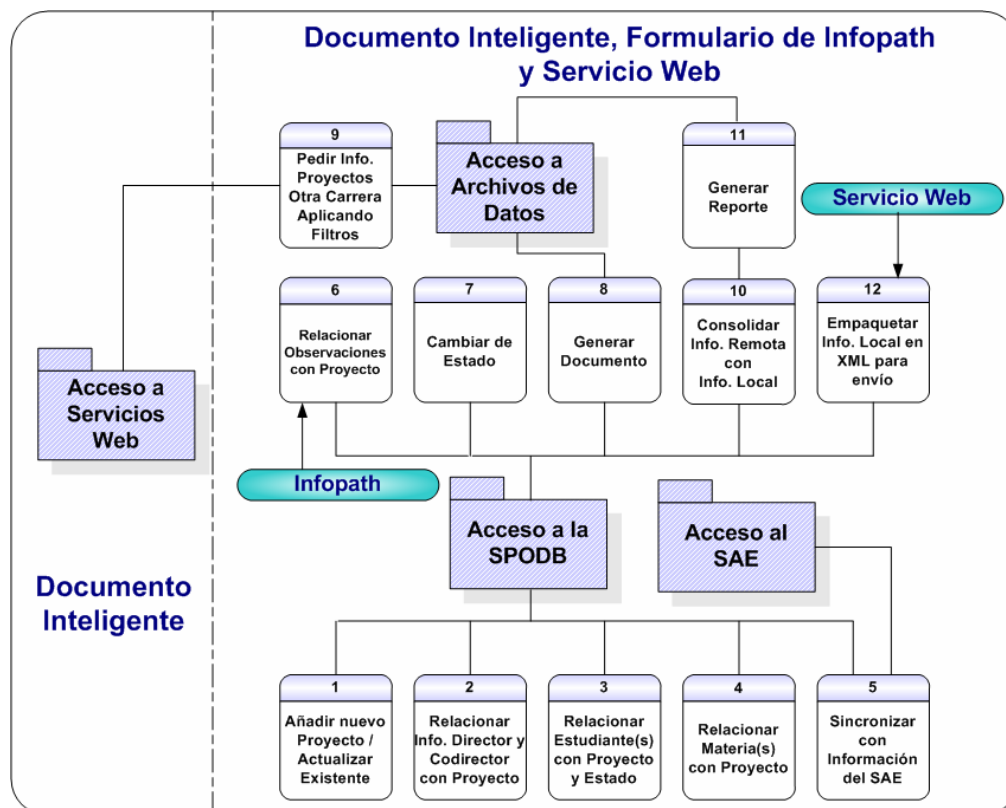


Ilustración 3-43 Relación de Procesos con la Capa Lógica de Acceso a Datos y Servicios.

En la Ilustración 3-43 se muestra la relación de los procesos del Diagrama 0 con los paquetes de la capa lógica de Acceso a Datos y Servicios, se puede ver claramente que todos los procesos estarán relacionados con esta capa de alguna manera, corroborando lo señalado en la Sección 3.2.2, en donde se dice que las fuentes de datos son la parte medular de la SPO.

3.3.1.3.1 Acceso a Servicios Web

Este paquete representa la lógica necesaria para consumir el Servicio Web de la SPO, ya sea este Local o Remoto; hay que recordar que para la generación de reportes se utilizarán los datos proporcionados por un Servicio Web. La clase que maneja el acceso a los Servicios Web será 'C_DatosSW' [Ilustración 3-44].

El fundamento de la lógica del paquete Acceso a Servicios Web, está en la utilización de un Proxy, que es el que se encarga de manejar toda la información empaquetada en el protocolo SOAP que envía cualquier Servicio Web. Al utilizar el Visual Studio .NET no hay que preocuparse por la creación de este Proxy, ya que se genera automáticamente al crear una Referencia Web.

Para la SPO se utilizó el Proxy generado por el entorno de desarrollo sin hacer ninguna modificación. Entre los elementos del Proxy se define una clase que es la que permitirá ejecutar los métodos del Servicio Web.⁶⁹

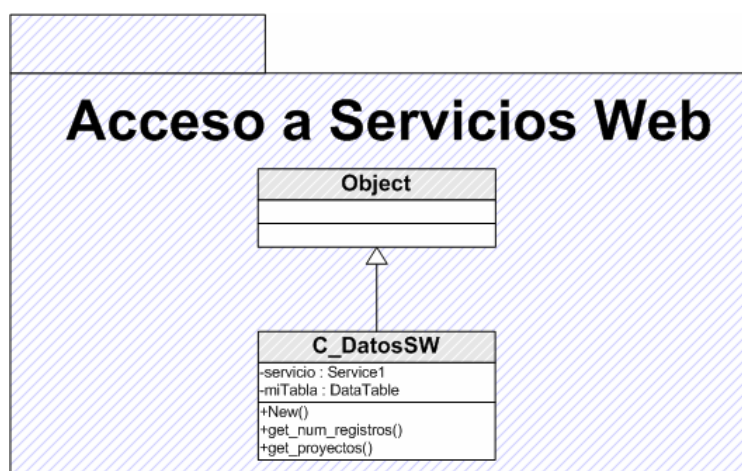


Ilustración 3-44 Clase para el Acceso a Servicios Web.

⁶⁹ Para mayor información de los elementos que componen el Proxy, y de los Servicios Web en general se puede revisar la Referencia 59.

La clase 'C_DatosSW' contendrá una instancia de un objeto definido por esta clase; en realidad, si bien es cierto que se pueden consumir hasta tres Servicios Web a la vez, en el caso de generar un reporte general de todas las Carreras de Electrónica, se puede utilizar la misma instancia porque los Servicios Web a ser implementados en cada una de las Carreras serán idénticos (misma estructura mismos métodos), y lo único que hay que hacer en el código es una iteración con las direcciones de los Servidores en los que estarán instalados los Servicios Web.

Esto quiere decir, que la información de cada una de las carreras se obtendrá en secuencia, una vez obtenidos los datos de la una se pasa a la siguiente, esto evitará que se intente consumir más de un Servicio Web simultáneamente, lo cual en una Intranet tal vez no sería de mucha importancia, pero para el caso de que la información viaje por el Internet, al tener menos recursos de ancho de banda se tendría un tiempo de respuesta mas elevado (el reporte no se presentará hasta que no llegue toda la información).

Los métodos implementados en la clase 'C_DatosSW', reciben un *Dataset* del Servicio Web, y retornan una colección que podrá ser utilizada sin problema por las clases de la capa Negocio, para tener facilidad de iteración con los datos del *Dataset*, se utilizan objetos '*System.Data.DataTable*'.

3.3.1.3.2 Acceso a Fuentes de Datos

Como se puede observar en la Ilustración 3-31, se consideraron tres paquetes secundarios para el paquete Acceso a Fuentes de Datos: Acceso a la SPODB, Acceso al SAE y Acceso a Archivos de Datos, y a pesar de que en la mencionada figura se presenta el paquete principal como común para los tres subsistemas de la SPO, no todos los paquetes secundarios son utilizados por todos los subsistemas.

Esto se puede ver claramente en la Ilustración 3-43, en donde los procesos correspondientes tanto en al formulario de InfoPath (proceso 6) como al Servicio Web (proceso 12), están relacionados únicamente con el paquete secundario de Acceso a la SPODB.

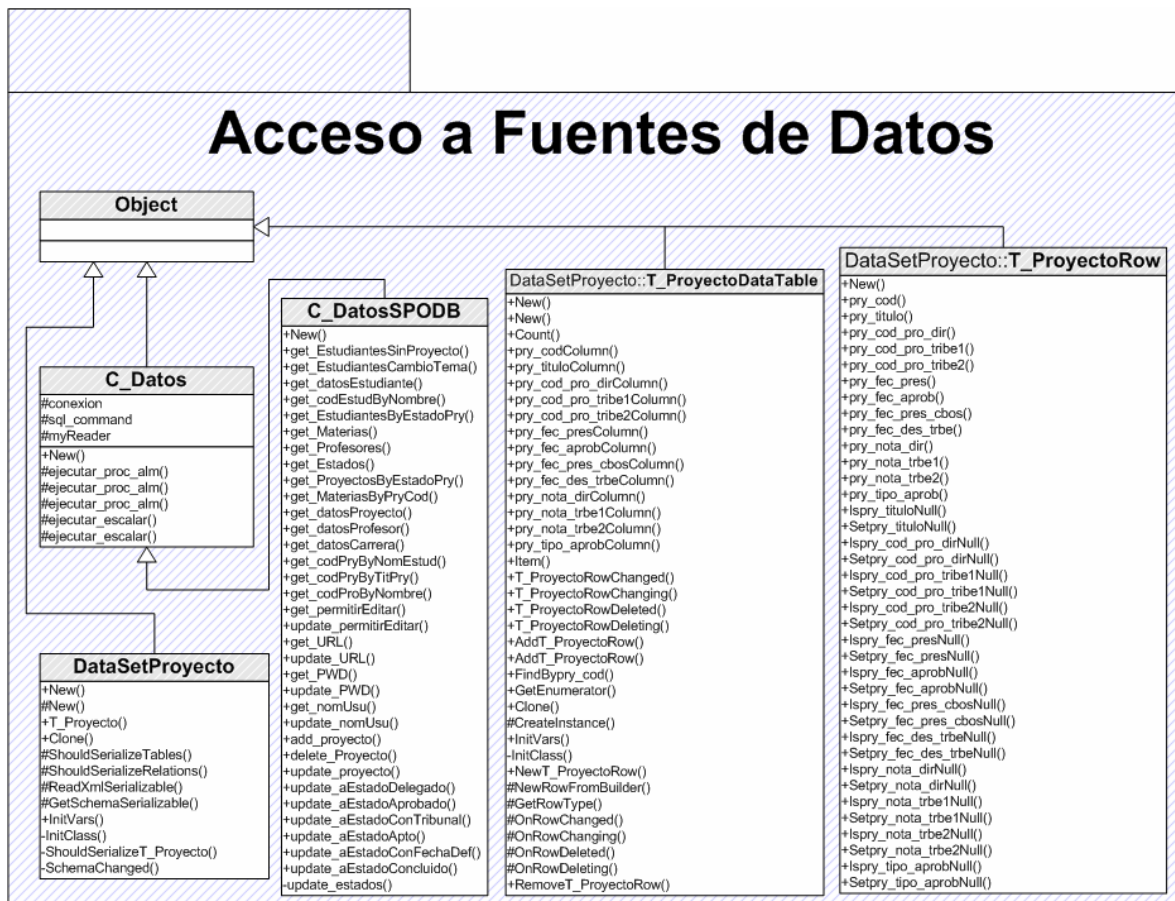


Ilustración 3-45 Clases para el Acceso a Fuentes de Datos.

En la Ilustración 3-45 se pueden apreciar las clases que se utilizarán para el acceso a las bases de datos, la clase 'C_Datos', servirá tanto para el acceso a la base de datos SAE como para el acceso a la SPODB, por esta razón la clase 'C_DatosSPODB' está derivada de dicha clase.

Para la implementación de los métodos definidos en las clases 'C_Datos' y 'C_DatosSPODB', se utilizan básicamente cuatro tipos de objetos: 'SqlConnection', 'SqlCommand', 'SqlParameter' y 'SqlDataReader', y los valores retornados son siempre colecciones o arreglos de datos.

A continuación en el Ejemplo 16 se presenta la implementación completa de uno de los métodos definidos en la clase 'C_Datos'.

```

'Función sobrecargada que retorna un arreglo de datos
'Requiere como parámetros el nombre del procedimiento almacenado a ser
'ejecutado, y el nombre y valor del parámetro como un objeto SqlParameter
'la instancia conexión del objeto SqlConnection ya debe estar inicializada
'y asignado un valor para el ConnectionString
Protected Overloads Function ejecutar_proc_alm(
ByVal nom_proc_alm As String, ByVal parametro As SqlParameter) As Array
Dim datos() As String 'arreglo de datos a ser retornado
Dim sql_command As New SqlCommand 'comando SQL a ser ejecutado
Dim sql_param As New SqlParameter 'parámetro para la ejecución
Dim myReader As SqlDataReader 'lector de resultados de la ejecución
Dim i_contador As Integer = 0 'contador de resultados
'tipo de comando a ejecutar:procedimiento almacenado con un parámetro
sql_command.CommandType = CommandType.StoredProcedure
sql_command.Connection = conexion
sql_command.CommandText = nom_proc_alm
sql_command.Parameters.Add(sql_param)
    'control de errores para la obtención de datos
    Try
        conexion.Open()'se abre la conexión
        myReader = sql_command.ExecuteReader()
            'se llenan todos los elementos del arreglo datos
            'con los datos leídos de la base de datos
            While myReader.Read()
                ReDim Preserve datos(i_contador)
                datos(i_contador) = myReader(0)
                i_contador += 1
            End While
    Catch ex As Exception
        MsgBox(ex.ToString())'mensaje de error
    Finally
        conexion.Close()'cierre de la conexión
    End Try
ejecutar_proc_alm = datos 'se retorna el arreglo de datos lleno
End Function

```

Ejemplo 16 Implementación del método 'ejecutar_proc_alm' de la clase 'C_Datos'.

Otro tipo de acceso a los datos es mediante *DataSets*, cuando se genera un *Dataset* con tipo, el entorno de desarrollo de .NET crea automáticamente clases a partir del esquema XML asociado al *DataSet*, este método de acceso a las bases de datos es muy útil cuando se trabaja con información XML, en este caso tanto el InfoPath como el Servicio Web de la SPO trabajarán con este tipo de acceso a la SPODB.

Finalmente para el paquete Acceso a Archivos de Datos, no se decidió crear una clase, sino un módulo (no se requieren muchos métodos)⁷⁰, en el cual se

⁷⁰ Los módulos son un tipo de Referencia muy similar a las clases pero con algunas diferencias importantes. Los miembros de un módulo son implícitamente compartidos-*Shared*. A diferencia de las clases, nunca se puede crear instancias de módulos, éstos no admiten la herencia y no pueden

especificaron todos los métodos para el acceso a cada uno de los archivos XML que se mencionaron en la Sección 3.2.2.2, el código de implementación de estos métodos es muy similar al presentado tanto en el Ejemplo 2 para los archivos de texto sin formato, como en el Ejemplo 4 para los archivos de texto con formato.

3.3.1.4 Consideraciones para la Escritura del Código Fuente

Esta Sección no estará dedicada a presentar mas ejemplos sobre el código que se escribió para la implementación de los métodos, sino mas bien a hacer una pequeña referencia de la guía que se utilizó para determinar cada una de las variables miembro, instancias de objetos, métodos, controladores de eventos parámetros o tipos de valores retornados de los métodos, para las clases de la SPO.

De los gráficos que muestran las relaciones entre los paquetes de las capas lógicas definidas para la SPO y los procesos mostrados en el Diagrama 0 [Ilustración 3-32, Ilustración 3-39 y Ilustración 3-43], se obtuvo un diagrama general [Ilustración 3-46] como resultado de la unión de los mismos, en este diagrama no se muestran los procesos relacionados a un paquete, sino más bien los paquetes relacionados a cada uno de los procesos.

En realidad las relaciones mostradas en la Ilustración 3-46, los dos diagramas de flujos de datos de la SPO (el Diagrama de Contexto y el Diagrama 0) y las tablas del diccionario de datos, fueron la guía más importante para la escritura del código fuente de la SPO.

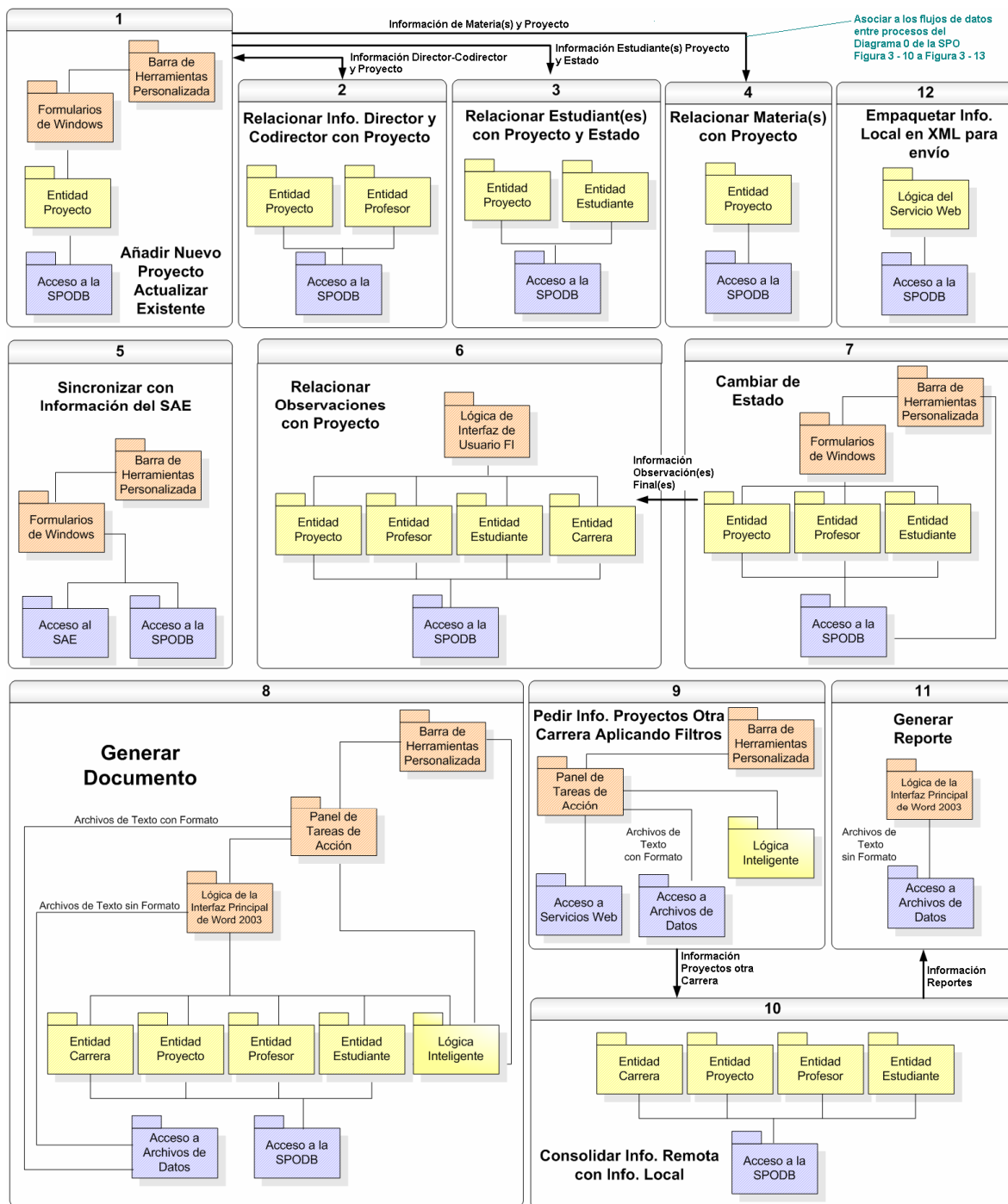


Ilustración 3-46 Paquetes asociados a cada proceso definido en el Diagrama 0 de la SPO.

Solo como un ejemplo, se puede interpretar la Ilustración 3-46 para el proceso Sincronizar con Información del SAE (proceso 5):

- La Barra de Herramientas personalizada del Documento Inteligente deberá contener un botón que permita acceder al Formulario de Windows, que será la interfaz de usuario principal para el proceso de Sincronización

con información del SAE, por lo tanto en la clase 'C_ToolBarDI', se deberá definir un controlador de eventos para dicho botón.

- La clase que define el Formulario de Windows para este proceso ('frm_Config'), no utilizará instancias de la capa Negocio, sino que utilizará dos instancias de la capa Acceso a Datos y Servicios, una instancia para manejar la información de la base de datos SAE (instancia de 'C_Datos') y otra para manejar la base de datos SPODB (instancia de 'C_DatosSPODB').
- Este proceso tiene 5 flujos de datos asociados: Información Estudiantes, Profesores, Materias, Carrera; Registro Estudiante; Registro Carrera; Registro Materia; Registro Profesor [ver Ilustración 3-10].
- Todos los flujos de datos que comienzan con la palabra registro son flujos internos, por lo tanto tendrán relación con la SPODB y por ser de dos direcciones, la clase 'C_DatosSPODB' deberá contener métodos que permitan la obtención y asignación de los datos asociados a cada uno de estos flujos de datos.
- En vista de que para el trabajo con datos de la SPODB se utilizarán procesos almacenados y los flujos asociados con esta base de datos para este proceso son cuatro (entrada y salida), deberán ser creados ocho procedimientos almacenados (cuatro de lectura y cuatro de inserción).
- Para el caso de la base de datos SAE, no se podrán crear procedimientos almacenados, por lo tanto en la clase 'C_Datos' deberá existir un método que permita pasar como parámetro una cadena SQL de selección (flujo en un solo sentido).
- La información que debe ser retornada o recibida como parámetro para cada uno de los métodos a ser implementados, se puede obtener muy fácilmente del diccionario de datos, todos los datos con calificador de opcional serán parámetros opcionales en el método relacionado, y aquellos que no tengan el parámetro opcional serán obligatorios.

Una descripción similar a la que se acaba de presentar, se podría hacer para cada uno de los procesos, lo cual en realidad no es necesario, ya que en el

momento de escribir el código tan solo con una mirada rápida de los diagramas se puede saber fácilmente lo que debe ser implementado y en donde, para que un proceso cumpla su función.

3.3.2 CONSIDERACIONES PARA EL DOCUMENTO INTELIGENTE

La clave para la implementación de Documento Inteligente es el Archivo de Manifiesto XML⁷¹, que contiene las especificaciones que permiten señalar tanto los archivos que componen el paquete de expansión XML, como el esquema XML que debe ser adjuntado.

3.3.2.1 Archivo de Manifiesto XML

La creación del Archivo de Manifiesto XML se basa en la utilización del esquema: "http://schemas.microsoft.com/office/xmlexpansionpacks/2003", razón por la cual en el elemento raíz (**<SD:manifest>**) de este archivo se hace referencia al mencionado esquema, tal como se muestra a continuación:

```
<!--elemento raíz del archivo de manifiesto-->
<SD:manifest
xmlns:SD="http://schemas.microsoft.com/office/xmlexpansionpacks/2003">
  <SD:version>1.0</SD:version>
  <SD:uri>
    EPN.wgvc/DI/Documentos/Electrónica</SD:uri> ...
```

Ejemplo 17 Archivo de Manifiesto XML – segmento 1.

En el Ejemplo 17 se presenta la parte inicial del Archivo de Manifiesto XML, la cual consta de las siguientes especificaciones generales:

- Número de la versión **<SD:version>**, en este caso representa la versión del Documento Inteligente.
- Dirección URI⁷² **<SD:uri>**, está en concordancia con el espacio de nombres definido por el esquema XML que utiliza el Documento Inteligente [ver Sección 3.2.3.1].

⁷¹ En la Sección 2.5.2.2 se encuentra una definición del Archivo de Manifiesto, el Paquete de Expansión y el Esquema de la Solución.

⁷² Ver nota al pie **¡Error! Marcador no definido.**, en la página **¡Error! Marcador no definido.**

3.3.2.1.1 Paquete de Expansión

El paquete de expansión es el que en realidad permite especificar todos los archivos que componen el Documento Inteligente de la SPO, dentro del Archivo de Manifiesto XML; está representado por un elemento **<SD:solution>**, el mismo que en esta Sección será dividido en varios segmentos para facilitar la explicación de los elementos hijos más importantes que lo componen.

```
<SD:solution> <!--inicio - Paquete de Expansión-->
    <SD:solutionID>
    DOCUMENTO-INTELIGENTE-ID-WGVC21-EPN</SD:solutionID>
    <SD:type>smartDocument</SD:type>
    <SD:alias lcid="">
    Documento Inteligente Electrónica EPN</SD:alias>
    <!--continúa-->
```

Ejemplo 18 Archivo de Manifiesto XML – segmento 2.

Los elementos XML definidos en el Ejemplo 18 son los siguientes:

- Identificador de la solución **<SD:solutionID>**, el valor de este elemento es el identificador para el paquete de expansión, y como se verá mas adelante es el nombre que aparecerá en las entradas del Registro de Windows.
- El elemento **<SD:type>** al contener el valor “smartDocument”, es el que indica que el elemento **<SD:solution>**, define el paquete de expansión.
- El nombre que verán los usuarios en el momento de adjuntar este paquete de expansión a un documento de Word [ver Ilustración 3-51], está dado por el valor del elemento **<SD:alias lcid="">**, el atributo **lcid** con el valor “*” indica que el contenido del elemento no está especificado en ningún idioma específico.

```
<SD:file><!--elemento para el Ensamblado Principal-->
    <SD:type>solutionActionHandler</SD:type>
    <SD:version>1.0</SD:version>
    <SD:filePath>DLLs\DocInteligenteEPN.dll</SD:filePath>
    <SD:CLSNAME>
    Wgvc.EPN.DocInteligente.DocumentoInteligente
    </SD:CLSNAME>
    <SD:managed />
</SD:file>
<!--continúa-->
```

Ejemplo 19 Archivo de Manifiesto XML – segmento 3

El elemento **<SD:file>** del Ejemplo 19 también está definido dentro del elemento **<SD:solution>**; indica el primer archivo que forma parte del paquete de expansión, y sus elementos hijos tienen las siguientes especificaciones:

- El elemento **<SD:type>**, al utilizar el valor "solutionActionHandler", indica que el archivo DocInteligenteEPN.dll, al que hace referencia su elemento padre (**<SD:file>**), es el ensamblado principal para el Documento Inteligente.
- El número de la versión **<SD:version>**, en este caso está especificado para el ensamblado principal.
- La ubicación del archivo DocInteligenteEPN.dll está dada por el elemento **<SD:filePath>**, esta dirección es relativa al directorio donde se encuentra el archivo de Manifiesto XML [ver Ilustración 3-47].
- El elemento **<SD:CLSNAME>**, indica el nombre de la clase de entrada (clase principal) del Documento Inteligente, que en este caso es: Wgvc.EPN.DocInteligente.DocumentoInteligente (incluido el espacio de nombres).
- El elemento **<SD:managed />**, indica que el archivo DocInteligenteEPN.dll es un ensamblado administrado de .NET.

```
<SD:file><!--elemento para la plantilla del documento-->
  <SD:type>template</SD:type>
  <SD:version>1.0</SD:version>
  <SD:templateID>
  DOCUMENTO-INTELIGENTE-ID-WGVC21-EPN</SD:templateID>
  <SD:filePath>
    ARCHIVOS DE CONFIGURACIÓN\DIplantilla.dot
  </SD:filePath>
</SD:file> <!--continúa-->
```

Ejemplo 20 Archivo de Manifiesto XML – segmento 4.

Se utilizará una plantilla distinta a la Plantilla Normal (predeterminada para Word), en vista que al utilizar código para manipular o crear una barra de herramientas personalizada, automáticamente la plantilla empleada por el Documento Inteligente sufrirá modificaciones; por esta razón, para evitar problemas con cualquier otro documento de Word, se decidió emplear una plantilla propia que se denominará DIplantilla.dot.

El Ejemplo 20 muestra el elemento definido en el Archivo de Manifiesto XML, que permite incluir el archivo Dplantilla.dot como parte del paquete de expansión, lo más importante a rescatar de este elemento se indica en los siguientes puntos:

- El elemento **<SD:type>** al utilizar el valor "template", indica que el archivo Dplantilla.dot al que hace referencia su elemento padre (**<SD:file>**), es la plantilla que utilizará el Documento Inteligente.
- El elemento **<SD:templateID>**, define el identificador de la solución donde será utilizada la plantilla, en este caso tiene que contener un valor igual al valor del elemento **<SD:solutionID>** mostrado en el Ejemplo 18.

Para todos los archivos restantes [ver Ilustración 3-53] que deben ser incluidos en el paquete de expansión: ensamblados de interoperabilidad⁷³, ensamblados adicionales, archivos XML con o sin formato y archivos de imagen, se utiliza un elemento con una estructura idéntica a la mostrada en el Ejemplo 21, en donde lo único que varía por cada archivo, es el contenido del elemento **<SD:filePath>**.

```
<SD:file><!--elemento para un ensamblado adicional-->
  <SD:type>other</SD:type>
  <SD:version>1.0</SD:version>
  <SD:filePath>
    DLLs\SmartDocumentWrapper.dll
  </SD:filePath>
</SD:file>

<!-- de aquí en adelante se especifican los
archivos restantes que formarán parte del
Paquete de Expansión -->
</SD:solution><!--cierre - Paquete de Expansión-->
```

Ejemplo 21 Archivo de Manifiesto XML – segmento 5

3.3.2.1.2 Esquema XML

Para especificar el esquema XML a ser utilizado por el Documento Inteligente, al igual que para el paquete de expansión también se ha definido un elemento **<SD:solution>**, con la misma estructura, pero con las siguientes particularidades en cuanto a los valores de sus elementos [Ejemplo 22]:

- El valor del elemento **<SD:solutionID>** es igual que el valor de su equivalente para el elemento del paquete de expansión.

⁷³ Se utilizan archivos de interoperabilidad para controles *ActiveX* utilizados en la solución, no se refiere a los de interoperabilidad de Office.

- El elemento **<SD:solution>** únicamente definirá un elemento **<SD:file>**, para indicar el archivo de esquema que utilizará el Documento Inteligente, el mismo que se denomina DI_Eschema.xsd.
- El nombre del esquema XML que verán los usuarios en el momento de asociarlo a un documento de Word será: “Esquema Documentos Electrónica EPN”, tal como se ve en la Ilustración 3-52 y viene definido por el elemento **<SD:alias lcid="*">**.
- El elemento **<SD:type>** como hijo del elemento **<SD:solution>**, y como hijo del elemento **<SD:file>** tiene el valor “*schema*”.

```

<SD:solution>
<SD:solutionID><!-- esquema XML de la solución-->
DOCUMENTO-INTELIGENTE-ID-WGVC21-EPN</SD:solutionID>
  <SD:type>schema</SD:type>
  <SD:alias lcid="*">
    Esquema Documentos Electrónica EPN
  </SD:alias>
  <SD:file>
    <SD:type>schema</SD:type>
    <SD:version>1.0</SD:version>
    <SD:filePath>DI_Eschema.xsd</SD:filePath>
  </SD:file>
</SD:solution>
</SD:manifest> <!-- viene del Ejemplo 18 -->

```

Ejemplo 22 Archivo de Manifiesto XML – segmento 6

3.3.2.2 Directorio del Archivo de Manifiesto

El directorio donde se aloja el Archivo de Manifiesto XML (ManifiestoDIElectrónica.xml) es fundamental, en vista de que hay que considerar que todas las direcciones contenidas en los elementos **<SD:filePath>** especificadas tanto para los archivos del paquete de expansión, como para el archivo del esquema XML (DI_Eschema.xsd) son relativas a este directorio.

En el directorio del Archivo de Manifiesto se encuentran todos los archivos necesarios para que el Documento Inteligente funcione, los mismos que han sido agrupados por tipo en subdirectorios como se muestra en la Ilustración 3-47; si alguno de los archivos contenidos en cualquiera de estos subdirectorios fuera eliminado o cambiado de ubicación, al utilizar el Archivo de Manifiesto XML se

provocarían errores tratando de encontrarlo, por esta razón es preferible utilizar un directorio compartido con permisos de lectura al que tenga acceso sólo el administrador.



Ilustración 3-47 Directorio donde se encuentra el Archivo de Manifiesto.

3.3.2.3 Directorio de Instalación

El directorio donde serán copiados (automáticamente al adjuntar el Archivo de Manifiesto [ver secuencia Ilustración 2-8]) todos los archivos a los que hace referencia el Archivo de Manifiesto XML, es creado automáticamente al asociar el manifiesto a un documento de Word por primera vez.

En realidad serán creados dos directorios adicionales bajo el siguiente directorio oculto: **<Drive>\Documents and Settings\<Nombre de Usuario>\Configuración local\Datos de programa\Microsoft\Schemas**, utilizado de forma predeterminada por Office para almacenar esquemas XML.

El primer directorio que se creará, tendrá un nombre concordante con el espacio de nombres del esquema XML (EPN.wgvc/DI/Documentos/Electronica), que para este caso será: **EPN_wgvc_DI_Documentos_Electronica**.

Y el segundo directorio que a su vez se creará bajo el directorio anterior, tendrá un nombre igual al valor de los elementos **<SD:solutionID>** especificados en el Archivo de Manifiesto es decir: **DOCUMENTO-INTELIGENTE-ID-WGVC21-EPN**.

Quedando la dirección completa del directorio de instalación de la siguiente manera:

“\\%USERPROFILE% \
 Configuración local \ Datos de programa \
 Microsoft \ Schemas \
 EPN_wgvc_DI_Documentos_Electronica \
 DOCUMENTO-INTELIGENTE-ID-WGVC21-EPN”

Donde %USERPROFILE% es una variable de entorno⁷⁴ que representa al directorio <Drive>\Documents and Settings\<Nombre de Usuario>.

3.3.2.4 Consideraciones de Seguridad

Luego de haber definido completamente el archivo de Manifiesto XML, es imprescindible firmarlo digitalmente, y configurar una directiva de seguridad .NET que permita la ejecución de los ensamblados administrados.

3.3.2.4.1 Firma Digital del Archivo de Manifiesto

La firma de este Archivo de Manifiesto se realizó con el utilitario XML *Sign*, el mismo que inserta varios elementos XML para especificar el valor de la firma y algunas características como por ejemplo información de clave, método de firma, etc. Una parte de la estructura del Archivo de Manifiesto XML ya firmado se puede observar en la Ilustración 3-49.

Para firmar digitalmente el archivo se creó un certificado digital utilizando la herramienta *Makecert.exe*⁷⁵, el mismo que en este caso podrá ser utilizado sin problemas en vista de que no es una aplicación comercial, y el certificado tendría un carácter personal. El certificado digital se encontrará en un archivo denominado *CertificadoDI.cer*, y debe ser instalado como cualquier otro certificado digital en la máquina donde se implementará el Documento Inteligente, en la Ilustración 3-48 se puede ver el certificado creado.

⁷⁴ Las variables de entorno son cadenas de texto utilizadas por el sistema operativo para saber la ruta de acceso a un archivo o directorio, mas información se puede encontrar en la ayuda incluida con el sistema operativo Windows XP.

⁷⁵ La herramienta *Makecert.exe* genera certificados para efectos de pruebas, posee varias opciones que se pueden consultar en la Referencia 55.

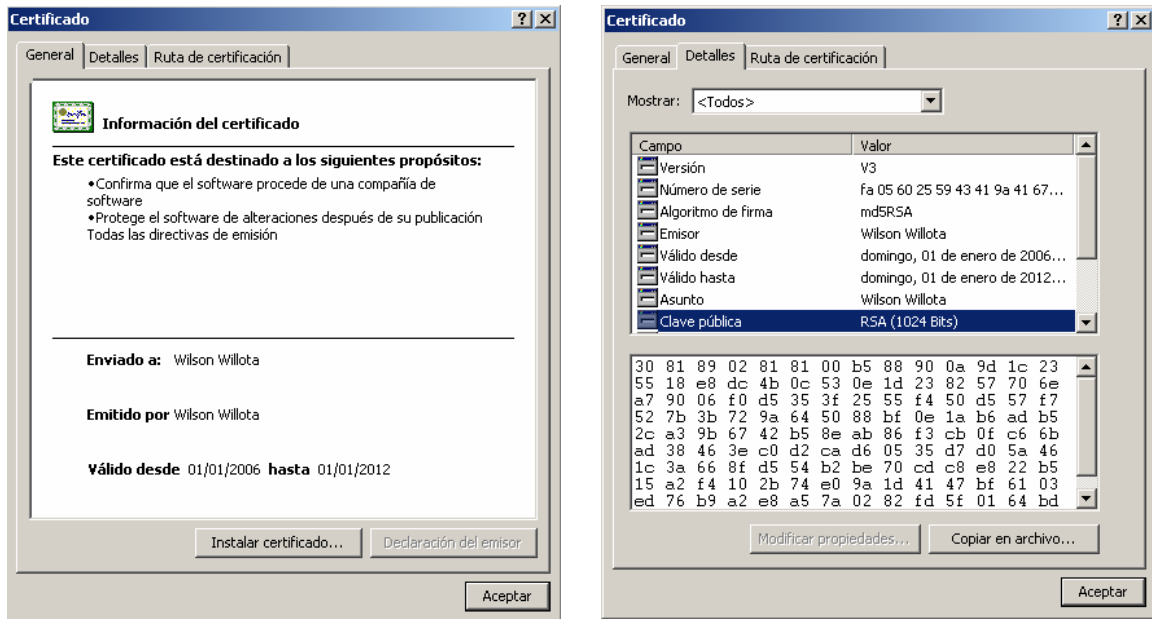


Ilustración 3-48 Certificado digital utilizado por el Documento Inteligente.

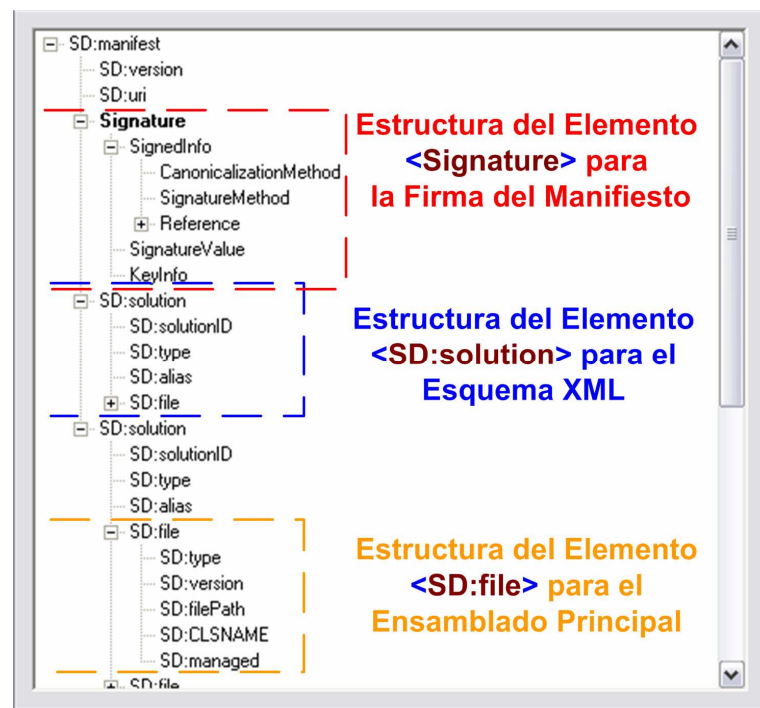


Ilustración 3-49 Estructura del Archivo de Manifiesto Firmado.

3.3.2.4.2 Directiva de Seguridad

Al estar utilizando código administrado para este Documento Inteligente, es necesario configurar los permisos para ejecución del código en el *Framework* .NET, para lo cual se creó un pequeño archivo de ejecución por lotes o archivo de comandos (.cmd), con el siguiente contenido:

%WINDIR%\Microsoft.NET\Framework\v1.1.4322\caspol

```
-pp           off
-u
-addgroup    "All_Code"
-url         "file://%USERPROFILE%\Configuración local\
               Datos de programa\Microsoft\Schemas\
               EPN_wgvc_DI_Documentos_Electronica\DOCUMENTO-INTELIGENTE-
               ID-WGVC21-EPN\*"
```

FullTrust

```
-name        "DOCUMENTO INTELIGENTE PROYECTOS"
-description "Código para la ejecución del Documento Inteligente que permite la
               elaboración de cartas y memorandos relacionados con los proyectos de
               titulación."
```

Ejemplo 23 Comandos para la configuración de seguridad del Documento Inteligente.

Se utiliza la herramienta Directiva de Seguridad de Acceso a Código (Caspol.exe)⁷⁶ que viene incluida con el *Framework* .NET, la misma que se ejecuta con las siguientes opciones:

- **-pp** con el valor “ *off* ” para deshabilitar la indicación que se mostraría al estar realizando cambios en las directivas de seguridad.
- **-u** para que la directiva de seguridad se aplique al usuario donde se está ejecutando el comando.
- **-addgroup** con el valor "All_Code" para que la directiva de seguridad se agregue en el grupo de código⁷⁷ All_code, que es el grupo de código raíz con el que siempre contará de manera predeterminada el *Framework* .NET.
- **-url** indica el directorio de instalación [Sección 3.3.2.3] en el que se encontrarán todos los ensamblados administrados para el Documento Inteligente.

⁷⁶ Una guía más completa de todas las opciones que se pueden utilizar con la herramienta Caspol.exe se puede consultar en la Referencia 54.

⁷⁷ Las directivas de seguridad de .NET se configuran utilizando una jerarquía de grupos de código; en este caso, el código administrado de una aplicación pertenece a un determinado grupo, y sobre él actúan las políticas de seguridad del grupo al que pertenece [Referencia 54].

- Finalmente se especifica el nombre y una pequeña descripción de la directiva de seguridad.
- Los ensamblados tendrán permisos *FullTrust*⁷⁸ para que puedan acceder a los recursos del sistema sin problemas.

3.3.2.4.3 Firma del código

La firma del código no es un requisito indispensable para la ejecución del Documento Inteligente, pero permite que la solución tenga un nivel extra de seguridad. En este caso el archivo que se firmó digitalmente con la ayuda de la herramienta Signcode.exe⁷⁹ y el certificado digital creado [Ilustración 3-48] fue el ensamblado principal (DocInteligenteEPN.dll). Se pueden ver las propiedades del ensamblado firmado en la Ilustración 3-50; para acceder a esta pantalla hay que hacer *clic* derecho sobre el ensamblado y utilizar el comando seguridad.

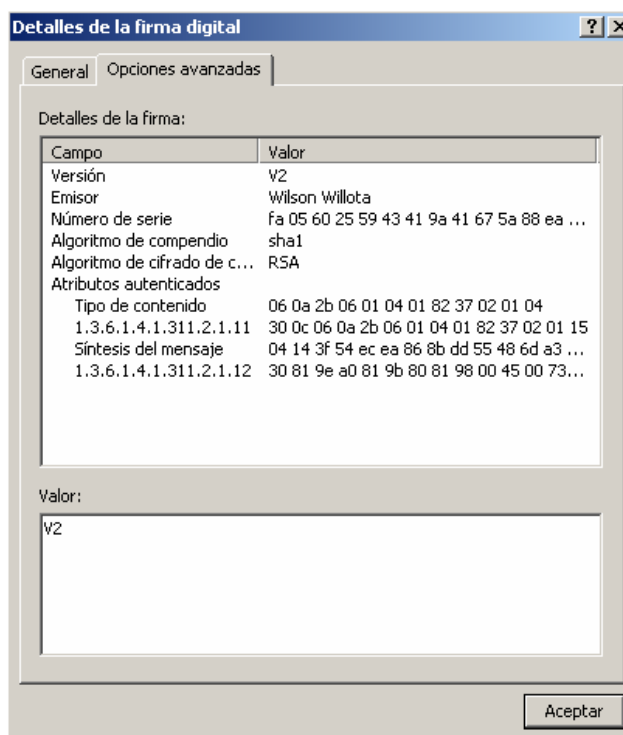


Ilustración 3-50 Detalles de la firma digital para el código del Documento Inteligente.

⁷⁸ El conjunto de permisos *FullTrust* proporciona al código acceso sin restricciones a todos los recursos protegidos; más información en la consola de configuración de políticas de seguridad del *Framework .NET*, a la que se accede por las herramientas administrativas del sistema operativo.

⁷⁹ La herramienta Signcode.exe, permite firmar ensamblados o los archivos contenidos en un ensamblado de varios archivos individualmente, para mayor información se puede consultar la Referencia 56.

3.3.2.5 Instalación

En el momento de asociar el Archivo de Manifiesto XML firmado digitalmente a un documento de Word, se ejecutará todo el proceso mostrado en la Ilustración 2-8 (Chequeos de seguridad para los archivos de manifiesto XML).

Si no se presenta ningún inconveniente, el paquete de expansión aparecerá en la lista de paquetes de expansión XML disponibles del cuadro de diálogo **Plantillas y complementos** como se puede apreciar en la Ilustración 3-51, y el esquema XML aparecerá en el cuadro de diálogo **Biblioteca de esquemas** como se observa en la Ilustración 3-52, en caso de existir algún problema (inconformidad de firma, pérdida de archivos, fallas en la estructura del archivo de manifiesto, etc.), se mostrará un mensaje indicando el error.

En total en el Archivo de Manifiesto XML se definieron veintisiete elementos, uno por cada uno de los archivos que necesita el Documento Inteligente, una vez que se haga *clic* en el botón **Aceptar** del cuadro de diálogo **Plantillas y complementos** mostrado en la Ilustración 3-51, el paquete de expansión y el esquema XML ya estarán instalados y se habrán copiado en la máquina todos los archivos que se muestran en la Ilustración 3-53, y también se habrán agregado en el Registro de Windows varias entradas (una por cada archivo); una muestra del registro se presenta en la Ilustración 3-54.

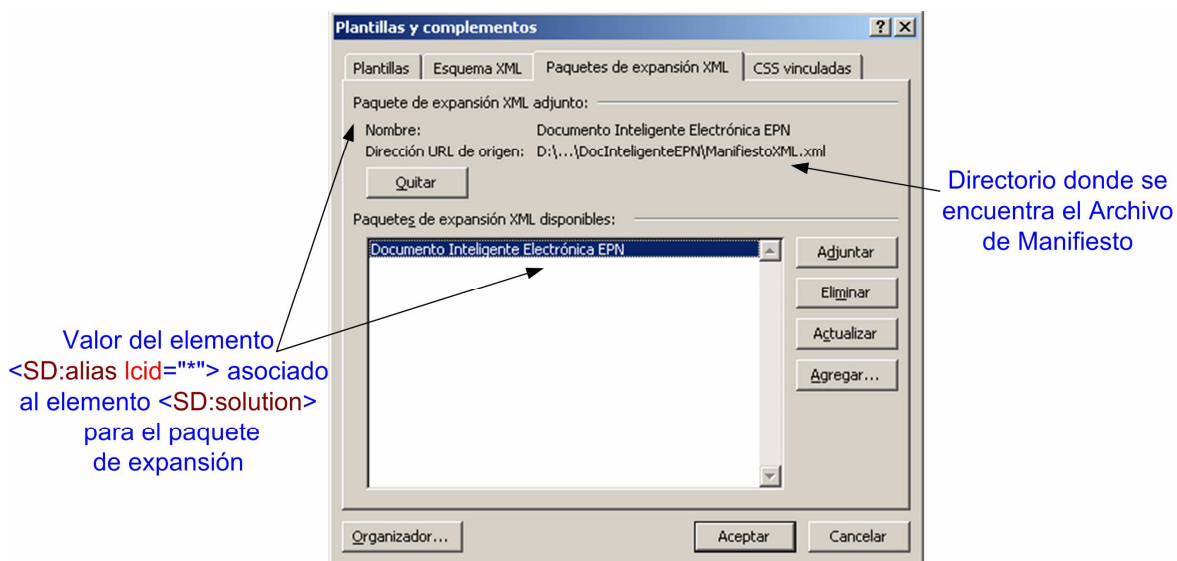


Ilustración 3-51 Cuadro de diálogo Plantillas y Complementos.

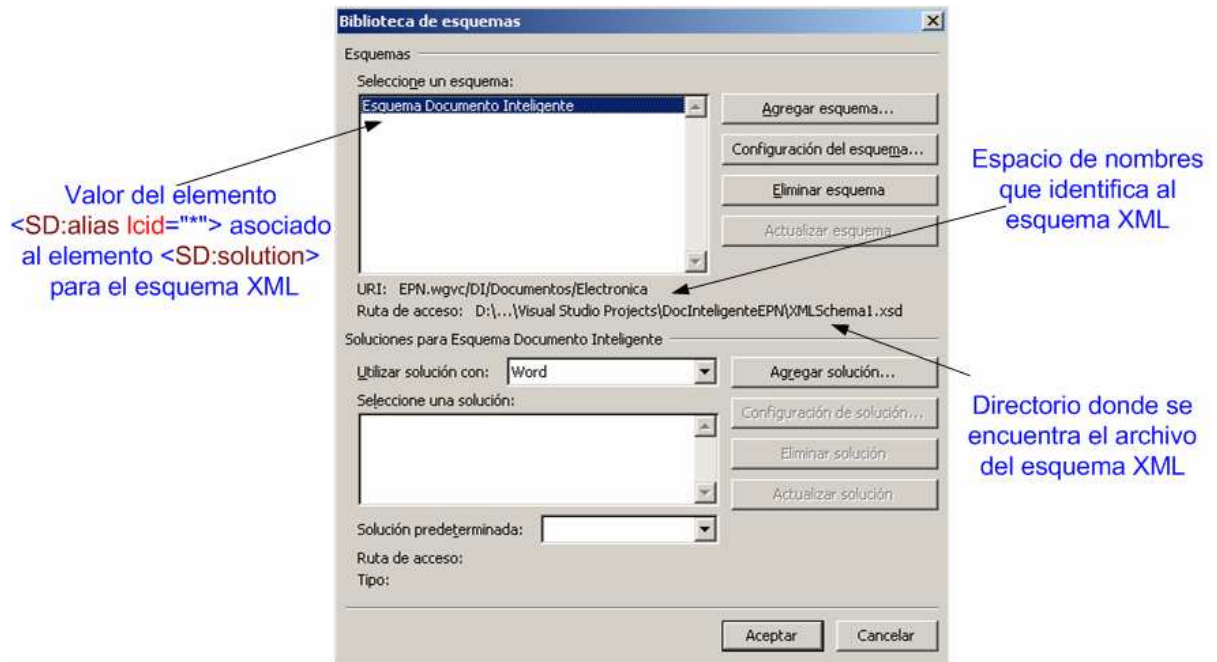


Ilustración 3-52 Cuadro de Dialogo Biblioteca de Esquemas.

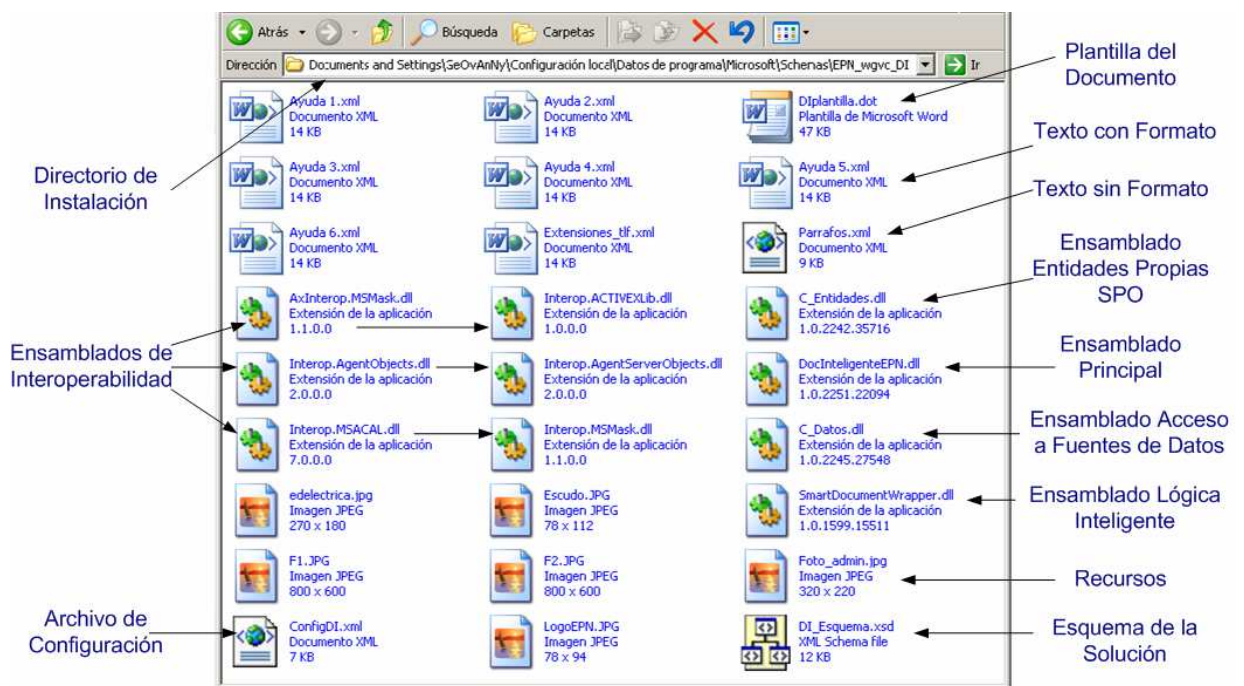


Ilustración 3-53 Archivos que componen el Documento Inteligente

En la Ilustración 3-53, los únicos archivos que requieren una explicación adicional son: C_Datos.dll y C_Entidades.dll; la presencia de estos archivos está justificada con el diseño de las capas lógicas de la SPO, en donde se puede ver claramente que las secciones de código (paquetes): “Entidades Propias de la SPO” (capa

Lógica de Negocio) y “Acceso a Fuentes de Datos” (capa Acceso a Datos y Servicios), tienen funciones comunes para los diferentes subsistemas de la SPO.

Por esta razón, el código (clases y sus métodos) de estos dos paquetes, fue implementado en los dos ensamblados mencionados, los mismos que también serán utilizados por el formulario de InfoPath y el Servicio Web XML según sus necesidades.

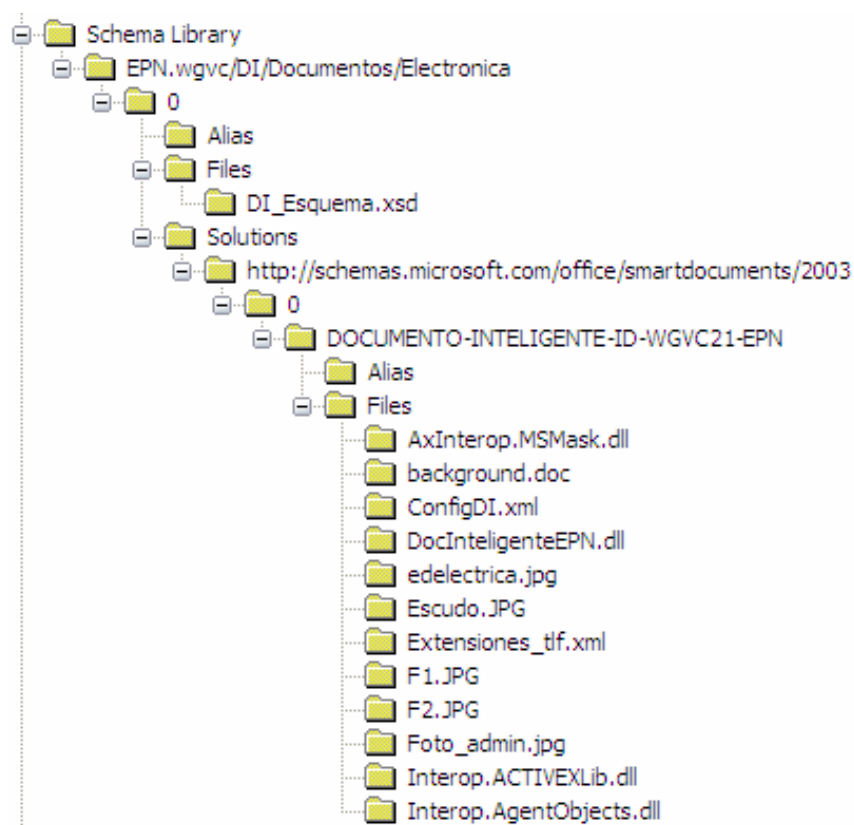


Ilustración 3-54 Algunas de las entradas del Registro de Windows asociadas con el Documento Inteligente.

3.3.3 CONSIDERACIONES PARA EL FORMULARIO DE INFOPATH

Existen varias formas de implementar un formulario de InfoPath, esta Sección se dedicará exclusivamente a explicar la metodología utilizada en este proyecto; para mayor información de todos los métodos y niveles de seguridad existentes, se pueden consultar las fuentes [Ref. 32 y 33] de donde fue tomada mucha de la información que se proporciona a continuación.

Para la implementación del Formulario de InfoPath de la SPO, hay que considerar que de manera predeterminada todos los formularios de InfoPath tienen restringidos los permisos para acceder a los recursos del sistema.

En este caso, al estar utilizando una conexión a la base de datos SPODB, es necesario que la plantilla del formulario tenga plena confianza, para que a partir de ella se puedan generar los denominados *Fully Trusted Forms* o Formularios de Plena Confianza.

En general, lograr que un formulario de InfoPath sea de plena confianza es un proceso un tanto complejo que requiere de bastante prolijidad, ya que olvidarse de alguno de los pasos causaría que el formulario no funcione. Una guía que permita realizar este proceso manualmente se puede consultar en las Referencias indicadas al inicio de esta Sección.

Para la plantilla del Formulario de InfoPath de la SPO, se utilizó una herramienta denominada RegForm.exe⁸⁰ que permite registrar la misma como una plantilla de plena confianza, y con lo que inmediatamente se concederá el conjunto de permisos *FullTrust* al código del formulario [Ref. 57].

Las tareas en las que ayuda esta herramienta son las siguientes [Ref. 33]:

- Crea automáticamente un respaldo de la plantilla del formulario.
- Realiza todos los cambios necesarios en los archivos de definición del formulario (**.xsf**) y la plantilla XML (**.xml**) [ver Tabla 2-2], estos cambios consisten básicamente en agregar algunos atributos o modificar el valor de otros existentes dentro de ciertos elementos XML definidos en estos dos archivos.
- Actualiza el número de la versión de la plantilla del formulario.
- Empaqueta los archivos de la plantilla del formulario en el formato **.xsn** (archivo CAB).
- Crea una programa de instalación

⁸⁰ La herramienta RegForm.exe viene incluida con el SDK de InfoPath [Ref. 33].

Lo más importante a considerar para hacer que la plantilla del formulario sea de plena confianza es que se debe definir un nombre URN, fácil de recordar e identificar. Este nombre está definido por el atributo 'name' del elemento **<xDocumentClass>** en el archivo **.xsf**.

En este caso, el nombre URN utilizado es: "urn:FormularioInfoPath:EPN" el mismo que debe ser proporcionado como parámetro de ejecución para la herramienta RegForm.exe, tal como se muestra a continuación:

RegForm /U urn:FormularioInfoPath:EPN /T Yes /MSI D:\<directorio donde se encuentra la plantilla empaquetada(.xsn)>

La utilidad de cada uno de los parámetros se describe a continuación:

- El parámetro **U** es el que indica el nombre URN que identifica a la plantilla.
- El parámetro **T** con el valor "Yes" es el que indica que se debe otorgar plena confianza a la plantilla.
- El parámetro **MSI** indica que el archivo instalador que se generará será un **.msi**; de forma predeterminada esta herramienta genera un *script* (archivo **.js**) que permite instalar la plantilla en un computador.
- El último parámetro es la dirección donde se encuentra la plantilla empaquetada del formulario; esta herramienta desempaqueta los archivos, los modifica y los vuelve a empaquetar, por esta razón puede sacar un respaldo de la plantilla original.

El instalador generado por esta herramienta fue modificado con la ayuda del Visual Studio .NET, para cambiar el idioma (el instalador es generado en inglés), agregar una nueva pantalla inicial y poner especificaciones que estén más acorde con lo que es el Formulario de InfoPath de la SPO. En la Ilustración 3-55, se puede observar la pantalla inicial del instalador.

La utilización del instalador no es compleja, y de forma predeterminada instala la nueva plantilla en el directorio <Drive>:\Archivos de programa\InfoPath Form Templates\ que es el predeterminado para almacenar todas las plantillas de InfoPath.

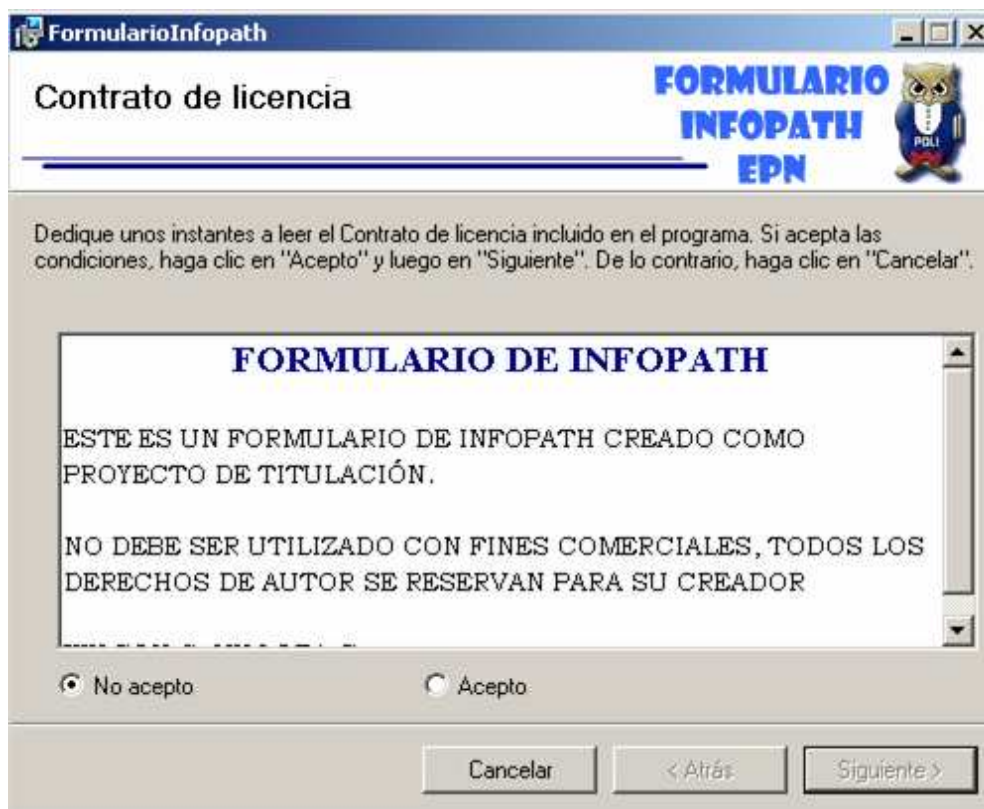


Ilustración 3-55 Pantalla inicial del instalador para el Formulario de InfoPath.

Adicionalmente, con este instalador se agregan varias entradas en el Registro de Windows, y las más importantes se pueden ver en la Ilustración 3-56. Luego de la instalación, el Formulario podrá ser utilizado sin problemas y el URN utilizado, será el identificador del mismo, tal como se muestra en la Ilustración 3-57.

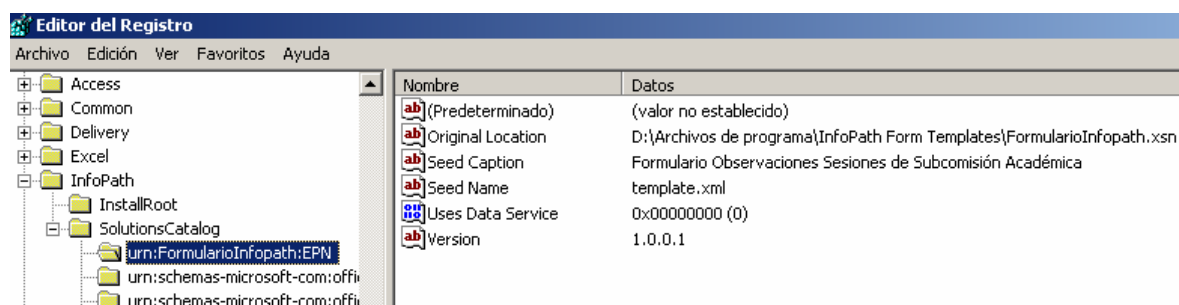


Ilustración 3-56 Entradas del Registro de Windows para el Formulario de InfoPath.

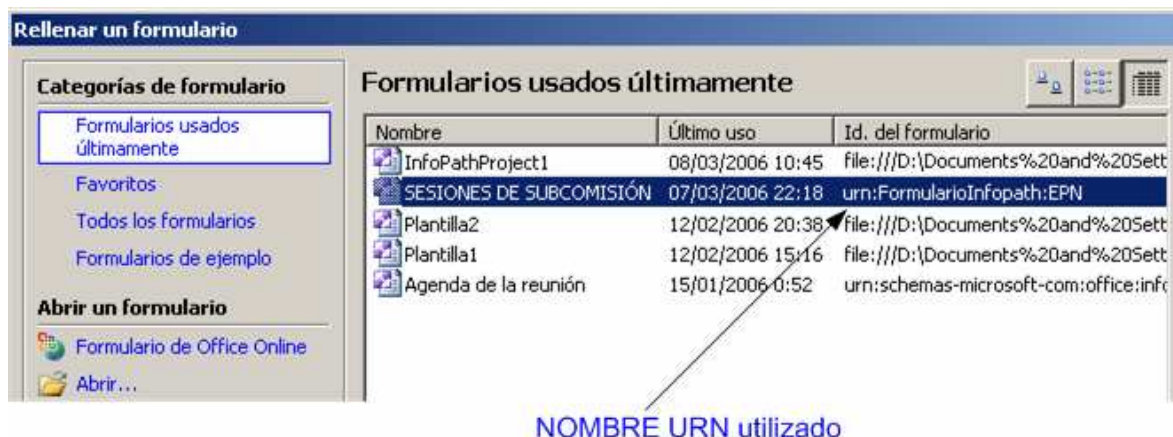


Ilustración 3-57 Formulario de InfoPath luego de ser instalado.

3.3.4 REQUERIMIENTOS DE SOFTWARE

Considerando los tres subsistemas que contiene la SPO (Documento Inteligente, Formulario de InfoPath y Servicio Web), las aplicaciones de base que deberá tener instalado el computador donde se alojará la solución son las siguientes:

- **Sistema Operativo Windows 2000 Profesional, o XP Profesional:** También funcionará sin problemas en sistemas operativos de servidor (ej. Windows 2003 Server), para el caso de Windows XP no se puede utilizar la versión Windows XP HOME, ya que tiene limitantes para la implementación de los Servicios Web.
- **Framework .NET V1.1:** Es necesario revisar que la versión del *Framework* .NET instalada en el sistema operativo sea la correcta para no tener problemas con la ejecución del código administrado.
- **Servicio de Microsoft Internet Information Server (IIS) V5.1 o superior:** Requerido para la implementación del Servicio Web local.
- **Microsoft SQL Server 2000:** Se debe instalar una versión compatible con el sistema operativo utilizado.
- **Microsoft Office Word 2003:** Requerido para el funcionamiento del Documento Inteligente, en caso de ya estar instalado el Office es muy importante revisar la versión, ya que el paquete comercial estándar más básico del Office 2003 no permite el trabajo con esquemas XML.
- **Microsoft Office Outlook 2003:** Requerido para el envío de los documentos por correo electrónico; esta aplicación debe tener la configuración básica inicial que permita su funcionamiento.

- **Ensamblados de Interoperabilidad Primaria:** Es necesario revisar si se encuentran instalados todos los ensamblados de Interoperabilidad Primaria necesarios dentro del caché global de ensamblados.
- **Microsoft Office InfoPath 2003:** Necesario para ejecutar el Formulario de InfoPath.
- **Acceso a Internet:** Necesario para el envío de documentos, publicación del Servicio Web local y consumo de Servicios Web Remotos, en caso de que la SPO funcione en un ambiente distribuido por Internet.

Si bien es cierto que al analizar la lista de requerimientos de la SPO, resultaría ser un sistema muy costoso para una implementación comercial en la que se tengan que comprar todas las licencias del software base, hay que considerar que esta solución se desarrollo sin escatimar recursos económicos, con el objetivo principal de probar las tecnologías más actuales para la personalización de las aplicaciones de Office.

Por otro lado, si se considera el caso hipotético de que la SPO fuera implementada para una empresa que ya tiene adquiridas todas las licencias del software base, esta sería una excelente manera de devengar de mejor manera los recursos económicos invertidos.

Sin embargo, si esta fuera una implementación comercial, se podrían prescindir de muchos de los requerimientos mencionados; como por ejemplo, el InfoPath, el IIS para los Servicios Web (se utilizaría otra alternativa de programación distribuida), el acceso a Internet (deshabilitaría las opciones de Outlook), o el mismo SQL podría ser reemplazado sin problema por Access 2003, con tan solo redefinir algunas de las clases de la SPO.

3.4 PRUEBAS

Existen diferentes tipos de pruebas que pueden ser realizadas a un sistema, lo cual no implica que para probar el correcto funcionamiento se deba recurrir a todas ellas; hay que tener en mente que la realización de pruebas implica un

costo que debe ser considerado en el momento de establecer una relación costo beneficio, y considerando que el gasto realizado en pruebas se ve reflejado en la calidad del sistema hay que tener un buen balance.

Para probar la SPO se contempló la realización de dos tipos de pruebas: las pruebas de Sistema y las Pruebas de Aceptación, la primera se realiza para hacer una comprobación del cumplimiento de los requerimientos, mientras que la segunda para comprobar la interacción del usuario con el sistema.

3.4.1 PRUEBAS DE SISTEMA

Las pruebas de sistema de la SPO se realizan revisando si cada uno de los Casos de Uso mostrados en la Ilustración 3-2, que son los que en realidad representan toda la funcionalidad que la solución debería ofrecer, se cumplen.

Por esta razón, en esta Sección se volverá a hacer mención de los casos de uso, pero ahora relacionados con los elementos que componen la Arquitectura de la SPO [Ilustración 3-5], y presentando los resultados obtenidos de las pruebas asociadas a cada uno de ellos.

3.4.1.1 Administración de los PPTs

La Administración de los PPTs es una función que engloba varias otras funciones, las mismas que podrán ser ejecutadas desde el Documento Inteligente, y como un adicional si las secretarías así lo desean, para agregar las observaciones realizadas en una sesión de Subcomisión Académica podrán emplear el Formulario de InfoPath.

Todas las funciones de la SPO utilizarán una conexión directa a la Base de Datos Local con sus respectivos permisos, por esta razón a menos que explícitamente se indique algo diferente, su utilización deberá estar sobrentendida.

PRUEBA:	Sincronización con la Información del SAE
DATOS DE ENTRADA:	<ul style="list-style-type: none"> • Nombre del Servidor de la Base de Datos • Nombre de la Base de Datos (SPODB) • Nombre de Usuario (de la Base SAE) • Contraseña (del usuario del SAE)
PROCEDIMIENTO DE PRUEBA:	Presionar el botón SINCRONIZAR (Ilustración 3-59), ingresar los datos requeridos en la ventana correspondiente, presionar el botón aceptar, repetir el procedimiento con datos erróneos y comprobar si el sistema valida toda la información ingresada.
RESULTADO OBTENIDO:	<ul style="list-style-type: none"> • En caso de ingresar datos correctos la sincronización se realiza sin problemas y se muestra la cantidad de registros leídos y copiados. • En caso de ingresar datos erróneos se presenta un cuadro de mensaje indicando que los datos no son válidos y se solicita al usuario el nuevo ingreso de información.

Tabla 3-33 Prueba para la Sincronización con la Información del SAE.

La función de Sincronización con la Información del SAE estará disponible a través de un formulario de Windows denominado **Formulario de Configuración** el mismo que aparecerá al presionar el botón de la barra de herramientas personalizada del Documento Inteligente, que se muestra en la Ilustración 3-58.



Ilustración 3-58 Barra de herramientas personalizada Documento Inteligente.



Ilustración 3-59 “Formulario de Configuración” opción “Sincronizar Datos SAE”.

En el **Formulario de Configuración**, existirá una pestaña (ficha) denominada **Sincronizar Datos SAE**, la misma que contendrá un solo botón, tal como se muestra en la Ilustración 3-59; su ejecución resulta en el apareamiento del cuadro de diálogo mostrado en la Ilustración 3-60.

Ilustración 3-60 Cuadro de Diálogo de Datos para Sincronización.

Un proceso bastante similar al realizado en la función de sincronización, se realiza en el proyecto de titulación de la Referencia 59, en donde se crea una base de datos denominada SAMI; ahí se realiza la sincronización de mucha más información que la requerida para la SPODB; esta Referencia podría ser consultada para conocer mas detalles de la estructura de la base de datos SAE.

PRUEBA:	Ingreso de Nuevos PPTs
DATOS DE ENTRADA:	Información nueva sobre un Plan de Titulación.
PROCEDIMIENTO DE PRUEBA:	<ul style="list-style-type: none"> • Ingresar en cada campo presentado en el formulario (Ilustración 3-62), la información con datos erróneos y luego correctos. • Ingresar datos en cada campo presentado en el formulario y presionar el botón Grabar. • Ingresar datos en cada campo presentado en el formulario y presionar el botón Cancelar. • Ingresar datos en cada campo presentado en el formulario y presionar el botón Salir.
RESULTADO OBTENIDO:	<ul style="list-style-type: none"> • Se realiza una validación instantánea de la información; si el usuario ingresa un dato equivocado en cualquiera de los campos el sistema muestra una alerta junto al error. • Al presionar el botón Grabar, se realiza una validación de toda la información y se presenta un listado de todos los errores encontrados, en caso de que la información esté correcta el sistema agrega el nuevo registro. • Al presionar el botón Cancelar, se establece el contenido de los controles al estado inicial y no se realiza ningún ingreso. • Al presionar el botón Salir, se verifica si se ha ingresado alguna información, de ser así, el sistema preguntará si se desea guardar dicha información, si la respuesta es afirmativa se inicial el proceso de grabado, caso contrario la información no se almacena y el formulario se cierra.

Tabla 3-34 Prueba de Ingreso de Nuevos PPTs.

El botón de la barra de herramientas personalizada que permitirá presentar el formulario para agregar un PPT, tendrá el nombre de **Agregar Proyecto** y se puede apreciar en la Ilustración 3-61, al presionar dicho botón aparecerá el cuadro de diálogo mostrado en la Ilustración 3-62 .

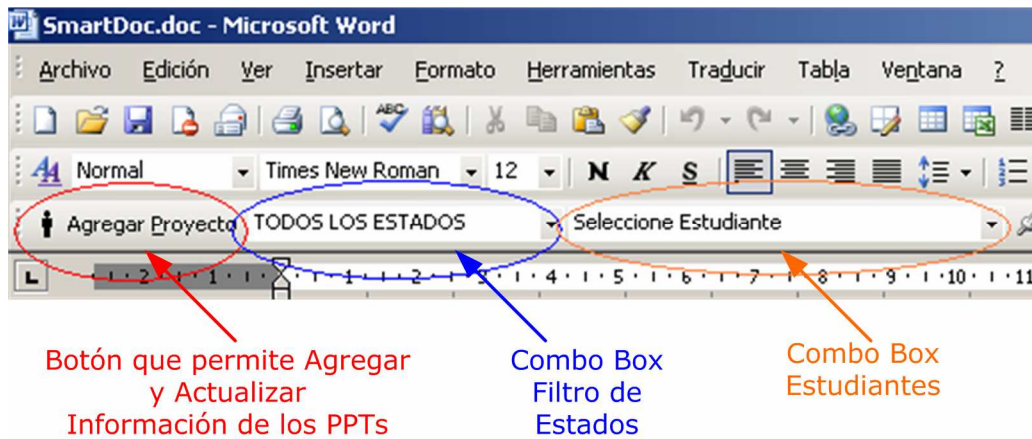


Ilustración 3-61 Barra de herramientas personalizada Documento Inteligente.

Agregar Proyecto

PLAN DE PROYECTO

Agregar Nuevo Plan Actualizar Existente

PROPUESTO POR

Un estudiante Dos Estudiantes

DAQUI SOLANO OSWALDO FERNANI ok

DAQUI SOLANO OSWALDO FERNANI !

DIRECTOR Y CODIRECTOR DEL PROYECTO

Proyecto con Codirector

ALDAZ OSWALDO ok

ALDAZ OSWALDO !

TEMA O TITULO DEL PROYECTO

El nombre del director no puede ser igual al del codirector

La Fecha de Presentación no puede ser mayor que la actual

AREAS TECNICAS DEL TEMA

Buscar: redes

REDES DE AREA EXTENDIDA

REDES DE AREA LOCAL

REDES DE ÁREA LOCAL INALAMBRIC

BASES DE DATOS

El nombre de los estudiantes no puede ser igual

FECHA PRESENTACION

jueves , 09 de marzo de 2006 !

Cancelar Grabar Salir

Borra el contenido actual de los controles

Almacena la información en la SPODB

Cierra el formulario

Ilustración 3-62 Validación inmediata de información.

Como resultado de las pruebas se mencionaron dos tipo de validación:

- **Validación instantánea:** comprueba que la información ingresada cumpla con los siguientes requerimientos elementales: el nombre del director y del codirector del proyecto deben ser diferentes; en caso de ser un proyecto de dos estudiante, los nombres de los mismos también deben ser distintos; la fecha de presentación del plan de proyecto debe ser menor o igual que la fecha en la que se está creando el registro. Los errores de validación instantánea se presentarán inmediatamente luego de ingresado un valor, y serán identificados por un icono junto al control en el que se presenta el inconveniente, como se puede apreciar en el Ilustración 3-62.
- **Validación General:** Según el diseño de la SPODB, existen varios datos que no pueden ser nulos, y esto es lo que se comprueba en una validación general, que se realizará en el momento que se desee almacenar la información en la SPODB, en este caso los errores se presentarán en un mensaje de error como el que se presenta en la Ilustración 3-63.

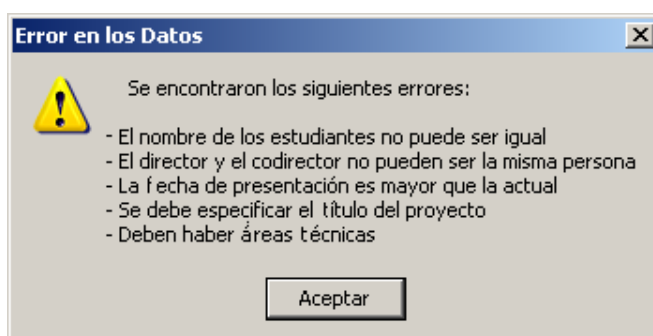


Ilustración 3-63 Mensaje de error en los datos.

También está contemplada la eliminación de los planes de proyecto, en este caso se presentará un cuadro de diálogo de confirmación, como el que se muestra en la Ilustración 3-64.

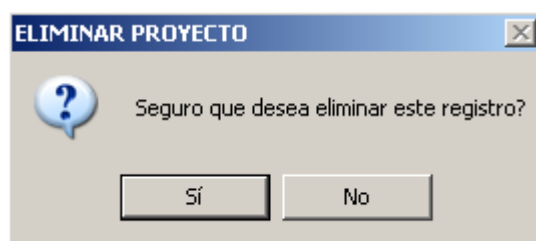


Ilustración 3-64 Confirmación de eliminación.

PRUEBA:	Actualización de la Información de los PPTs
DATOS DE ENTRADA:	<ul style="list-style-type: none"> • Nombre del estudiante del cual se desea actualizar la información. • Información actualizada sobre un Plan de Proyecto de Titulación.
PROCEDIMIENTO DE PRUEBA:	<ul style="list-style-type: none"> • Seleccionar el nombre del estudiante y verificar si se presenta la información asociada al mismo y es correcta. • Realizar las modificaciones de los datos que sean necesarias y presionar el botón Actualizar (Ilustración 3-65). • Seleccionar el nombre del estudiante cuyo registro se desee quitar y presionar el botón Eliminar (Ilustración 3-65).
RESULTADO OBTENIDO:	<ul style="list-style-type: none"> • Al seleccionar el estudiante: la información del PPT asociado se carga automáticamente, completa y sin errores. • Al realizar las modificaciones, el sistema valida la nueva información y la almacena. • Cuando se desea eliminar un registro el sistema presenta un mensaje de confirmación, si la respuesta es afirmativa el registro se elimina, caso contrario se mantiene.

Tabla 3-35 Prueba de Actualización de la Información de los PPTs.

Para actualizar la información de un PPT se accederá utilizando el mismo botón de la Barra de Herramientas que se utiliza para el ingreso de un nuevo PPT, el formulario de Windows que se utiliza es el mismo solo que en este caso con la opción **Actualizar Existente** que se puede ver tanto en la Ilustración 3-62 (ingreso de información) como en la Ilustración 3-65 (actualización de información).

Para el proceso de actualización de información se aplican las mismas validaciones que para el ingreso de un nuevo plan, y a pesar de que el formulario es el mismo, cambian algunos controles de la interfaz de usuario como se puede observar en la Ilustración 3-65.

El título del formulario cambia al seleccionar la opción Actualizar Existente

Actualizar Proyecto

PLAN DE PROYECTO

Agregar Nuevo Plan Actualizar Existente

PROPUESTO POR

Un estudiante Dos Estudiantes

MOYA LEIMBERG FERNANDO ANDRE

REA GUALOTO VICTOR HUGO

DIRECTOR Y CODIRECTOR DEL PROYECTO

Proyecto con Codirector

SINCHE MAITA SORAYA LUCIA

AREAS TECNICAS DEL TEMA

Buscar:

REDES DE COMPUTADORAS

REDES E INTRANET

REDES TCP/IP

REDES DE ÁREA LOCAL INALAMBRICAS
REDES DE COMPUTADORAS
REDES E INTRANET
REDES TCP/IP
SEGURIDAD EN REDES

FECHA PRESENTACION

viernes , 03 de junio de 2005

TEMA O TITULO DEL PROYECTO

DISEÑO Y SIMULACIÓN DE LA RED DE ACCESO INALÁMBRICO DE BANDA ANCHA PARA PANCHONET S.A. EN LA ZONA DE LOS VALLES CIRCUNDANTES A LA CIUDAD DE QUITO UTILIZANDO LAS TECNOLOGÍAS WIMAX (IEEE 802.16) Y MOBILEFI (802.20)

Eliminar **Actualizar** **Salir**

Elimina el registro de la SPODB

Actualiza la información en la SPODB

Cierra el formulario

Ilustración 3-65 Actualización de información.

3.4.1.2 Administración del Estado de los PPTs Y PTs

Los dos casos de uso asociados con la Administración del Estado de los PPTs y PTs: 1) Identificación del estado actual y 2) Modificación del estado, tienen que regirse a las condiciones (datos requeridos) que se presentan en el diagrama de estados presentado en la Ilustración 3-1. Esto quiere decir, que para poder ingresar ciertos datos asociados a un proyecto, éste primero tiene que encontrarse en un determinado estado.

Por esta razón las pruebas que se efectuaron para verificar el correcto funcionamiento de este caso de uso, también están en íntima relación con el cumplimiento de las condiciones especificadas en el diagrama de estados; los resultados obtenidos se presentan en las Tabla 3-36 y 3-37.

PRUEBA:	Identificación del estado actual
DATOS DE ENTRADA:	<ul style="list-style-type: none"> • Nombre del estudiante del cual se desea conocer el estado en el cual se encuentra el proyecto.
PROCEDIMIENTO DE PRUEBA:	<ul style="list-style-type: none"> • Seleccionar el nombre del estudiante en la Barra de Herramientas personalizada (ver Ilustración 3-66) y verificar si se presenta la información del estado del proyecto en el Panel de Tareas de Acción y si es correcta. • Seleccionar un estado de proyecto en la barra de Herramientas Personalizada y verificar si en la lista de estudiantes se presentan únicamente aquellos que tengan proyectos en el estado seleccionado.
RESULTADO OBTENIDO:	<ul style="list-style-type: none"> • Al seleccionar el estudiante en la Barra de Herramientas Personalizada: se presenta en el Panel de Tareas de Acción el estado en el cual se encuentra el proyecto junto con la información general, en el área de despliegue de información. • Al seleccionar un estado en la Barra de Herramientas Personalizada, el sistema presenta en la lista desplegable de estudiantes únicamente aquellos que poseen un proyecto en el estado seleccionado. • Al seleccionar como estado la opción "Todos los estados" en la Barra de Herramientas Personalizada, en la lista de estudiantes se presentan todos los estudiantes ordenados alfabéticamente, que ya han presentado un proyecto sin importar el estado en el que se encuentre el mismo.

Tabla 3-36 Prueba de identificación del Estado Actual de un Proyecto.

Para entender mejor la forma en la que se presentan los estados y los estudiantes en la Barra de Herramientas Personalizada, se puede observar la Ilustración 3-61 así como también la Ilustración 3-66 en donde se pueden apreciar los dos controles del tipo *Combo Box* destinados para éste propósito.

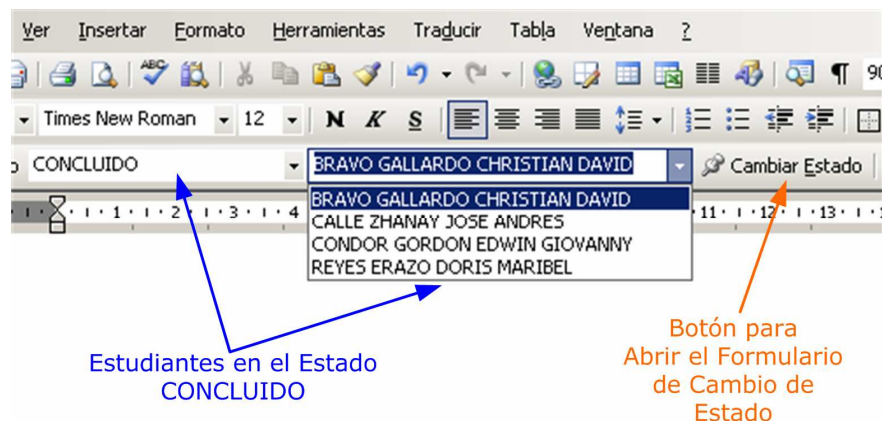


Ilustración 3-66 Barra de herramienta personalizada Documento Inteligente.

PRUEBA:	Modificación del Estado
DATOS DE ENTRADA:	<ul style="list-style-type: none"> • Nombre del estudiante del cual se desea modificar el estado en el cual se encuentra su proyecto. • Datos requeridos según el estado actual y el estado al que se desee cambiar.
PROCEDIMIENTO DE PRUEBA:	<ul style="list-style-type: none"> • Seleccionar el nombre del estudiante cuyo proyecto se desea cambiar de estado, verificar si los posibles estados y los datos solicitados están en concordancia con el diagrama de estados (Ilustración 3-1); ingresar los datos requeridos y presionar el botón Cambiar. • Realizar las pruebas respectivas con los botones Cancelar y Salir.
RESULTADO OBTENIDO:	<ul style="list-style-type: none"> • Al seleccionar el estudiante dentro del formulario de cambio de estado (Ilustración 3-67) se presenta el estado actual del proyecto, se muestran las posibles opciones para realizar el cambio y los controles para ingresar los datos que requiere la transición. • Los datos requeridos para cada cambio de estado son validados, con el mismo esquema de validación instantánea y grupal, y si no existe ningún inconveniente se realiza el cambio y se almacena la nueva información.

Tabla 3-37 Prueba de la Modificación del Estado de un Proyecto.

Cambio de Estados

CAMBIO DEL ESTADO DE LOS PROYECTOS

Estado Actual
 EN APROBACIÓN

Estudiantes
 ASTUDILLO MENDOZA ISABELA DEL ROCIO

Estados Posibles

NEGADO/ABANDONADO

APROBADO/EN DESARROLLO

DELEGADO A COORDINADOR

Datos Requeridos

Fecha de la Aprobación
 Quito DM, miércoles, 08 de marzo de 2006

SISTEMA DE INFORMACIÓN PARA LA ADMINISTRACIÓN DEL ESPECTRO RADIOELÉCTRICO DE ENLACES ENTRE RADIOBASES DE LOS SISTEMAS DE TELEFONÍA MÓVIL CELULAR UTILIZANDO SISTEMAS DE INFORMACIÓN GEOGRÁFICA (GIS)

PRESENTADO POR:
 ISABELA DEL ROCIO ASTUDILLO MENDOZA

Observaciones **Cancelar** **Cambiar** **Salir**

Ilustración 3-67 Cuadro de diálogo para cambio de Estado.

El formulario de Cambio de Estado se presenta en la Ilustración 3-67, en donde se puede apreciar un botón denominado **Observaciones** y que no ha sido mencionado con anterioridad.

Este botón está con la finalidad de permitir agregar como dato, las observaciones que se realizan a los Planes de Proyectos de Titulación durante las sesiones de Subcomisión Académica, no es indispensable para realizar un cambio de estado, pero si está en relación con el estado del proyecto, razón por la cual se decidió ubicarlo dentro de este formulario.

3.4.1.3 Elaboración de Cartas y Memorandos

Ésta es la función principal que cumplirá el Documento Inteligente y se podría decir que la SPO en general. Y aquí es donde se realizaron la mayor cantidad de pruebas y donde también se obtuvo la mayor cantidad de errores.

Si bien es cierto que hasta el momento en todos los resultados presentados no se ha mencionado la existencia de fallas, esto no quiere decir el sistema pasó todas las pruebas y justamente en esta Sección se podrán encontrar algunos errores, con sus respectivas causas y las soluciones encontradas.

PRUEBA:	Generación automática de documentos
DATOS DE ENTRADA:	Nombre del estudiante del cual se desea generar un documento específico.
PROCEDIMIENTO DE PRUEBA:	<ul style="list-style-type: none"> • Seleccionar el nombre del estudiante de la Barra de Herramientas Personalizada, verificar si el Panel de Tareas de Acción muestra la información relacionada con el proyecto del estudiante seleccionado, verificar si las opciones de documentos posibles de generar coinciden con el estado actual del proyecto, y con los señalados en el diagrama de estado. • Una vez generado el documento se debe verificar si el texto y el formato corresponden al documento, y que la concatenación de información se encuentre bien establecida. • Se debe verificar que todos los documentos estén bien formados, lo que quiere decir que no existan errores con el sexo de las personas, que se coloque el calificativo correcto a un profesor (Ing., Ph.D., etc.) que el documento sea entendible a su totalidad (palabras completas, signos de puntuación espaciado, ortografía, etc.)
RESULTADO OBTENIDO:	<ul style="list-style-type: none"> • Al seleccionar el estudiante, en el Panel de Tareas de Acción se carga la información del Proyecto de Titulación asociado con el estudiante; si el estudiante ha presentado más de un plan de proyecto, se despliega un cuadro de diálogo presentando todos los proyectos relacionados con el estudiante, para que el usuario seleccione con el que desea trabajar. • En el Panel de Tareas de Acción se presenta el estado correcto de los proyectos, y las opciones para generación de documentos acorde al estado y siguiendo el diagrama de estados. • La generación de los documentos fue satisfactoria, desde el punto de vista de presentar el documento correcto y con la información correcta, sin embargo se presentaron problemas en lo que se refiere al texto y formato.

	<ul style="list-style-type: none"> • Se detectaron problemas con ciertos artículos (el, la, los, etc.), con los calificativos de los profesores, con la escritura de ciertas calificaciones, y con el espaciado (concatenación) entre palabras.
CAUSAS DE LOS ERRORES:	<ul style="list-style-type: none"> • Existe mucha información errónea en la base de datos SAE, que no fue considerada en el momento del desarrollo, y recién cuando fueron facilitadas las copias de las bases de datos con información completa y real saltaron a la vista varios de los mencionados errores. • No se consideró que al editar un archivo XML en Word se eliminan automáticamente los espacios iniciales y finales del texto que se encuentra entre las etiquetas, ya que esto no ocurre en otros editores de XML, esto causó algunos problemas con el espacio entre ciertas palabras.
SOLUCIONES:	<ul style="list-style-type: none"> • Se hicieron las modificaciones necesarias en el código para que se consideren todas las posibilidades de error que pudiera tener la información del SAE. • Se realizó una modificación del código para que el espacio entre palabras sea realizado mediante programación y no por escritura y concatenación directa.

Tabla 3-38 Pruebas para la Generación Automática de Documentos.

PRUEBA:	Visualización de información sensitiva al contexto
DATOS DE ENTRADA:	No requiere
PROCEDIMIENTO DE PRUEBA:	<ul style="list-style-type: none"> • En un documento generado, mover el cursor sobre el texto para verificar la ayuda al contexto que debe aparecer en el Panel de Taras de Acción.
RESULTADO OBTENIDO:	<ul style="list-style-type: none"> • El sistema reacciona bien a la sensibilidad al contexto, cuando el usuario se ubica en una sección específica del documento que debe presentar ayuda o información, el Panel de Tareas despliega la información sin presentar mayor inconveniente. • Se detecto la presencia de un error del sistema cuando se hacía una selección de

	<p>varias líneas en un documento, si dentro de esa selección se encuentran dos secciones del documento que tienen ayuda en el panel de tareas, el sistema no sabe en que sitio exacto está el usuario y que información desplegar y se produce una excepción.</p>
CAUSAS DE LOS ERRORES:	<ul style="list-style-type: none"> • No se consideró en el momento de programar la posibilidad de que un usuario seleccione dos lugares a la vez, que es lo que puede ocurrir cuando se realiza una selección de todo el documento o de un segmento grande, por lo que no se puede determinar una ubicación exacta.
SOLUCIONES:	<ul style="list-style-type: none"> • Se incrementó el chequeo de errores para que la excepción no sea visible para el usuario, este es una excepción que siempre se generará cuando el usuario realice una selección múltiple, sin embargo lo que si se puede evitar es que cause problemas para el usuario, haciendo que el sistema mantenga su estado cuando la excepción ocurra.

Tabla 3-39 Prueba para la Visualización de Información Sensitiva al Contexto.

PRUEBA:	Envío Directo de Documentos Vía e-mail
DATOS DE ENTRADA:	<p>Dirección de correo electrónico Asunto del correo Introducción (opcional)</p>
PROCEDIMIENTO DE PRUEBA:	<ul style="list-style-type: none"> • En un documento generado, mover el cursor sobre el nombre de alguna persona sea profesor o estudiante, presionar el botón a manera de hipervínculo con el texto "Enviar a" del Panel de Taras de Acción, se abrirá una ventana solicitando los Datos de Entrada, que ya el sistema debe haber generado, y probar el envío presionando el botón enviar.
RESULTADO OBTENIDO:	<ul style="list-style-type: none"> • Los documentos son enviados satisfactoriamente al destinatario del correo electrónico, el título y la introducción del mensaje corresponde correctamente al contenido. • Se detecto la presencia de un error del

	<p>sistema cuando el Microsoft Outlook no había sido configurado con anterioridad a la utilización de esta función.</p> <ul style="list-style-type: none"> • Cuando se realiza un envío sin tener conectado el servicio de Internet, sin importar que se cierre el Documento Inteligente, el servicio de Outlook permanece ejecutándose en espera de que exista una conexión disponible.
CAUSAS DE LOS ERRORES:	<ul style="list-style-type: none"> • Al configurar el Outlook se activa una especie de motor de correo electrónico en el Word, el mismo que es utilizado para el envío de los mensajes, al no encontrar este servicio, e intentar utilizarlo se produce una excepción.
SOLUCIONES:	<ul style="list-style-type: none"> • Al igual que con caso anterior se incrementó el chequeo de errores para que la excepción no sea visible para el usuario, y que impida utilizar el servicio mientras éste no se encuentre bien configurado.

Tabla 3-40 Pruebas para el Envío directo vía e-mail.

PRUEBA:	Edición del Contexto de los Documentos
DATOS DE ENTRADA:	No requiere datos específicos
PROCEDIMIENTO DE PRUEBA:	<ul style="list-style-type: none"> • Configurar los permisos de edición en los documentos a ser generados, probar si las configuraciones son aplicadas correctamente, y si no existe ninguna manera de alterar el texto y el formato de los párrafos que han sido bloqueados. • Si el documento no tiene configurado el bloqueo de párrafos, se debe poder editar sin ningún problema tanto el texto como el formato.
RESULTADO OBTENIDO:	<ul style="list-style-type: none"> • Los permisos de edición se aplican correctamente; cuando un documento está bloqueado aparecen resaltadas las secciones que puede editar el usuario, y en las que no, se puede apreciar un mensaje en la barra de estado indicando que el usuario no tiene permiso de

	<p>escritura.</p> <ul style="list-style-type: none"> • Cuando el documento no está protegido, la edición del documento auto generado se puede realizar como si se tratara de un documento de Word común y corriente, cualquier cambio se verá reflejado cuando se lo envíe por mail o se lo imprima, pero no será almacenado, lo que quiere decir que cuando el sistema vuelva a generar el mismo documento, lo hará tal y como lo hizo la primera vez que lo generó. • El documento modificado puede ser almacenado como un archivo .doc o .rtf sin ningún problema en caso de ser requerido, en este caso los cambios si serán mantenidos. • No existe una forma simple de evadir este bloqueo, es imprescindible conocer el nombre de usuario y la contraseña para hacerlo.
--	---

Tabla 3-41 Pruebas para la edición del contexto de los documentos.

PRUEBA:	Edición del Texto Común (repetitivo)
DATOS DE ENTRADA:	No requiere datos específicos
PROCEDIMIENTO DE PRUEBA:	Abrir el archivo XML que contiene el texto común repetitivo, base para la generación de documentos, realizar varios cambios y ver si los mismos se ven reflejados en los documentos que genera el sistema.
RESULTADO OBTENIDO:	El archivo XML se abre sin problema desde el Documento Inteligente, la edición del texto de este archivo no presenta mayor inconveniente, los cambios son almacenados y si se ven reflejados en los documentos generados por el sistema.

Tabla 3-42 Prueba de Edición del Texto Común (repetitivo).

3.4.1.4 Visualización de Reportes

Para acceder a la funcionalidad de los reportes se utiliza un botón en la barra de herramientas personalizada como se muestra en la Ilustración 3-68.

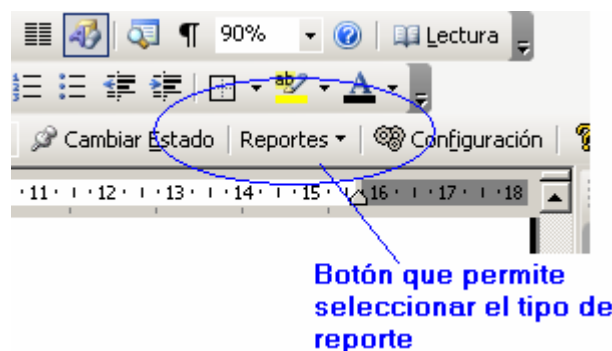


Ilustración 3-68 Barra de herramientas personalizada del Documento Inteligente.

El correcto funcionamiento de los reportes fue probado, montando un sistema idéntico al que se especificó en el diagrama de la Arquitectura de la solución (Ilustración 3-5), lo que implica que se probó la transferencia de información tanto desde una Intranet, como desde el Internet; para ello se utilizó un servidor Web provisional con su respectiva salida a Internet.

PRUEBA:	Utilización de Filtros de Información
DATOS DE ENTRADA:	Estado del proyecto, nombre del director, carrera, etc. Cualquiera de los filtros de información que se desee aplicar al reporte.
PROCEDIMIENTO DE PRUEBA:	Luego de seleccionar la opción reportes de la Barra de Herramientas Personalizada, en el Panel de Tareas de acción ingresar diferentes tipos de datos de entrada, presionar el botón para generar el reporte y ver si la información presentada cumple con el filtro de información.
RESULTADO OBTENIDO:	Los datos para filtrar la información pueden ser ingresados sin ningún problema y la información de los reportes cumple correctamente con los diferentes filtros utilizados.

Tabla 3-43 Prueba para la Utilización de Filtros de Información.

PRUEBA:	Configuración de Direcciones URL
DATOS DE ENTRADA:	Direcciones de los servidores Web donde se encuentran alojados los Servicios Web XML de las diferentes Carreras.
PROCEDIMIENTO DE PRUEBA:	Ingresar las direcciones URL de los servidores y presionar el botón para comprobación de la conexión, cuando el servicio esté subido y cuando esté bajado.
RESULTADO OBTENIDO:	El establecimiento de la conexión con los

	servidores Web se realiza sin presentar problemas, cuando el servidor no está disponible o cuando la dirección del mismo ha sido modificada, se presenta en pantalla un mensaje indicando que la conexión no está disponible, caso contrario se establece el enlace.
--	--

Tabla 3-44 Prueba para la Configuración de Direcciones URL.

3.4.2 PRUEBAS DE ACEPTACIÓN

La Solución Personalizada de Office fue desarrollada teniendo en mente que al estar utilizando como base una interfaz de usuario conocida para los usuarios finales como es el Microsoft Word, debe ser de fácil aprendizaje y utilización, y justamente ese es el propósito de la realización de las pruebas de aceptación, corroborar o desmentir que la personalización de aplicaciones de Office fue una opción acertada para simplificar la tarea de creación de determinadas cartas y memorandos en las secretarías de las Carreras de Electrónica.

CRITERIO DE PRUEBA
Qué tanta predisposición presenta el usuario para utilizar el nuevo sistema
DEFINICIÓN DE LA PRUEBA
En el momento en el que es presentado el sistema al usuario, se tiene en consideración si existe interés por descubrir por sí mismo las nuevas tareas y la funcionalidad del sistema.
RESULTADO
<ul style="list-style-type: none"> • En ninguno de los casos de prueba del Documento Inteligente (DI) el usuarios sintió temor o se sintió sorprendido al ver la interfaz del sistema y comenzaron a utilizar el Documento Inteligente como si se tratara de un nuevo documento de Word cualquiera. • El panel de Tareas de Acción del DI causó curiosidad y los usuarios intentaron descubrir cuál es su funcionalidad. • Algunos usuarios no se percataron de la existencia de la Barra de Herramientas personalizada en el instante mismo en el que vieron por primera vez el Documento Inteligente, siempre fue primero el Panel de Tareas y luego la Barra de Herramientas Personalizada. • Cuando los usuarios tomaban conciencia de la existencia de la Barra de Herramientas Personalizada comenzaban a presionar los botones para descubrir la función que desempeñaban, abrían los Formularios Windows y los cerraban casi inmediatamente. • Muy pocas personas hicieron una pregunta como: “Ahora que hago?” o “Qué hay que hacer?” sin haber presionado por lo menos alguno de los botones de la Barra de Herramientas. • En el caso del Formulario de Infopath, ocurrió lo contrario los usuarios no se mostraron familiares con lo que estaban viendo y tampoco se arriesgaron o se

mostraron muy predispuestos a presionar botones o investigar por si solos el funcionamiento del formulario.

- En síntesis los usuarios presentan una mejor predisposición de aprender la utilización del Documento Inteligente que la utilización del Formulario de Infopath, sin embargo hay que mencionar que esta prueba fue realizada con personas que saben manejar Office y tienen destreza en su utilización.

Tabla 3-45 Prueba de Aceptación número uno.

CRITERIO DE PRUEBA
Qué tan fácil es para el usuario aprender el uso del sistema
DEFINICIÓN DE LA PRUEBA
Para esta prueba es necesario realizar ejercicios sobre las funcionalidades del sistema, para verificar que tanto aprendió el usuario. Se consideraron tres formas de aprendizaje: <ul style="list-style-type: none"> • Realizando una explicación al usuario de que es lo que puede hacer con el Sistema pero no como lo hace (por intuición). • Realizando una capacitación inicial completa al usuario. • Luego de que el usuario ha leído el manual de usuario.
RESULTADO
<p>Intuición:</p> <ul style="list-style-type: none"> • Con este mecanismo de aprendizaje se suscitaron problemas con la Barra de Herramientas Personalizada y con algunos de los Formularios Windows; en un principio los controles para seleccionar el estado y el nombre de los estudiantes en la Barra de Herramientas Personalizada, solo provocaban cambios en el Panel de Tareas, y cuando se abría el formulario para cambiar el estado había que volver a seleccionar al estudiante, eso causaba confusión al usuario, y tuvo que ser modificado. • En la Barra de Herramientas Personalizada el botón de configuración se encuentra al lado derecho del usuario, y todas las personas diestras tendían a utilizar los botones del lado izquierdo, por lo que el botón para abrir el Formulario de Configuración era el último en ser presionado, sin embargo la primera vez que se utiliza el sistema lo primero que tiene que hacer el usuario es configurarlo, razón por la cual el aprendizaje intuitivo se dificultaba; por otro lado para el uso cotidiano y en el que se espera que el formulario de configuración sea el menos utilizado, se puede decir que la ubicación de los botones es correcta. • El formulario de ingreso y actualización de la información de los PPTs y PTs, no causó ningún inconveniente y el usuario pudo manejarlos intuitivamente. • El formulario para el cambio de estados, no pudo ser manejado por las personas que no habían comprendido bien el mecanismo de utilización de estados en los proyectos que propone el nuevo sistema; fue necesario incluir un diagrama de estados en el sistema y hacer una explicación más detallada de este proceso, luego de que el usuario comprendió bien los requerimientos y el porqué de cada uno de los estados, este formulario pudo ser manejado sin ningún inconveniente. • En el caso del formulario de configuración, el usuario realizaba una exploración por todas las fichas disponibles y probaba la funcionalidad de cada una de ellas, el manejo fue intuitivo y la única función que el usuario no comprendía o no podía apreciar con claridad en primera instancia fue la

configuración de permisos para la edición de los documentos.

- En cuanto a la generación de documentos y la utilización de la ayuda al contexto el usuario se sintió muy cómodo en la utilización y no existieron problemas ni trabas para el aprendizaje; el usuario presenta muy buena predisposición para trabajar con el Panel de Tareas.

Capacitación:

- Para aquellos usuarios que fueron capacitados en la utilización del sistema, el aprendizaje y utilización del sistema fue mucho más fácil y rápido que para aquellos que intentaron aprenderlo intuitivamente, los usuarios se mostraban deseosos de probar cada una de las funciones que se les iba enseñando y el proceso de enseñanza-aprendizaje fue fácil y sin complicaciones.
- El único punto en el que los usuarios no sentían confianza para manejar el sistema, fue el de la configuración de las direcciones de los servidores Web, en donde manifestaban muy abiertamente que para realizar esa configuración pedirán ayuda a otra persona que entienda del tema, lógicamente que fue porque se intentó explicar a los usuarios como obtener las direcciones de sus servidores para que configuren el sistema sin ayuda de nadie especializado.

Lectura de Manuales:

- Al proporcionar manuales a los usuarios para que se capaciten en la utilización del sistema, se pudo apreciar que se eliminaron algunos de los problemas del aprendizaje intuitivo, como por ejemplo: al leer el manual, los usuarios lo primero que hacen es configurar el sistema, entienden mucho mejor el proceso de cambio de estados, y saben con exactitud cual es la función que cumple el bloqueo de párrafos y secciones.
- Como conclusión de esta prueba, que es la más importante de todas las pruebas de aceptación del sistema, se puede decir que la SPO no es de aprendizaje cien por ciento intuitivo, el usuario presenta algunos problemas al tratar de aprender por sí sólo, sin embargo cuando se le proporciona una guía o ayuda, ya sea escrita con el manual de usuario o personal con la capacitación, el usuario responde favorablemente y aprende con mucha rapidez, no tiene temor de provocar algún error en el sistema, y principalmente cuando aprende a manejar el Documento Inteligente parece como si estuviera manejando un sistema completamente conocido.

Tabla 3-46 Prueba de Aceptación número dos.

CRITERIO DE PRUEBA
Que tanta productividad tiene el usuario al utilizar el sistema
DEFINICIÓN DE LA PRUEBA
Realizar la comparación de productividad del usuario en el tiempo en el que no disponía del sistema contra la productividad utilizando el sistema.
RESULTADO
<ul style="list-style-type: none"> • En ningún momento la redacción de una carta por el método tradicional fue más rápido que la utilización del sistema. • El aumento de productividad con el sistema es directamente proporcional a la cantidad de estudiantes que tenga la Carrera, en este caso se llegó a determinar que al utilizar el sistema se disminuye hasta en un factor de seis,

el tiempo que se emplea para la redacción de cartas y memorandos, y como mínimo en un factor de cuatro, sin embargo a pesar de que el factor de reducción de tiempo es una constante promedio general para cualquier usuario, al multiplicarlo por una constante de utilización, es fácil determinar que el sistema se ve reflejado con mayor productividad en Telecomunicaciones que en Redes de la información o en Control, es decir que por ejemplo, mientras en Redes se requiere hacer una carta, en Telecomunicaciones se requieren hacer tres; por lo tanto, mientras en Redes el ahorro de tiempo es de cuatro veces, en Telecomunicaciones es de doce.

- Se comprobó que el aumento de productividad también va en relación con la variedad de documentos que se tienen que redactar consecutivamente; es decir que por ejemplo, si una secretaria tiene que hacer una carta de Aprobación de Proyecto, una de Designación del Tribunal Calificador, y un Proveído de Notas, una a continuación de la otra, el sistema hace que el usuario ahorre mucho más tiempo que al redactar cartas del mismo tipo, principalmente por el hecho de que los usuarios no tienen que buscar y abrir plantillas de documento diferentes y todo lo realizan sin abandonar la SPO.

Tabla 3-47 Prueba de Aceptación número tres.

CRITERIO DE PRUEBA
Qué tan fácil es para el usuario recordar la funcionalidad del sistema después de un tiempo.
DEFINICIÓN DE LA PRUEBA
Luego de algunos días de haber presentado el sistema y de que el usuario haya aprendido toda la funcionalidad del mismo, se vuelve a hacer ejercicios sobre la funcionalidad del sistema, para ver si el usuario recuerda lo que aprendió.
RESULTADO
<ul style="list-style-type: none"> • El usuario no presenta ninguna dificultad por recordar la utilización del sistema, no existen una gran cantidad de funciones por lo que el usuario las recuerda con facilidad. • Los usuarios olvidaron fácil el nombre de usuario y la contraseña que utilizaron, los parámetros de configuración como el nombre de la base de datos, del servidor en donde se encuentra alojada la SPODB, las direcciones de los servidores Web, etc., sin embargo eso es previsible y lo importante es que el usuario recuerde la información que debe colocar y en donde, ya que es fácil para cualquier técnico proporcionar estos datos.

Tabla 3-48 Prueba de Aceptación número cuatro.

CRITERIO DE PRUEBA
Qué tan fácil es para el usuario recobrar el control del sistema cuando ha ocurrido un error no catastrófico.
DEFINICIÓN DE LA PRUEBA
Luego de ocurrido un error, observar la forma en la que reacciona el usuario, que hace por tratar de resolver el problema y finalmente como logra salir del mismo.
RESULTADO
<ul style="list-style-type: none"> • Uno de los inconvenientes más comunes que se presentaba es que los usuarios cerraban el Panel de Tareas de Acción, y luego no encontraban la

manera de volverlo a abrir, lógicamente esto no es un error del sistema sino mas bien una propiedad de Microsoft Word (mostrar y ocultar los paneles de tareas), que los usuarios no han utilizando con anterioridad por lo que les resultó difícil recuperarse de este inconveniente; esto fue tomado en consideración y se capacitó a los usuarios para que los solucionen.

- Otro inconveniente no muy frecuente pero que sucedió y que los usuarios no pudieron solucionar por si solos, fue el desacoplamiento del Paquete de Expansión XML, esto es un error poco frecuente que genera Word cuando hay un apagón repentino del sistema, o cuando se genera un error del sistema operativo que hace que se cierren todas las aplicaciones abiertas. Word tiene un panel de recuperación para cuando ocurren estos problemas, en donde se puede volver a utilizar el o los últimos documentos utilizados por el usuario sin perder el trabajo, en el caso del Documento Inteligente el paquete de expansión en ciertas ocasiones (muy rara vez) se desasocia del documento y hay que volver a adjuntarlo. La forma de realizar esta tarea está especificada en el manual de instalación, y los usuarios con los que se realizó las pruebas no habían tenido la posibilidad de leerlo, por esta razón se les explicó como solucionar el inconveniente y en donde encontrar la información en caso de que lo requieran.
- Cuando ocurrían errores pequeños como ingresar mal los datos, o presionar por equivocación algún botón los usuarios no tenían inconveniente en resolverlo, en algún caso extremo se recurrió a cerrar Word y volverlo a abrir, pero el usuario pudo trabajar con normalidad luego de ello.

Tabla 3-49 Prueba de Aceptación número cinco.

CRITERIO DE PRUEBA
Qué tan placentero es el uso del sistema para los usuarios
DEFINICIÓN DE LA PRUEBA
Observar si el usuario luego de haber aprendido a utilizar el sistema prefiere utilizar la nueva aplicación u ofrece resistencia y prefiere mantenerse utilizando el método antiguo.
RESULTADO
<ul style="list-style-type: none"> • Para el Documento Inteligente, en vista de que no implica tener abierta una aplicación diferente a la que el usuario utiliza la mayor parte del tiempo (Word), y no resulta complicado manejarlo una vez que el usuario lo conoce a fondo, es utilizado con continuidad y comodidad, e incluso como permite realizar sin problema cualquier otro tipo de documento, los usuarios lo tienen abierto para redactar documentos que no tienen nada que ver con Proyectos de Titulación. • Para el caso del Formulario de Infopath, la situación es diferente ya que la misma funcionalidad está disponible en el Documento Inteligente, y los usuarios prefieren utilizar Word a utilizar Infopath, con lo cual se puede decir que el personalizar aplicaciones que los usuarios ya conocen si es una gran ventaja frente a desarrollar aplicaciones completamente nuevas, en cuanto a la adaptación y aprendizaje de los usuarios a un nuevo sistema.

Tabla 3-50 Prueba de Aceptación número seis.

CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES

- En lo personal, la única razón por la que descartaría la creación de una solución personalizada de Office como una alternativa para desarrollar aplicaciones que minimicen los problemas de oficina, es el costo de licenciamiento que una implementación de este tipo conllevaría, especialmente en nuestro país donde no está bien arraigada la cultura de utilización del software legal. En el caso de desarrollar este tipo de soluciones para instituciones que ya poseen las licencias, las cosas cambian completamente, ya que se estarían explotando de mejor manera los recursos económicos empleados, al utilizar al máximo las características del software que ya tienen adquirido.
- A pesar de que el marco teórico de este proyecto presenta una mínima parte de las características que pueden ser utilizadas por un usuario de Office, luego de la revisión bibliográfica realizada para este proyecto, se puede concluir que existen una gran cantidad de funciones, no tan conocidas, que muchas veces son consideradas como trucos utilizados por los usuarios experimentados o *Power Users* (como los denomina Microsoft). Un buen desarrollador de soluciones personalizadas de Office, debería tratar de conocer la mayoría de estas funciones para saber cuales son las herramientas con las que cuenta para crear sus aplicaciones; por ejemplo, si se sabe que en Word 2003 existe la posibilidad de presentar un historial de los cambios que se realizan a un documento (incluyendo: fechas, horas de modificación, usuarios responsables de los cambio, etc.), de seguro en el modelo de objetos se podrán encontrar propiedades y métodos que permitan manipular este historial por código, con ellos se podría crear fácilmente un registro de cambios de varios documentos a la vez, y presentarlo a manera de reporte en una hoja de Excel; tratar de hacer una aplicación con la misma funcionalidad sin ayuda del modelo de objetos de Office sería una tarea un tanto compleja y demorada. La idea

siempre debe ser reutilizar lo existente para mejorarlo o simplemente para no perder tiempo desarrollando algo que ya está hecho.

- Con la iniciativa de Microsoft de permitir que usuarios normales, con pocos conocimientos sobre tecnología, estén en contacto con el XML mediante la utilización del Office, se puede lograr que muchos procesos se realicen más rápidamente, incluso sin la ayuda de un profesional de tecnología; por esta razón, es recomendable profundizar sobre las particularidades de la utilización del XML, para que de esta manera se pueda ofrecer a cualquier persona que requiera un servicio, la solución más adecuada que se ajuste a sus necesidades, que incluso podría ser una simple capacitación para que aprenda a utilizar las funciones, que le permiten manejar información en formato XML, ofrecidas por las aplicaciones de Office.
- En el Internet se pueden encontrar algunos artículos que comparan la utilización de los Documentos Inteligentes versus los Formularios de InfoPath; como una conclusión de la experiencia en la utilización de estos dos elementos, adquirida durante el desarrollo de este trabajo, se puede decir que a pesar de que un Documento Inteligente podría realizar funciones similares a las de un Formulario de InfoPath para la recopilación de información mediante el llenado de formularios electrónicos, el InfoPath al ser una herramienta exclusivamente diseñada con este propósito, ofrece características mucho más sofisticadas para el tratamiento de la información; como por ejemplo, permite un acceso más fácil a fuentes de datos ya sean servidores de bases de datos o Servicios Web XML, presenta una interfaz de usuario más amigable a manera de formularios electrónicos con un estilo Web, etc., por esta razón cuando la solución a ser desarrollada implique la utilización de formularios electrónicos, la mejor opción técnicamente utilizando el Office, son los Formularios de InfoPath; sin embargo, económicamente habría que considerar que cada una de las máquinas que lo utilicen deben estar licenciadas. En el caso específico de este proyecto, el Formulario de InfoPath creado para la recopilación de las Observaciones que se realizan a cada uno de los Planes de Proyectos de

Titulación tiene una ventaja sobre el Documento Inteligente, que es el permitir trabajar con información de varios proyectos al mismo tiempo, intentar hacer una interfaz similar a la de InfoPath con un Documento Inteligente no sería una tarea nada fácil, sobretodo en el sentido de agregar y quitar las secciones como se lo hace en InfoPath.

- De la experiencia obtenida, se puede decir que luego de entender el modelo de objetos de las aplicaciones de Office, el desarrollo de soluciones personalizadas no es complejo, sin embargo hay que dedicar tiempo al aprendizaje de la utilización de los objetos más importantes, no es indispensable memorizar cada uno de los elementos que componen el modelo de objetos, pero si hay que saber como buscar la información; en realidad existen varios lugares donde conseguir ayuda sobre la utilización de los modelos de objetos, incluso existen libros completos dedicados únicamente a su estudio; como recomendación creo que es buena idea tener como referencia un libro, pero la herramienta que en realidad permite buscar objetos, métodos y propiedades rápidamente, brindando una ayuda muy valiosa, es sin duda el examinador de objetos incluido con el *Visual Basic Editor*.
- Se pudo notar que no todas las funciones del modelo de objetos de Office pueden ser utilizadas con código .NET; en especial, para el desarrollo de este proyecto se tuvieron problemas con instancias de objetos definidos en la librería de tipos STDOLE (archivo .lib independiente del modelo de objetos de Office); estos objetos son requeridos como parámetros por algunos métodos del modelo de objetos de Office (en especial aquellos relacionados con el manejo de íconos y tipos de letras), este problema fue detectado cuando se quiso dar una mejor apariencia a los controles de la barra de herramientas personalizada del Documento Inteligente; el ensamblado de interoperabilidad primaria asociado con la librería de tipos STDOLE; por ejemplo, no ofrece soporte para el método *LoadPicture* necesario para inicializar objetos del tipo *IPictureDisp* requeridos para poner íconos en los botones de las barras de herramientas, por esta razón

cuando se vaya a utilizar código administrado para personalizar el Office, es preferible prescindir de estos métodos, en vista de que se pierde mucho tiempo tratando de encontrar algún objeto que esté soportado en .NET y que sea equivalente al requerido.

- Al utilizar controles, tales como: cuadros de texto, *como box*, *check box*, etc., para personalizar la interfaz de usuario de las aplicaciones de Office (en especial para la programación del panel de tareas), hay que tener presente que en tiempo de diseño se trabaja sin tener ninguna ayuda visual que permita saber la ubicación de los controles en la pantalla (tiempo de ejecución); es decir, este tipo de controles no se pueden “arrastrar y colocar”, de igual forma no existe ninguna ayuda para generar controladores de eventos automáticamente; los métodos para controlar los eventos requieren de parámetros especiales por lo que se recomienda revisar las interfaces (ej. *CommandBarComboBox*, *CommandBarButton*, etc.) asociadas a los controles que se van a utilizar.
- En algunas ocasiones la ayuda existente para manejar el modelo de objetos de Office, no fue suficiente para poder utilizar algún objeto; por ejemplo, un botón de las barras de herramientas tiene una propiedad denominada ‘*OnAction*’, que permite especificar el nombre de un macro (código VBA) que se ejecutará al hacer un *clik* en el botón, esta propiedad no puede ser utilizada de ninguna manera desde código .NET, en su lugar hay que crear una instancia (del tipo *CommandBarButton*) que pueda contener una referencia al botón, y definir un método con los parámetros necesarios para controlar el evento *clik* del mismo. En realidad hay que contar con mucha suerte para encontrar ejemplos o documentación sobre éste y otros particulares más, por lo que en varias ocasiones al desarrollar soluciones personalizadas de Office con .NET, hay que tener mucha paciencia, analizar la interfaz del objeto que se está utilizando y en el último de los casos recurrir al método de prueba y error.

- Para el reconocimiento del texto dentro de un documento de Office, es muy importante especificar el idioma que se desea utilizar, ya que el componente COM *SmartTags*, que es el que implementa las interfaces necesarias tanto para las Etiquetas Inteligentes como para los Documentos Inteligentes, se apoya en el corrector ortográfico y gramatical utilizado en las aplicaciones de Office, para realizar el reconocimiento del texto; si el idioma no está bien especificado existen problemas para el reconocimiento de las palabras. Durante la realización de este proyecto, se hicieron varias pruebas utilizando el esquema MOSTL para la creación del Documento Inteligente, y aquí la especificación del idioma fue un inconveniente, ya que el archivo XML que definía al Documento Inteligente, al parecer no contenía ningún error y sin embargo no funcionaba correctamente; luego de revisar minuciosamente el archivo XML se detectó que el idioma no estaba bien especificado; se definió correctamente el idioma y todo funcionó sin problemas. Para una persona que no está familiarizada con el esquema MOSTL este error puede no ser tan evidente.
- Un problema que se presenta con el corrector ortográfico, es que al insertar una cantidad considerable de texto en un documento utilizando código; al parecer, no existe el tiempo necesario para hacer un correcto reconocimiento de las palabras y todas son marcadas como errores. Para evitar este molesto inconveniente, en un principio se decidió desactivar mediante código el corrector ortográfico, luego se insertaba el texto, y se volvía a activar el corrector, este método a pesar de que funcionaba no evitaba que el usuario aprecie una inserción secuencial (palabra por palabra) del texto; por esta razón, posteriormente se decidió utilizar el apoyo de un proceso secundario o mejor dicho un hilo del proceso principal, en donde se inserta el texto requerido, luego se copia todo al portapapeles de Windows, se pega el contenido del portapapeles en el documento de Word y se borra el contenido del portapapeles; este método no tiene fundamento bibliográfico, es uno de los resultados de varias horas de pruebas que se realizaron durante el desarrollo de este proyecto, con este método el texto será presentado al usuario de una sola vez y sin

marcas de errores ortográficos (da la impresión de apertura de documentos).

- Junto con los archivos que instala el Office, se incluyen varias ayudas (archivos **.chm**) que para un usuario normal pasarían desapercibidas y que en realidad tampoco serían de mucha utilidad, pero por otro lado constituyen una ayuda valiosa para un desarrollador de aplicaciones personalizadas de Office, estas ayudas se encuentran bajo el subdirectorio denominado 3082 del directorio principal donde se encuentra instalado el Office, aquí se puede encontrar información sobre el XML que debe ser utilizado en el Office, una referencia sobre el lenguaje VBScript que podría ser de mucha utilidad para el desarrollo de soluciones personalizadas de InfoPath con código no administrado, la documentación de todos los modelos de objetos de las aplicaciones de Office instaladas, etc., por esta razón se recomienda revisar el contenido de este directorio.
- Los estándares relacionados con el XML, no son difíciles de entender, pero si son un caso de estudio bastante largo si se quiere profundizar en el tema, por esta razón para cualquier persona que no esté familiarizada con la utilización del XML, es importante comenzar simplemente identificando claramente las características generales que debe contener un archivo XML, y seguir buscando la información según sea requerida, porque es muy fácil perderse con los diferentes esquemas, que permiten definir elementos, establecer restricciones, validaciones, etc. Uno de los estándares de mayor utilidad para el desarrollo de este proyecto fue el denominado lenguaje de rutas XPATH.
- A pesar de que los modelos de objetos de las aplicaciones de Office tiene cosas en común, la gran mayoría de objetos son particulares de cada aplicación y sería una buena práctica identificar los objetos principales antes de comenzar con la escritura de código; solo por poner un ejemplo, personalmente, antes de la realización de este proyecto, tenía algo de experiencia con la utilización del modelo de objetos de Access con VBA,

sin embargo, al comenzar a utilizar para este proyecto el modelo de objetos de Word con .NET fue una experiencia completamente distinta, lo mismo ocurrió con el modelo de objetos de Outlook, y con el modelo de objetos de Infopath, siendo este último el más distinto de los demás, dificultándose un poco su aprendizaje en vista de que la mayor parte de documentación está enfocada a la utilización de *scripts* para la personalización, y en lo personal considero que es más fácil traducir código de VBA a VB .NET, que código de VB *script* a VB .NET.

- Antes de presentar el Plan de Proyecto de Titulación, se realizaron algunas pruebas de escritura de código con VB .NET, C# .NET y C++ .NET, de donde se concluyó que el lenguaje que presenta mayores facilidades para el desarrollo de soluciones personalizadas de Office con código administrado, es el VB .NET; si bien es cierto que los tres lenguajes son similares en ciertos aspectos, siendo el principal que sean orientados a objetos (permiten: herencia, polimorfismo, sobrecarga de operadores, etc.), en este caso el Visual Basic .NET tiene la ventaja sobre los otros dos, debido a que una gran cantidad de código VBA puede ser copiado y compilado como si fuera VB .NET (en muchos de los casos basta con hacer pequeñas modificaciones para que compile el código), lo que no ocurre con C# o con C++, en donde hay que hacer varias modificaciones, o mejor dicho hay que traducir el código casi en su totalidad, esta tarea a pesar de no ser tan compleja, demanda más tiempo lo cual no es conveniente ni para el desarrollador ni para el usuario, por esta razón desde un principio se planteó desarrollar el proyecto con VB .NET.
- Entre los objetivos planteados para este proyecto, estaba la comparación de alternativas para la creación de soluciones personalizadas de Office, en el Capítulo 1 se presentaron algunas ideas generales sobre el desarrollo con algunas de las alternativas, en realidad se hicieron pequeñas pruebas y se compilaron algunos ejemplos con cada una de las tecnologías, pero en vista de que el tema central era la utilización del .NET y el XML, no se profundizó en el estudio de estas alternativas; sin embargo, con esto se

puedo concluir que cada una de las tecnologías tiene un uso específico, no son sustitutivas (pueden funcionar sin ningún problema en conjunto), y su utilización depende más bien de los requerimientos, del alcance del proyecto y de los recursos que se tenga a disposición. El principal inconveniente con la tecnología de Documentos Inteligentes, es que sólo funciona con Office 2003, y únicamente con Word y Excel, por esta razón no se puede descartar la utilización de los *COM Add-Ins* para ciertas soluciones, ya que también pueden ser desarrollados sin problema con código administrado (con las mismas ventajas de acceso a datos y seguridad que ofrecería una solución de Documentos Inteligentes). La ventaja de los *COM Add-Ins*, es que permiten personalizar casi todas las aplicaciones de Office y en sus diferentes versiones, y la ventaja que tiene la tecnología de Documentos Inteligentes y que no se puede realizar con ninguna de las otras alternativas, es la sensibilidad al contexto (gracias al etiquetado XML), por lo demás, la personalización de barras de herramientas, el acceso a datos y servicios, etc. son tareas que si se pueden realizar con otras alternativas. En conclusión, si se quieren soluciones seguras y robustas lo recomendable es utilizar código .NET ya sea en *COM Add-Ins* o en Documentos Inteligentes (siempre y cuando sea posible), y si se requieren soluciones rápidas y en cierto sentido temporales (pueden ser reemplazadas fácilmente) la mejor opción es utilizar tecnologías no administradas utilizando el VBA y los complementos de aplicación específica.

- En cuanto a la seguridad, también fue un objetivo ofrecer seguridad a diferentes niveles, en realidad las pruebas efectuadas permiten asegurar que este objetivo si se cumplió: el Documento Inteligente no se ejecuta si el archivo de manifiesto no está firmado digitalmente, los documentos tienen protección de párrafos, el Formulario de Infopath requiere de un proceso complejo de instalación (es transparente con el instalador creado) en donde se establecen directivas de seguridad difíciles de modificar, etc.; sin embargo, una solución nunca será cien por ciento segura. Lo que es importante tener presente, es que las aplicaciones de Office son muy

flexibles, y por ende muy vulnerables para ser atacadas, por esta razón no se deben descuidar las políticas de seguridad de ninguna manera.

- Personalmente creo que en este proyecto la tecnología estudiada más importante es el XML; la tecnología .NET y el Office a pesar de las ventajas que ofrecen, son productos Microsoft, y hay muchas personas que están en desacuerdo con la utilización de estos productos, pero el XML es un estándar no comercial que puede ser utilizado con una gran variedad de propósitos, por esta razón luego de haber estudiado un poco de sus ventajas y facilidades, como una recomendación creo que debería darse una pequeña introducción a su utilización en alguna de las materias que se dictan en las Carreras de Electrónica, en especial en Redes de Información.
- Finalmente, al desarrollar este proyecto lo que se buscaba era ofrecer una ayuda para las secretarías de cada una de las Carreras de Electrónica, por lo cual espero que la solución desarrollada sea de utilidad y que cumpla su objetivo principal: “Automatizar la tarea de elaboración de cartas y memorandos relacionados con los Proyectos de Titulación de los estudiantes”.

REFERENCIAS BIBLIOGRÁFICAS

- 1 Evan Lenz, Mary McRae, Simon St. Laurent., *Office 2003 XML*, (O'Reilly, Sebastopol, CA, 2004)
- 2 Davis McAmis, *How to Do Everything with Microsoft Office InfoPath 2003* (McGraw-Hill, New York Chicago, 2004)
- 3 Katherine Murray, *First Look Office 2003*, (Microsoft Press 2003), ISBN 07351914
- 4 Peter Kent, Cary Prague, *Microsoft Office 2003 Super Bible eBook*, (Wiley Publishing, Inc., 10475 Crosspoint Boulevard, 2003)
- 6 Análisis y Diseño de Sistemas, Tercera Edición, Kenneth E. Kendall & Julie E. Kendall, Pearson Educación, ISBN 968-880-694-3, Atlacomulco Num 500-5º Piso Col. Industrial Atoto, 53519, Naucalpan de Juárez Edo. De México.
- 7 Michael J. Young, Michael Halvorson, *Microsoft Office System Inside Out-2003 Edition*, (Microsoft Press, Redmond Washington, 2004).
- 8 Rod Stephens, *Microsoft Office Programming: A Guide for Experienced Developers*, (Apress, Springer-Verlag, NY, 2003).
- 9 Joe Habraken, *Microsoft Office 2003 All in One*, (QUE Publishing, U.S., Octubre 13 / 2003).
- 10 Woody Leonhard, *Office 2003 timesaving techniques for dummies* (Wiley Publishing, 2003).
- 11 *Microsoft Office XP Developer's Guide*, Microsoft Corporation, 2001/03/28 ISBN7356-1242-0
- 12 Debrauwer, Laurent y Heyde, Fien Van Der, *UML: Iniciación, Ejemplos y Ejercicios Corregidos* (Recursos Informáticos), ENI, ISBN: 2746027984
- 13 Richard Mansfield, *Office 2003 Application Development: All-In One desk Reference for Dummies* (Wiley Publishing, 2003).

REFERENCIAS WEB

- 13 http://es.wikipedia.org/wiki/Suite_ofim%C3%A1tica
Nombre página: "Wikipedia la Enciclopedia Libre - Suite Ofimática"
Última visita: 20/03/2006
- 14 http://es.wikipedia.org/wiki/Microsoft_Office
Nombre página: "Wikipedia la Enciclopedia Libre - MS Office"
Última visita: 20/03/2006
- 15 <http://www.pcworld.com/reviews/article/0,aid,111391,00.asp> (en inglés)
Nombre página: "*Sneak Peek: Microsoft Office 2003*"
Última visita: 20/03/2006
- 16 <http://www.desarrolladormsdn.com/msdn/SmartClients/default.aspx>
Nombre página:
Última visita: 20/03/2006
- 17 <http://www.jegsworks.com/Lessons-sp/presentations/basics/taskpane.htm>
Nombre página: "Fundamentos de PowerPoint: Panel de Tareas"
Última visita: 20/03/2006
- 18 <http://www.microsoft.com/office/editions/prodinfo/compare.msp>
Nombre página: "*Office 2003 Editions: Compare them to Previous Versions*"
Última visita: 20/03/2006
- <http://www.microsoft.com/latam/office/editions/prodinfo/topten.msp>
Nombre página: "10 principales beneficios de *Office Professional 2003*"
Última visita: 20/03/2006
- <http://www.microsoft.com/spain/office/editions/prodinfo/smallbusiness/tour/getmore.asp>
Nombre página: "*Tour por Microsoft Office Edición Pyme 2003*"
Última visita: 20/03/2006
- 19 <http://msdn.microsoft.com/office/technologyinfo/developing/overview/default.aspx>
Nombre página: "*Understanding Office Development*"
Última visita: 20/03/2006
- 20 http://msdn.microsoft.com/library/en-us/vbawd11/html/wotocOMMap_HV01049667.asp
Nombre página: "*Word Object Model [Word 2003 VBA Language Reference]*"
Última visita: 20/03/2006
- http://msdn.microsoft.com/library/en-us/vbaxl11/html/XltocOMMap_HV01049651.asp
Nombre página: "*Word Object Model [Excel 2003 VBA Language Reference]*"
Última visita: 20/03/2006
- http://msdn.microsoft.com/library/en-us/vbaof11/html/oftocOMMap_HV01047935.asp
Nombre página: "*Microsoft Office Object Model*"
Última visita: 20/03/2006

- http://msdn.microsoft.com/library/en-us/vbagr11/html/grtocObjectModelApplication1_HV05207683.asp
Nombre página: "Microsoft Graph Objects"
Última visita: 20/03/2006
- 21 <http://www.w3.org/TR/REC-xml>
Nombre página: "Extensible Markup Language (XML) 1.0 (Third Edition)"
Última visita: 20/03/2006
- 22 <http://www.w3.org/TR/xmlschema-0/>
Nombre página: "XML Schema Part 0: Primer Second Edition"
Última visita: 20/03/2006
- 23 <http://www.w3.org/TR/xslt>
Nombre página: "XSL Transformations (XSLT) Version 1.0"
Última visita: 20/03/2006
- 24 <http://www.w3.org/TR/xsl/>
Nombre página: "Extensible Stylesheet Language (XSL) Version 1.0"
Última visita: 20/03/2006
- 25 <http://www.w3.org/TR/xpath>
Nombre página: "XML Path Language (XPath) Version 1.0"
Última visita: 20/03/2006
- 26 <http://www.microsoft.com/downloads/details.aspx?familyid=ca83cb4f-8dee-41a3-9c25-dd889aea781c&displaylang=en>
Nombre página: "Word 2003: XML Software Development Kit (SDK)"
Última visita: 20/03/2006
- 27 <http://office.microsoft.com/es-hn/default.aspx>
Nombre página: "Office Online"
Última visita: 20/03/2006
- 28 <https://partner.microsoft.com/global/competency/iwsolutions/40020355>
Nombre página: "Take Advantage of Power, Performance, and Flexibility"
Última visita: 20/03/2006
- 29 http://www.elquille.info/colabora/lavb_msos/lavb_msos_m01.htm
Nombre página: "Laboratorio de Desarrollo MS Office System"
Última visita: 20/03/2006
- 30 http://msdn.microsoft.com/library/default.asp?url=/library/en-us/rssdk/html/rsconAboutRSSDK_HV01082263.asp
Nombre página: "Welcome to the Microsoft Office Research Service SDK"
Última visita: 20/03/2006

- 31 <http://www.oag.com/smarttag/&SmartTagVersion=1.0.0.0>
Nombre página: "Welcome to the OAG Smart Tag"
Última visita: 20/03/2006
- 32 <http://www.microsoft.com/downloads/details.aspx?displaylang=es&FamilyID=7e9ebc57-e115-4cac-9986-a712e22879bb>
Nombre página: "Kit de herramientas de InfoPath 2003 para VS .NET"
Última visita: 20/03/2006
- 33 <http://www.microsoft.com/downloads/details.aspx?FamilyID=351f0616-93aa-4fe8-9238-d702f1bfbab4&DisplayLang=en>
Nombre página: "InfoPath 2003 Software Development Kit (SDK)"
Última visita: 20/03/2006
- 34 http://www.msdn.microsoft.com/office/default.aspx?pull=/library/en-us/dno2k3ta/html/odc_smartdocswrapper.asp
Nombre página: "Smart Documents Made Easier for .NET Developers"
Última visita: 20/03/2006
- 35 <http://www.vertigosoftware.com/>
Nombre página: "Vertigo Software is a leading provider..."
Última visita: 20/03/2006
- 36 <http://www.microsoft.com/downloads/details.aspx?FamilyID=24a557f7-eb06-4a2c-8f6c-2767b174126f&DisplayLang=en>
Nombre página: "Office 2003 Smart Document SDK"
Última visita: 20/03/2006
- 37 <http://www.msdn.microsoft.com/library/default.asp?url=/library/en-us/sdsdk/html/sdconSecurityXEPDiagram.asp>
Nombre página: "Security Checks"
Última visita: 15/11/2005
- 38 <http://www.microsoft.com/downloads/details.aspx?FamilyID=4c586367-c733-4c68-9971-373c180ec114&DisplayLang=en>
Nombre página: "Office Tool: COM Shim Wizards"
Última visita: 20/03/2006
- 39 <http://www.microsoft.com/downloads/details.aspx?FamilyID=4bb7cb10-a6e5-4334-8925-3bcf308cfbaf&DisplayLang=en>
Nombre página: "Office 2003 Editions Resource Kit Tools"
Última visita: 20/03/2006

REFERENCIAS DE LA DOCUMENTACIÓN DE VISUAL STUDIO .NET

Para utilizar estas ayudas se requiere el *Microsoft Document Explorer*, incluido con el entorno de desarrollo de Visual Studio .NET.

- 40 ms-help://MS.MSDNQTR.2003FEB.3082/dntaloc/html/websvcs_platform.htm
Título del artículo: “Una plataforma para servicios Web”
- 41 <ms-help://MS.MSDNQTR.2003FEB.3082/vsintro7/html/vxoriAutomatingRepetitiveActions.htm>
Título del artículo: “Automatizar acciones repetitivas mediante macros”
- 42 ms-help://MS.MSDNQTR.2003FEB.3082/adosql/adoprg01_1kwv.htm
Título del artículo: “*Programming ADO SQL Server Applications*”
- 43 <ms-help://MS.MSDNQTR.2003FEB.3082/cpguide/html/cpconreflectionoverview.htm>
Título del artículo: “Información general sobre la reflexión”
- 44 <ms-help://MS.MSDNQTR.2003FEB.3082/netstart/html/cpqlom.htm>
Título del artículo: “Glosario de .NET *Framework*”
- 45 ms-help://MS.MSDNQTR.2003FEB.3082/com/html/cmi_q2z_9dwu.htm
Título del artículo: “*IUnknown*”
- 46 <ms-help://MS.VSCC.2003/ms.infopath.1033/ipclrcpt/html/749c2105-8b94-40ba-9a91-12bfb078f414.htm>
Título del artículo: “Niveles de seguridad de InfoPath”.
- 47 ms-help://MS.MSDNQTR.2003FEB.3082/com/html/com_757w.htm
Título del artículo: “*The Component Object Model*”
- 48 <ms-help://MS.VSCC.2003/MS.MSDNQTR.2003FEB.3082/dnmaq00/html/GCI.htm>
Título del artículo: “*Garbage Collection: Automatic Memory Management in the Microsoft .NET Framework*”
- 49 <ms-help://MS.MSDNQTR.2003FEB.3082/cpguide/html/cpconmarshalingselectedinterfaces.htm>
Título del artículo: “Calcular referencias de interfaces seleccionadas”
- 50 <ms-help://MS.VSCC.2003/MS.MSDNQTR.2003FEB.3082/dncscol/html/csharp07192001.htm>
Título del artículo: “Escribir código para excepciones”
- 51 <ms-help://MS.MSDNQTR.2003FEB.3082/cpguide/html/cpconruntimecallablewrapper.htm>
Título del artículo: “Los contenedores a los que se puede llamar”

- 52 <ms-help://MS.MSDNQTR.2003FEB.3082/cpguide/html/cpconcomwrappers.htm>
Título del artículo: “Contenedores COM”
- 53 <ms-help://MS.VSCC.2003/MS.MSDNQTR.2003FEB.3082/vblr7/html/vakeyModule.htm>
Título del artículo: “*Module (Instrucción)*”
- 54 <ms-help://MS.VSCC.2003/MS.MSDNQTR.2003FEB.3082/cptools/html/cpgrfcodeaccesssecuritypolicyutilitycaspol.exe.htm>
Título del artículo: “Herramienta Directiva de seguridad de acceso a código (Caspol.exe)”
- 55 <ms-help://MS.VSCC.2003/MS.MSDNQTR.2003FEB.3082/cptools/html/cpgrfcertificatecreationtoolmakecert.exe.htm>
Título del artículo: “Herramienta Creación de certificados (Makecert.exe)”
- 56 <ms-help://MS.VSCC.2003/MS.MSDNQTR.2003FEB.3082/cptools/html/cpgrffilesigningtoolsigncode.exe.htm>
Título del artículo: “Herramienta Firma de archivos (Signcode.exe)”
- 57 <ms-help://MS.VSCC.2003/ms.infopath.1033/ipclrcpt/html/836d03a1-2705-40fc-9ba8-81342deac429.htm>
Título del artículo: Ensamblados de Interoperabilidad de InfoPath
- 58 ms-help://MS.VSCC.2003/MS.MSDNQTR.2003FEB.3082/dv_wrcore/html/wrconWhiteRabbitArchitecture.htm
Título del artículo: “Arquitectura de las soluciones de Office que utilizan extensiones de código administrado”

REFERENCIA DE PROYECTO DE TITULACIÓN

- 59 Torres Jaramillo Cristina, Gudiño Carrillo Santiago, INTEGRACIÓN DE SERVICIOS WEB, LA TECNOLOGÍA ASP.NET Y LA PLATAFORMA DE SERVICIOS DE REPORTE SQL PARA EL DESARROLLO DE UN SISTEMA DISTRIBUIDO, (EPN 2005)

ANEXO A

MODELO DE OBJETOS DE MICROSOFT OFFICE

Fuente: *Microsoft Office XP Developer's Guide*

Sección: “APPENDIX B - Microsoft Office Developer Object Model Guide”

Autor: © Microsoft Corporation, ISBN7356-1242-0

Un modelo de objetos describe las características que una aplicación pone a disposición de los desarrolladores; es decir, el modelo de objetos está conformado por las definiciones de clases, los objetos específicos y los métodos que un desarrollador puede utilizar para personalizar una aplicación.

En el caso particular de Microsoft Office existe un modelo de objetos por cada una de sus aplicaciones, y un modelo de objetos general con los elementos comunes.

La Clase *Application*

Todas las aplicaciones de Office definen una clase ***Application***, la misma que constituye la base en la jerarquía de objetos. Cuando una aplicación inicia, ésta crea una instancia de la clase ***Application***.

Mediante código se puede utilizar el objeto ***Application*** para manipular la aplicación en sí. El objeto ***Application*** puede contener referencias a otros objetos utilizados por la aplicación.

El modelo de objetos de cada una de las diferentes aplicaciones que constituyen Microsoft Office tiene algunas similitudes. Todas las aplicaciones de Microsoft Office poseen objetos ***Application***; todas tienen algún objeto que representa el documento que la aplicación manipula; por ejemplo, Word tiene el objeto ***Document***, Excel tiene el objeto ***Workbook***, PowerPoint tiene el objeto

Presentation, Access tiene el objeto **Database** y Outlook dispone de varios objetos como **MailItem** y **ContactItem**.

Todas las aplicaciones de Office, excepto Access, tienen una clase que representa un grupo de objetos que el usuario ha seleccionado: la clase **Selection**. Todas las aplicaciones de Office, excepto Access y Outlook, disponen de una clase que representa un grupo de objetos: la clase **Range**.

Se debe notar, que una propiedad de un objeto por lo general devuelve una referencia a una clase con el mismo nombre que la propiedad; por ejemplo: el objeto **Paragraphs** tiene una propiedad **Range** que devuelve un objeto de la clase **Range**. En la documentación de Microsoft, se hace referencia a un ítem como objeto, propiedad o método.

El “Objeto” *Global*

Existen algunos objetos y propiedades que son comunes e independientes de un tipo de documento en particular. Estos objetos y propiedades han sido colocados dentro de la clase **Application**, y pueden ser accedidos mediante un “objeto” Global. Este “objeto” Global permite acceder a las propiedades y métodos sin tener que utilizar el calificador del objeto, la palabra **Application**.

Por ejemplo, las siguientes dos sentencias de código VBA realizan la misma tarea:

```
MsgBox Application.RecentFiles(1).Name ' Usando el objeto Application
MsgBox RecentFiles(1).Name           ' Usando el “objeto” Global
```

Cabe mencionar que este “objeto” Global no puede ser utilizado como un objeto, sólo en la literatura se lo llama objeto.

Cada aplicación de Microsoft Office dispone de una gran cantidad de clases, propiedades, métodos, enumeraciones (constantes predefinidas), colecciones y eventos que pueden ser utilizados por los programadores. La información necesaria para hacer uso del modelo de objetos de Microsoft Office se la puede

obtener mediante varias herramientas, pero en especial se pueden mencionar dos: la ayuda en línea y el Examinador de Objetos.

El Modelo de Objetos de Word

Todos los elementos de Microsoft Word (documentos, tablas, párrafos, marcadores, campos, etc.) pueden representarse mediante objetos. Un objeto representa un elemento de Word, como un documento, un párrafo, un marcador o un carácter.

Una colección es un objeto que contiene otros objetos, normalmente del mismo tipo; por ejemplo, todos los objetos **bookmark** de un documento están incluidos en un único conjunto. Mediante propiedades y métodos, se puede modificar un único objeto o todo un conjunto de objetos.

Una propiedad es el atributo de un objeto o un aspecto de su comportamiento; por ejemplo, en las propiedades de un documento se incluyen el nombre, el contenido y el estado con que se guarda; para cambiar las características de un objeto, se cambian los valores de sus propiedades.

Un método es una acción que puede realizar un objeto; por ejemplo, igual que puede imprimirse un documento, el objeto **Document** tiene un método **PrintOut**. Con frecuencia, los métodos tienen argumentos que califican cómo se realiza la acción.

En la mayoría de los casos, los métodos son acciones y las propiedades cualidades. Si se utiliza un método la acción recae sobre el objeto, mientras que si se usa una propiedad se devuelve información acerca del objeto o se cambia una de sus cualidades.

A continuación se presentan algunas características de los objetos más importantes del modelo de Objetos de Word.

Objeto *Application*

El objeto ***Application*** representa la aplicación Word y es el padre de todos los otros objetos. Este objeto permite controlar el ambiente de Word mediante sus propiedades y métodos.

Objeto *Document*

El objeto ***Document*** es la parte central al programar con Word. Cada vez que se abre un documento existente o se crea un nuevo documento, se está creando un nuevo objeto ***Document***, el cual es añadido a la colección ***Documents***. El documento que tiene el foco se conoce como documento activo y está representado en el objeto ***Application*** mediante la propiedad ***ActiveDocument***.

Objeto *Selection*

El objeto ***Selection*** representa el área actualmente seleccionada. Cuando se realiza una operación en la interfaz de usuario de Word, como cambiar el formato del texto, se selecciona el texto y luego se le aplica un formato. El objeto ***Selection*** siempre está presente en un documento, incluso si el usuario no ha seleccionado nada, en cuyo caso, el objeto ***Selection*** representa el punto de inserción. El objeto ***Selection*** incluso puede contener bloques de texto no contiguo.

Objeto *Range*

El objeto ***Range*** representa un área contigua en un documento, y está definido por la posición inicial de un carácter y por la posición final de otro carácter. Se pueden tener varios objetos ***Range*** en un mismo documento. Un objeto ***Range*** posee las siguientes características:

- Puede consistir de: el punto de inserción, un rango de texto o de todo el documento.
- Puede incluir caracteres no imprimibles, como espacios, tabulaciones o marcas de párrafos.
- Puede ser el área representada por la selección actual o puede representar un área diferente a la seleccionada.

- Es dinámico, sólo existe mientras el código que lo creó está siendo ejecutado.

Cuando se inserta texto al final de un rango, Word de forma automática expande el rango para incluir el texto insertado.

Objeto Bookmark

El objeto **Bookmark** es similar al objeto **Range** en el sentido de que ambos representan áreas contiguas en un documento, con un punto inicial y un punto final. Un *bookmark* permite marcar una ubicación en un documento, o puede ser un contenedor de texto en un documento.

Un objeto **Bookmark** puede consistir de un punto de inserción únicamente o puede llegar a ser tan grande como el documento completo. En un documento se puede tener más de un objeto **Bookmark**. Un objeto **Bookmark** posee las siguientes características:

- Se puede asignar un nombre al objeto.
- Se guarda con el documento, y no se elimina cuando el código que lo creó deja de ser ejecutado o si se cierra el documento.
- Por defecto está oculto, pero puede ser visible configurando la propiedad ShowBookmarks del objeto **View**.

En la Tabla A - 1 se presenta una descripción breve de otros de los objetos típicos de Microsoft Word 2003.

Objeto	Descripción
Break	Representa saltos de página, columna y sección individuales de una página.
Breaks	Representa un conjunto de saltos de página, de columna o de sección en una página.
Editor	Representa un único usuario a quien se han otorgado permisos específicos para editar partes de un documento.
Editors	Representa un conjunto de objetos Editor que representa a un conjunto de usuarios o grupos de usuarios a los que se han concedido permisos específicos para editar partes de un documento.

Line	Representa una línea de texto específica en un objeto Rectangle .
Lines	Representa un conjunto de objetos Line que representa las líneas de texto de un objeto Rectangle .
Page	Representa una página de un documento.
Pages	Representa un conjunto de páginas de un documento.
Rectangle	Representa una parte de texto o un gráfico de una página.
Rectangles	Representa un conjunto de objetos Rectangle de una página que representan partes de texto y gráficos.
SmartTagAction	Representa una acción para una etiqueta inteligente.
SmartTagActions	Representa un conjunto de acciones para una sola etiqueta inteligente o un tipo de etiqueta inteligente.
SmartTagRecognizer	Representa un componente instalado que etiqueta los datos con tipos de información.
SmartTagRecognizers	Representa componentes instalados que etiquetan datos con tipos de información.
SmartTagType	Representa un tipo de etiqueta inteligente. Los tipos de etiqueta inteligente se identifican como elementos aislados en las listas de etiquetas inteligentes.
SmartTagTypes	Representa un conjunto de objetos SmartTagType .
XMLChildNodeSuggestion	Representa un nodo que es un posible elemento secundario del elemento actual, según el esquema, pero del que no hay garantía de que sea válido.
XMLChildNodeSuggestions	Representa un conjunto de elementos que pueden ser secundarios válidos del elemento especificado según el esquema.
XMLNamespace	Representa un esquema individual dentro de la Biblioteca de esquemas.
XMLNamespaces	Representa el conjunto completo de esquemas de la Biblioteca de esquemas.
XMLNode	Representa un elemento XML aislado aplicado a un documento.
XMLNodes	Representa los nodos en la vista de árbol del panel de tareas Estructura XML , que indica los elementos aplicados por un usuario a un documento.
XMLSchemaReference	Representa un esquema XML adjunto a un documento.
XMLSchemaReferences	Representa un conjunto de los espacios de nombres exclusivos adjuntos a un documento.
XSLTransform	Representa un archivo XSLT (transformación del lenguaje de hojas de estilo extensibles, XSL) registrado.
XSLTransforms	Representa todas las transformaciones XSL de un espacio de nombres XML específico.

Tabla A - 1 Objetos de Microsoft Word 2003

En la Tabla A -2 se enumeran los objetos exclusivos del modelo de objetos de Microsoft Office 2003, no existen en el modelo de objetos de versiones anteriores.

Objeto	Descripción
<u>DocumentLibraryVersion</u>	Copia de seguridad o versión única de un documento compartido.
<u>DocumentLibraryVersions</u>	Colección de objetos DocumentLibraryVersion .
<u>Permission</u>	Administra los permisos de documentos; además es una colección de objetos UserPermission .
<u>SharedWorkspace</u>	Administra un documento guardado en un área de trabajo compartida.
<u>SharedWorkspaceFile</u>	Archivo guardado en un área de trabajo compartida.
<u>SharedWorkspaceFiles</u>	Colección de objetos SharedWorkspaceFile .
<u>SharedWorkspaceFolder</u>	Subcarpeta de un área de trabajo compartida.
<u>SharedWorkspaceFolders</u>	Colección de objetos SharedWorkspaceFolder .
<u>SharedWorkspaceLink</u>	Vínculo guardado en un área de trabajo compartida.
<u>SharedWorkspaceLinks</u>	Colección de objetos SharedWorkspaceLink .
<u>SharedWorkspaceMember</u>	Usuario de un área de trabajo compartida.
<u>SharedWorkspaceMembers</u>	Colección de objetos SharedWorkspaceMember .
<u>SharedWorkspaceTask</u>	Tarea guardada en un área de trabajo compartida.
<u>SharedWorkspaceTasks</u>	Colección de objetos SharedWorkspaceTask .
<u>SmartDocument</u>	Administra un paquete de expansión XML adjunto que convierte un documento en un documento inteligente.
<u>Sync</u>	Administra la sincronización entre la copia local y la copia del servidor de un documento compartido.
<u>UserPermission</u>	Conjunto de permisos de un documento para un solo usuario.

Tabla A - 2 Objetos de Microsoft Word 2003.

ANEXO B

ELEMENTOS DEL VBE Y LA UTILIZACIÓN DEL EXAMINADOR DE OBJETOS

Fuente: *Microsoft Office System Inside Out-2003 Edition*

Sección: Escritura de Macros

Autor: Michael J. Young, Michael Halvorson

Elementos del Editor Visual Basic (VBE - *Visual Basic Editor*)

El editor Visual Basic, muestra una interfaz de usuario bastante similar a la de cualquier ambiente de programación; presenta herramientas básicas típicas; tales como, ventana de propiedades, explorador de archivos, botón para depurar, etc. y herramientas que ayudan al trabajo con componentes COM, como el administrador de complementos. En la Figura B - 1 se muestra el interfaz de usuario del Editor Visual Basic.

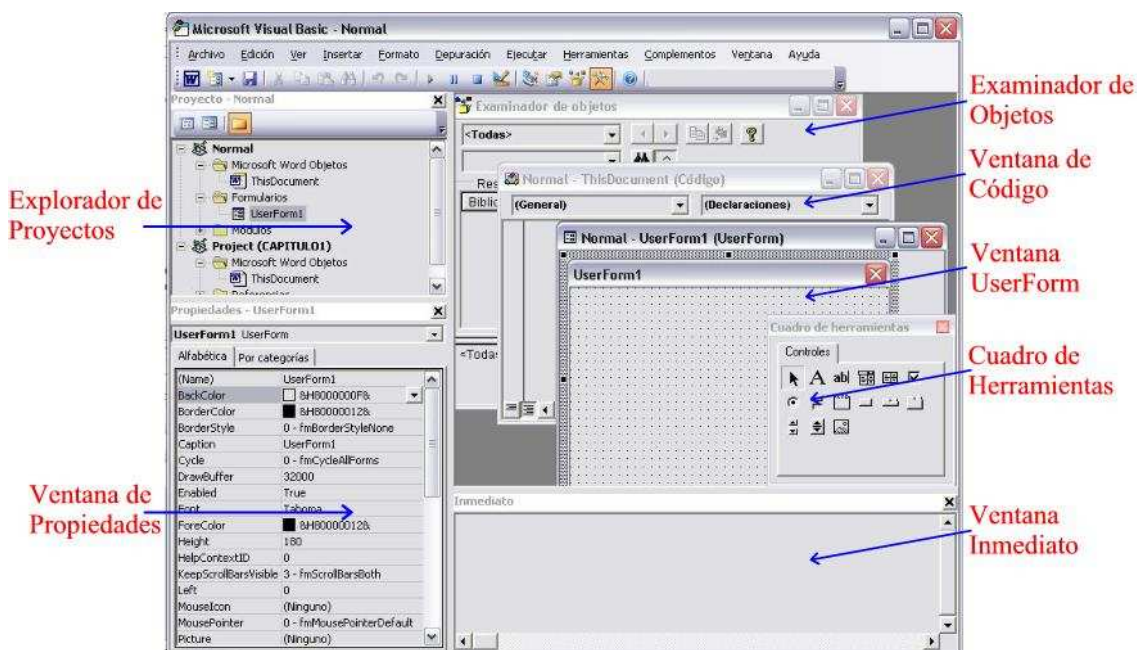


Figura B - 1 Interfaz del Editor Visual Basic

- **Explorador de Proyectos:** Muestra una lista jerárquica de proyectos y todos los elementos contenidos y referenciados por ellos.
- **Ventana de Código:** Permite escribir todo el código Visual Basic.
- **Ventana *UserForm*:** Contiene un formulario de usuario que permite crear cuadros de dialogo personalizados.
- **Ventana de Propiedades:** Muestra las propiedades de los controles *ActiveX* utilizados en un formulario, ordenados alfabéticamente o por categorías.
- **Cuadro de Herramientas:** Muestra un conjunto de controles *ActiveX* disponibles, para integrarlos en los formularios.
- **Examinador de Objetos:** Permite navegar a través de objetos, métodos, propiedades y eventos proporcionados por componentes COM.
- **Ventana Inmediato:** Permite ingresar y ejecutar una línea de código de Visual Basic y ver si funciona.
- **Examinador de Objetos:** Para cumplir con el objetivo de desarrollar cualquier tipo de soluciones de Office (código administrado o no administrado), el Examinador de Objetos, es una herramienta muy útil para el trabajo con el modelo de objetos de las aplicaciones de Office. En la Figura B - 1 se muestra el interfaz de usuario del Examinador de Objetos.

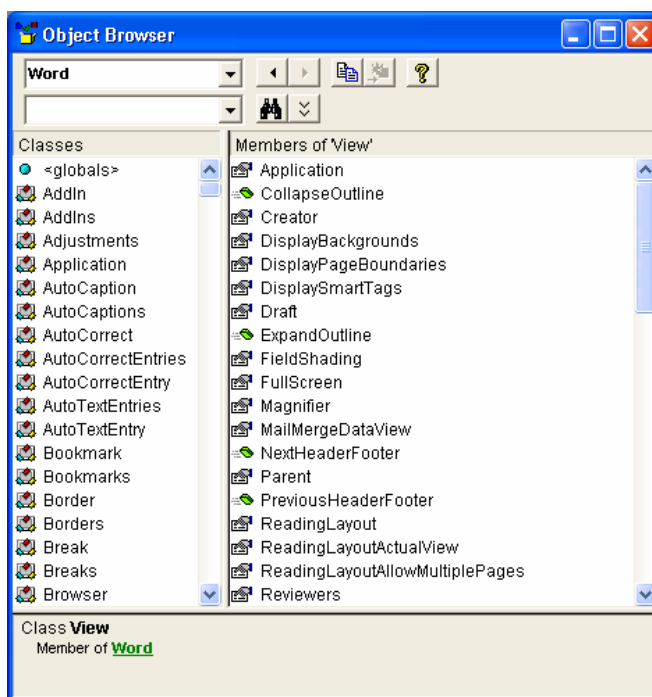


Figura B - 2 Interfaz de Usuario del Examinador de Objetos.

- **Caja de selección Proyecto/Librería:** Permite presentar todas las librerías de tipos referenciados disponibles.
- **Caja de Búsqueda:** Permite presentar la lista de palabras buscadas o permite ingresar una palabra para realizar una búsqueda en las librerías disponibles mediante el botón de **Búsqueda**. Los resultados de una búsqueda presentan cualquier clase, método, evento o constante que contenga la palabra ingresada en la caja.
- **Lista de Clases:** Despliega todos los objetos y colecciones de la librería seleccionada.
- **Miembros de la Lista:** Despliega todos los métodos, propiedades, eventos y constantes asociadas con el objeto seleccionado en la lista de Clases.
- **Panel de Detalles:** Despliega información acerca del ítem seleccionado, como el tipo de objeto, el tipo de dato, los argumentos que puede tomar, la librería o colección a la que pertenece.

ANEXO C

ENSAMBLADOS DE INTEROPERABILIDAD PRIMARIA REQUERIDOS PARA PERSONALIZAR EL OFFICE 2003 CON CÓDIGO ADMINISSTRADO

Fuente: Documentación del Visual Studio .NET

Artículo: “Ensamblados de interoperabilidad primaria de Office”

Autor: © Microsoft Corporation.

Aplicación o Componente de Office	Nombre del Ensamblado de Interoperabilidad Primaria
Biblioteca de objetos de Microsoft Access 11.0	microsoft.office.interop.access.dll
Biblioteca de objetos de Microsoft Excel 11.0	microsoft.office.interop.excel.dll
Biblioteca de referencia de objetos Web de Microsoft FrontPage 6.0	microsoft.office.interop.frontpage.dll
Biblioteca de referencia de objetos de página de Microsoft FrontPage 6.0	microsoft.office.interop.frontpageeditor.dll
Biblioteca de objetos de Microsoft Graph 11.0 (utilizada por PowerPoint, Access y Word para gráficos)	microsoft.office.interop.graph.dll
Biblioteca de objetos de Microsoft Outlook 11.0	microsoft.office.interop.outlook.dll
Control de vistas de Microsoft Outlook (se puede utilizar en páginas y aplicaciones Web para tener acceso a la bandeja de entrada)	microsoft.office.interop.outlookviewctl.dll
Microsoft Office Web Components 11.0 (hoja de cálculo, gráfico y tabla dinámica de páginas Web)	microsoft.office.interop.owc11.dll
Biblioteca de objetos de Microsoft PowerPoint 11.0	microsoft.office.interop.powerpoint.dll
Biblioteca de objetos de Microsoft Publisher 11.0	microsoft.office.interop.publisher.dll
Biblioteca de tipos de Etiquetas inteligentes de Microsoft 2.0	microsoft.office.interop.smarttag.dll
Biblioteca de objetos de Microsoft Word 11.0	microsoft.office.interop.word.dll
Microsoft Visual Basic para extensibilidad de aplicaciones 5.3	microsoft.vbe.interop.dll
Biblioteca de objetos de Microsoft Forms 2.0 (controles programables que se pueden utilizar en las aplicaciones)	microsoft.vbe.interop.forms.dll
Biblioteca de objetos de Microsoft Office 11.0 (funciones compartidas de Office)	office.dll

Tabla C - 1 Ensamblados de Interoperabilidad de Office 2003.

Nota: Algunos ensamblados se agregan automáticamente a un proyecto cuando se incluye un ensamblado que hace referencia a ellos. Por ejemplo, las referencias a los ensamblados Office.dll y microsoft.vbe.interop.dll se agregan automáticamente cuando se incluye una referencia a los ensamblados de Word, Excel, Microsoft *Forms* o *Graph*.

Cuando se crea un nuevo proyecto de Microsoft Office Word 2003 o Microsoft Office Excel 2003 en Visual Studio .NET, se agrega automáticamente una referencia al ensamblado de interoperabilidad primaria (PIA) en el proyecto si el ensamblado está instalado en la caché de ensamblados global (GAC) de Windows. Los ensamblados de interoperabilidad primaria de Microsoft Office 2003 se instalan en la caché de ensamblados global durante una instalación Completa.

En una instalación Personalizada, puede seleccionar los ensamblados que desea instalar. De forma predeterminada, los ensamblados de interoperabilidad primaria se marcan para instalarse durante el primer uso; sin embargo, esto sólo funciona cuando un usuario final ejecuta la solución de Office, y no en el equipo de desarrollo cuando se agrega una referencia a un ensamblado.

Para utilizar las características de las aplicaciones de Office 2003 distintas de Word y Excel mediante código administrado, debe agregar manualmente una referencia a la aplicación o componente apropiados en el proyecto. Al agregar una referencia al componente, se crea una referencia al ensamblado de interoperabilidad primaria, si el ensamblado está instalado en la caché de ensamblados global. A las aplicaciones y componentes de Office se tiene acceso desde la ficha COM del cuadro de diálogo Agregar referencia.

Si el ensamblado de interoperabilidad primaria no está instalado en la caché de ensamblados global, Visual Studio .NET genera un ensamblado de interoperabilidad para el proyecto que podría no funcionar en todas las circunstancias

ANEXO D

GUÍA DE INSTALACIÓN DE LOS ENSAMBLADOS DE INTEROPERABILIDAD PRIMARIA DE OFFICE

- Fuente:** Documentación del Visual Studio .NET
- Artículo:** “Cómo: Instalar ensamblados de interoperabilidad primaria de Office”
- Autor:** © *Microsoft Corporation.*

Cuando se realiza una instalación completa, se instalan los ensamblados de interoperabilidad primaria de Microsoft Office 2003. En una instalación personalizada, puede elegir instalarlos siguiendo el procedimiento siguiente, a partir del paso 5.

Cuando se realiza una instalación típica, los ensamblados de interoperabilidad primaria están configurados para instalarse al utilizarlos por primera vez. Sin embargo, esta opción de instalación sólo funciona en equipos de usuarios finales cuando un usuario final ejecuta una solución de Office. Agregar una referencia a un ensamblado de interoperabilidad primaria en un proyecto de Visual Studio .NET no instala el ensamblado en el equipo de desarrollo.

Hay cuatro ensamblados de interoperabilidad primaria, necesarios para casi todas las soluciones de Office, que utilizan extensiones de código administrado. Si instala los ensamblados de interoperabilidad primaria manualmente, asegúrese de incluir:

- Microsoft Office Excel 2003
- Microsoft Office Word 2003
- Microsoft Forms 2.0 (en Office Tools)
- Microsoft Graph (en Office Tools)

Nota Cuando agregue estos ensamblados, los ensamblados a los que se hace referencia, como Office.dll y microsoft.vbe.interop.dll se agregan automáticamente.

Las opciones de Compatibilidad con programación de .NET sólo aparecen si Microsoft .NET *Framework* 1.1 está instalado en el equipo. Microsoft .NET *Framework* 1.1 se instala con Visual Studio .NET 2003. Se puede descargar de la página Web Microsoft .NET *Framework Redistributable* (<http://msdn.microsoft.com/library/default.asp?url=/downloads/list/netdevframework.asp>).

Para instalar los ensamblados de interoperabilidad primaria de Office

1. En el **Panel de control**, elija **Agregar o quitar programas**.
2. Seleccione Microsoft Office en la lista de **Programas actualmente instalados**.
3. Haga clic en **Cambiar**.
4. Seleccione **Agregar o quitar funciones** y haga clic en **Siguiente**.
5. Seleccione **Elija la personalización avanzada de las aplicaciones** y haga clic en **Siguiente**.
6. Expanda **Microsoft Excel para Windows**, haga clic en el cuadro situado junto a **Compatibilidad con programación de .NET** y elija **Ejecutar desde mi PC**.
7. Expanda **Microsoft Word para Windows**, haga clic en el cuadro situado junto a **Compatibilidad con programación de .NET** y elija **Ejecutar desde mi PC**.
8. Expanda **Herramientas de Office**, haga clic en el cuadro situado junto a **Compatibilidad con programación de Microsoft Forms 2.0 .NET** y elija **Ejecutar desde mi PC**.
9. En **Herramientas de Office**, expanda **Microsoft Graph**, haga clic en el cuadro situado junto a **Compatibilidad con programación de .NET** y elija **Ejecutar desde mi PC**.
10. Siga el mismo procedimiento para instalar ensamblados de interoperabilidad primaria para cualquier otra aplicación de Office.
11. Haga clic en **Actualizar**.

ANEXO E

DIAGRAMAS DE FLUJO DE DATOS Y DICCIONARIO DE DATOS

Fuente 1: Análisis y Diseño de Sistemas, Tercera Edición, ISBN 968-880-694-3

Sección: “Diagramas de Flujo de Datos”

Autor: Kenneth E. Kendall & Julie E. Kendall

Fuente 2: http://www.gestiopolis.com/recursos2/documentos/fulldocs/fin/modproolga_archivos/frame.htm

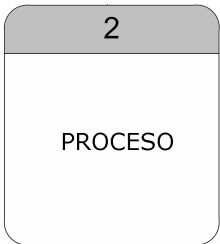
Sección: “Conceptos básicos para la modelización de los procesos de un sistema de información, utilizando los diagramas de flujo de datos.”

Autor: Olga García

DIAGRAMAS DE FLUJO DE DATOS (DFD)

El DFD es una de las herramientas del análisis estructurado moderno, más importante para el análisis de modelos gráficos, que permite visualizar un sistema como una red de procesos funcionales conectados entre sí por canales (flujo de datos) y depósitos de almacenamiento de datos. Estos diagramas nos permiten ver como los datos fluyen a través de la organización, los procesos y transformaciones que sufren dichos datos y los diferentes tipos de salidas.

COMPONENTES Y SÍMBOLOS UTILIZADOS

	<p>PROCESOS</p> <p>Generalmente, el proceso (función ó transformación) viene representado por un círculo o por una burbuja, y son acciones que se toman sobre los datos, como por ejemplo, Calcular, Comparar, Imprimir, señalar, marcar, autorizar, almacenar, validar, informar, producir, otros. Los procesos muestran una parte del sistema que transforma entradas en salidas, esto es, muestra cómo es que una o varias entradas se transforma en una o varias salidas.</p>
---	--




	<p>FLUJO DE DATOS</p> <p>Son vectores etiquetados o flechas, o simplemente líneas con notación direccional, que muestran el contenido de lo que entra o sale de un proceso. Además, muestran el movimiento de bloques o paquetes de información de un lugar del sistema a otro. La punta de la flecha señala el destino u origen de los datos.</p>
	<p>ALMACÉN</p> <p>Es el depósito de los datos, que se utiliza para modelar una colección de paquetes de datos en reposo. Se denota por dos líneas paralelas, pero cualquier símbolo sugerido es aceptado. Además de la forma física que toma un almacén, éste puede existir por necesidad o por conveniencia. Un almacén puede ser necesario cuando dos procesos ocurren en momentos diferentes, por ejemplo, el proceso de entrada de órdenes puede operar en tiempos diferentes que el proceso de investigación de órdenes.</p>
	<p>ENTIDADES EXTERNAS</p> <p>Las entidades externas o agentes externos e internos, son las fuentes o destinos de los datos. Normalmente, se considera como externo a un agente cuando es claramente exterior a la empresa, algunos ejemplos de éstos son: Clientes, Proveedores, etc. Los agentes son internos, cuando se refieren a tareas efectuadas dentro de la empresa pero que no forman parte del sistema; sin embargo, suministran entradas o reciben salidas de él. Se pueden citar como agentes internos otros departamentos, empleados o sistemas de información. Los agentes internos pueden englobar también a los usuarios finales de un sistema, que con frecuencia son fuentes de las entradas (datos) y destinos de las salidas (información).</p>

Tabla E - 1 Símbolos de los Diagramas de Flujo de Datos.

PASOS PARA LA CONSTRUCCIÓN DE LOS DIAGRAMAS DE FLUJO DE DATOS

Elaborar un diagrama de flujo de datos de CONTEXTO

Éste ubica el sistema dentro de un contexto de entorno; vale decir, como interactúa el sistema con otros sistemas y con la empresa considerada en su conjunto. Define el campo de acción y los límites del sistema y el proyecto.

Al dibujar un diagrama de contexto:

- Use un solo símbolo de proceso (asignando el número cero al proceso).
- Rotule el símbolo de proceso de modo que represente todo el sistema. Se puede usar un verbo más un objeto.
- Incluya todas las entidades externas del sistema.
- Muestre todos los flujos de datos entre las entidades.

Estrategias para determinar los flujos de datos:

- Pregunte a sus usuarios finales cuáles son los sucesos o transacciones a los cuales debe responder el sistema.
- Para cada suceso, pregunte a sus usuarios finales cuáles son las respuestas que debería producir el sistema.
- Pregunte cuáles son los informes de formato fijo que ha de producir el sistema.
- Identifique las fuentes netas de datos para cada suceso.
- Identifique los recipientes netos de cada respuesta o salida que debería generar el sistema.
- Identifique todos los posibles almacenes de datos externos.
- Dibuje un diagrama de contexto para todas las informaciones anteriores.

Elaborar un Diagrama de Flujo de Datos General (Diagrama 0)

Si es necesario documentar un sistema con mayor detalle que el Diagrama 0, se puede usar uno o más Diagramas n. Un Diagrama n documenta un solo proceso de un DFD con mayor detalle. La n representa el número del proceso del siguiente nivel más alto que se está documentando.

Guía para la construcción del Diagrama 0:

- Se comienza su construcción una vez que se conozcan sus componentes, los cuales deben ser identificados conjuntamente con los usuarios.
- Escoger nombres significativos perdurables para los componentes.

- Numerar los procesos para que sirvan de referencia para su explosión posterior.
- Evitar los DFD excesivamente complejos.
- Mantener la consistencia entre los procesos y los otros modelos.
- Evite sumideros infinitos - burbujas que solo tienen entradas pero no salidas.
- Evite burbujas de generación espontánea - son burbujas de solo salidas.
- Tener cuidado con los flujos y procesos no etiquetados.
- Tener cuidado con los almacenes de solo escritura o solo lectura.
- Todo almacén debe tener, tanto entradas como salidas, excepto, el almacén externo que sirve de interfaz entre el sistema y alguna entidad externa.
- Restringir un solo DFD a no más de seis u ocho procesos.
- Se debe usar una página para un DFD en particular.

De los Diagramas de Flujos de Datos se puede obtener un Diccionario de Datos, que no es nada más que el reflejo de todos los datos que viajan por los flujos de datos especificados en los diagramas. Para entender el Diccionario de Datos es necesario tener presente la siguiente simbología:

- Dato opcional ‘ (nombre o descripción del dato)’
- Dato de Iteración ‘{nombre o descripción del dato }’
- Dato identificador ‘@ nombre o descripción del dato’
- Dato de selección ‘[opción 1 | opción |.....]’
- Comentarios ‘*comentario*’
- Está compuesto de ‘ = ’
- Adicionado a (y) ‘ + ’

ANEXO F

Manual de
Instalación

**SOLUCIÓN
PERSONALIZADA DE
OFFICE**

**PARA LAS CARRERAS DE
ELECTRÓNICA**

E.P.N 2006

1. INSTALACIÓN DEL SOFTWARE BASE (Requerido).

En este manual se hará referencia únicamente al proceso de instalación del *Framework .NET V1.1* y de los Ensamblados de Interoperabilidad Primaria .NET de Office 2003, en vista de que cada uno de los programas restantes requeridos, incluyen en sus respectivos CDs de instalación manuales o indicaciones que sirven de guía en el proceso.

1.1. Instalación del Framework .Net

El primer paso en el proceso de instalación del Software Base, es comprobar que el equipo tenga instalada la versión 1.1 del *Framework .NET*, y su respectivo paquete para el lenguaje en español.

Para ello hay que ingresar por el menú: **Inicio** → **Panel de Control** → **Agregar o Quitar Programas**.

Si el *Framework* ya se encuentra instalado, se podrá apreciar en la lista de programas los ítems mostrados y resaltados en la Ilustración 1-1.

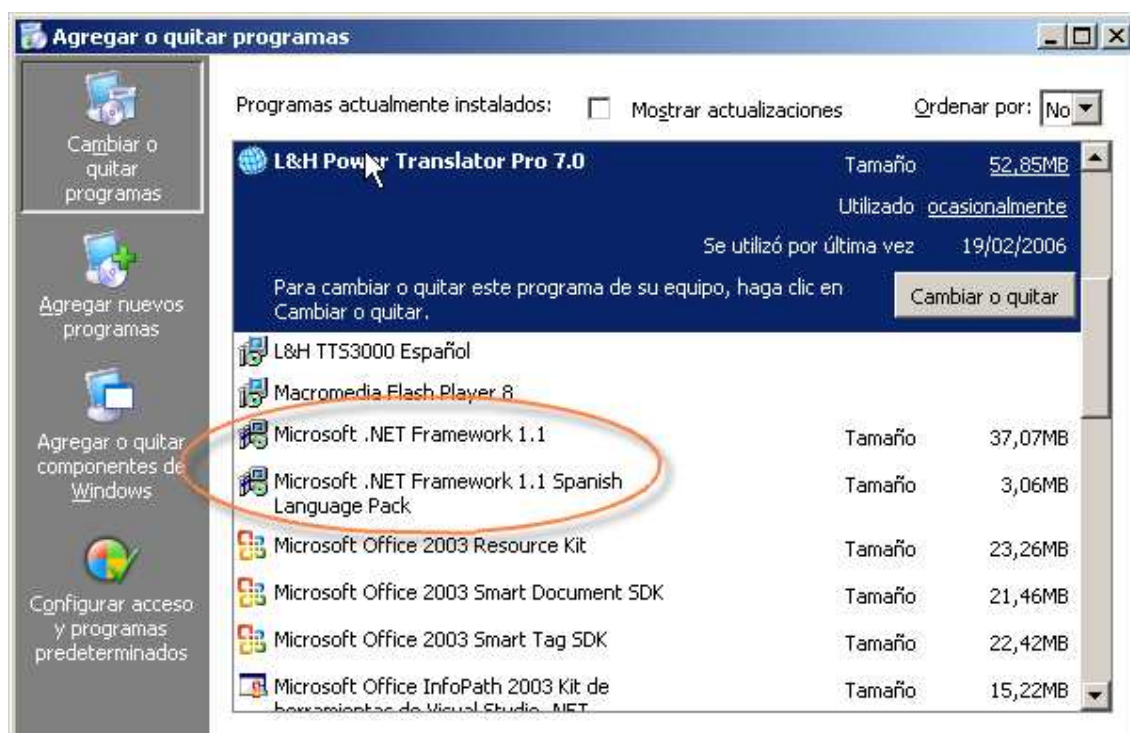


Ilustración 1-1 Componentes .NET que deben estar instalados en el equipo.

En caso de que los elementos señalados en la Ilustración 1-1 no aparezcan en la lista, en el CD de instalación de la SPO bajo el directorio (carpeta) **DotNetFramework**, se encuentran los instaladores tanto del *Framework* .NET V1.1 como del Paquete del Lenguaje en Español (**dotnetfx.exe** y **langpack.exe** respectivamente), ambos son ejecutables con indicaciones paso a paso que le guiarán durante el proceso de instalación.

1.2. Instalación de los Ensamblados de Interoperabilidad Primaria .NET

Es necesario utilizar el CD de instalación de Office 2003 para agregar los Ensamblados de Interoperabilidad Primaria .NET requeridos por el Documento Inteligente. Se debe realizar una **Instalación Personalizada**, empleando **Personalización avanzada de aplicaciones**, como se puede apreciar en la Ilustración 1-2.

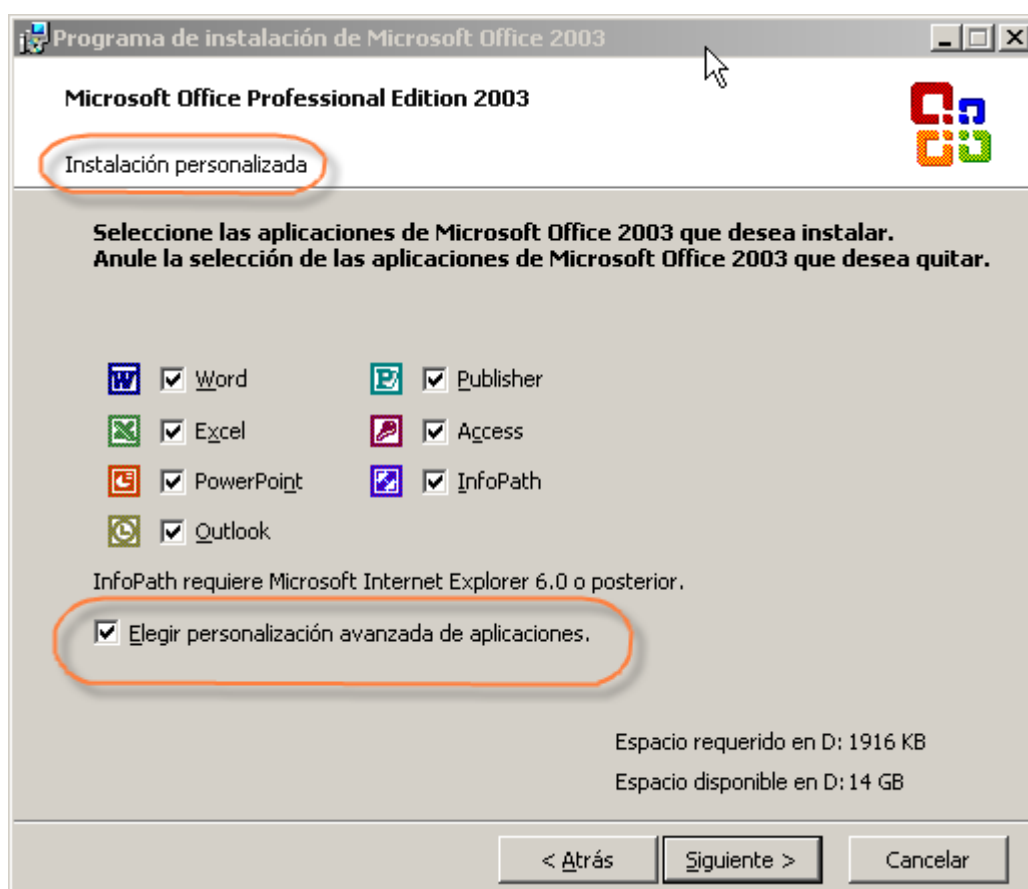


Ilustración 1-2 Cuadro de diálogo para la instalación de personalizada de Office 2003.

La opción de **Personalización avanzada de aplicaciones** permite seleccionar cada uno de los componentes del Office que se desean instalar en el equipo.

Los siguientes componentes del Office 2003 son necesarios:

- .NET Programmability Support
- Microsoft Forms 2.0 .NET Programmability Support
- Smart Tag Programmability Support
- .NET Programmability Support (dentro de Microsoft Graph)
- Smart Tag Plugins

La ubicación de estos componentes dentro de la lista, se puede apreciar más fácilmente en la Ilustración 1-3.

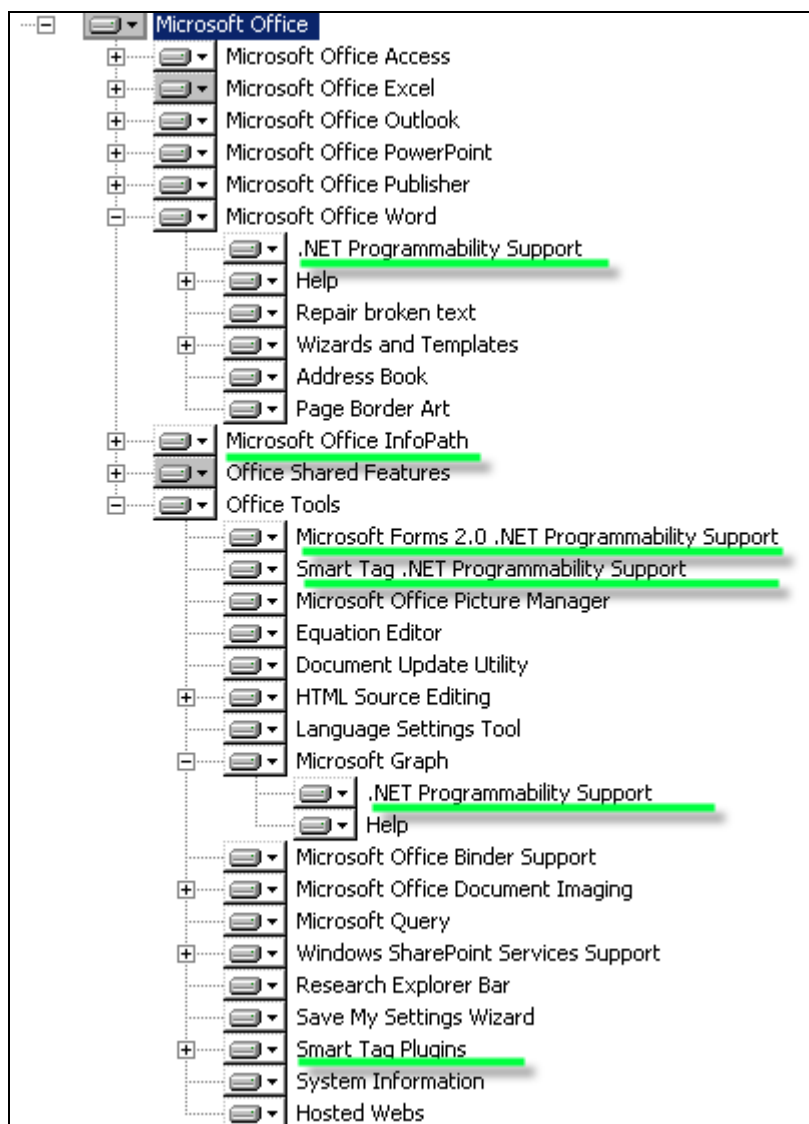


Ilustración 1-3 Componentes de Office 2003 que deben ser instalados en el equipo.

Para instalar los componentes en el equipo se debe hacer clic sobre el ícono del componente que se desea instalar y seleccionar la opción **Ejecutar desde mi computador** (*Run from My Computer*), como se muestra en la Ilustración 1-4.

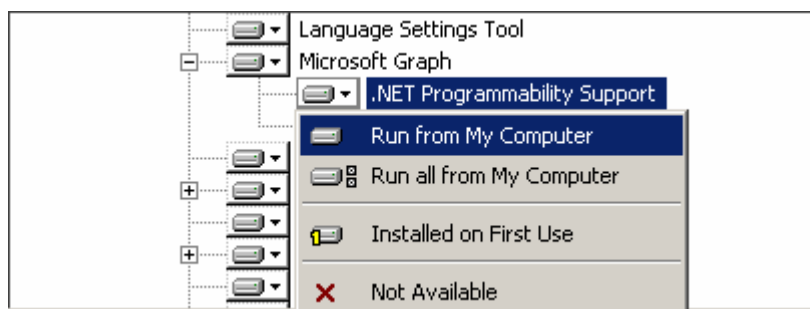


Ilustración 1-4 Menú contextual del instalador del Office 2003.

2. INSTALACIÓN DEL DOCUMENTO INTELIGENTE DE MICROSOFT WORD.

2.1. Instalación de la Base de Datos

Dentro del CD de instalación bajo el directorio SPODB se encuentran los archivos de la Base de Datos **SPODB_Data.MDF** y **SPODB_Log.LDF**, requeridos para instalar o “Adjuntar la SPODB” al Servidor de SQL que debe estar instalado previamente en el equipo.

Para ello en la ventana del **Administrador Corporativo** (*Enterprise Manager*), en la sección **Bases de Datos** (*Databases*), se debe hacer clic derecho y aparecerá un menú contextual con la opción “**Attach Database**” como el mostrado en la Ilustración 2-1.

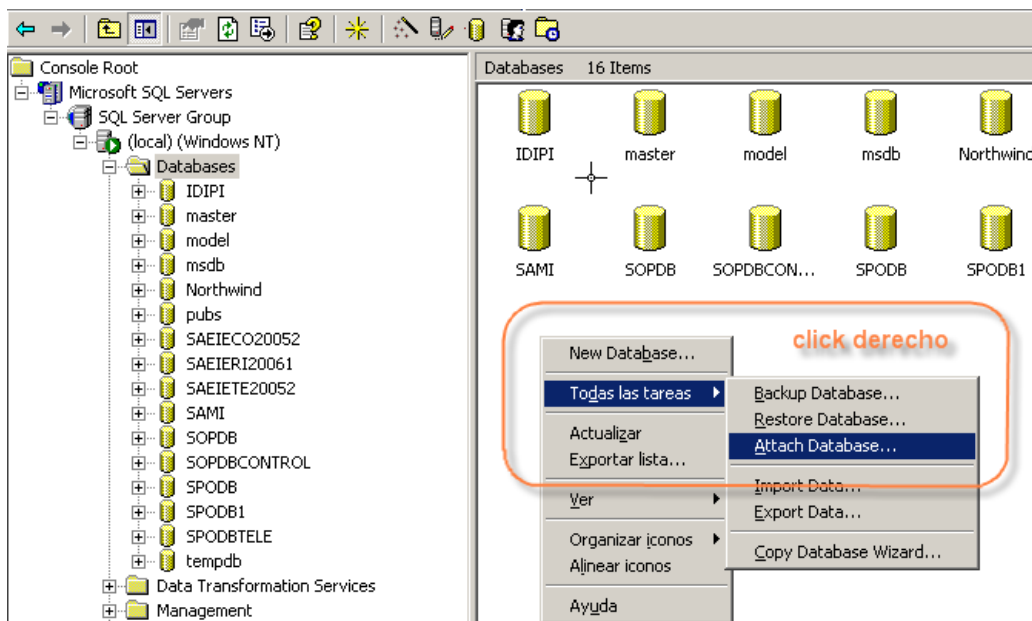


Ilustración 2-1 Ventana Enterprise Manager de SQL Server sección Databases.

Una vez seleccionada la opción “**Attach Database**”, aparecerá en pantalla un cuadro de diálogo que permitirá seleccionar el *Path* del archivo **SPODB_Data.MDF** del CD de instalación (imprescindible para poder adjuntar la base de datos), y un combo box que permitirá especificar el usuario propietario de la nueva base de datos, tal como se puede apreciar en la Ilustración 2-2.

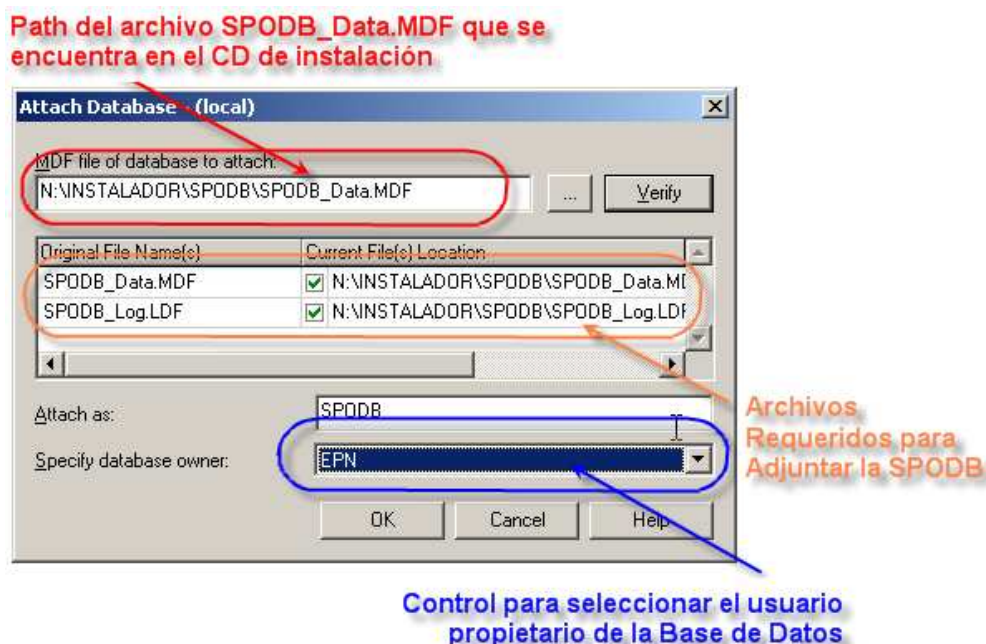


Ilustración 2-2 Cuadro de Diálogo para Adjuntar la Base de Datos en SQL Server.

Luego de hacer clic en el botón **OK** del cuadro de diálogo mostrado en la Ilustración 2-2 si no se presentan errores; tales como, nombre duplicado, falta de espacio en disco, etc., la base de datos se adjuntará con éxito y estará lista para ser utilizada.

2.2. Instalación del Certificado Digital

En el CD de instalación se ha incluido un archivo denominado **CertificadoDI.cer**, éste es el certificado digital requerido para el correcto funcionamiento del Documento Inteligente, para instalar el mismo hay que hacer doble clic sobre el archivo, con lo cual aparecerá la siguiente pantalla (Ilustración 2-3).

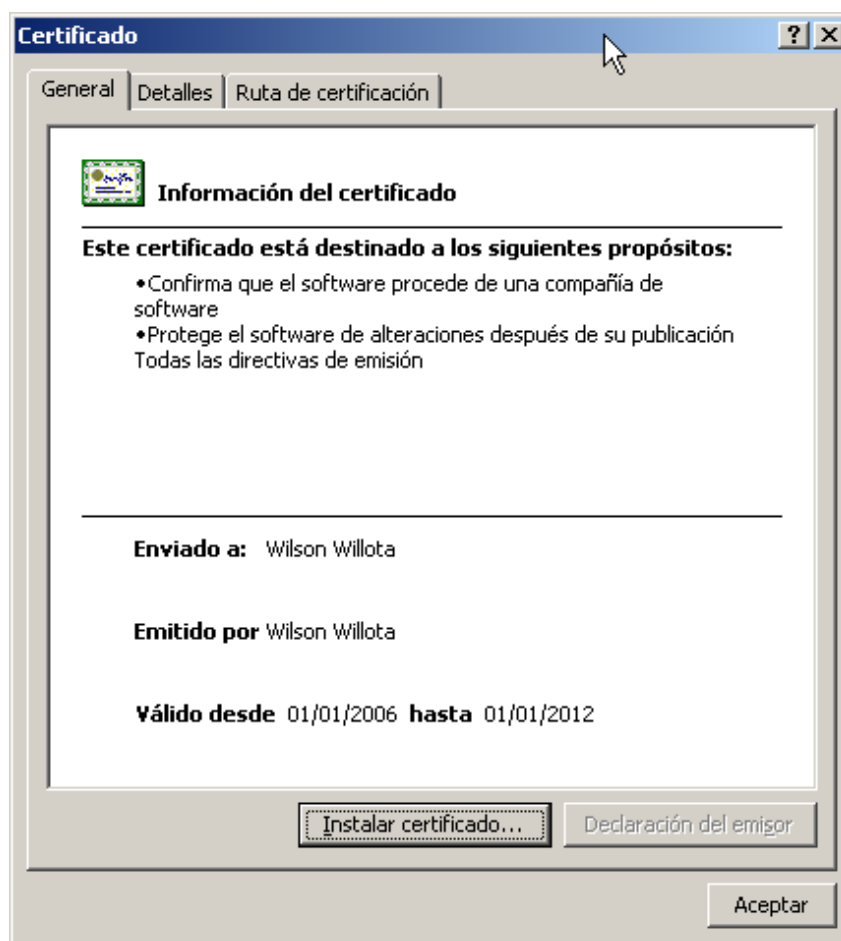


Ilustración 2-3 Pantalla principal del Certificado Digital.

Al hacer clic sobre el botón **Instalar Certificado**, aparecerá un Asistente (Ilustración 2-4).

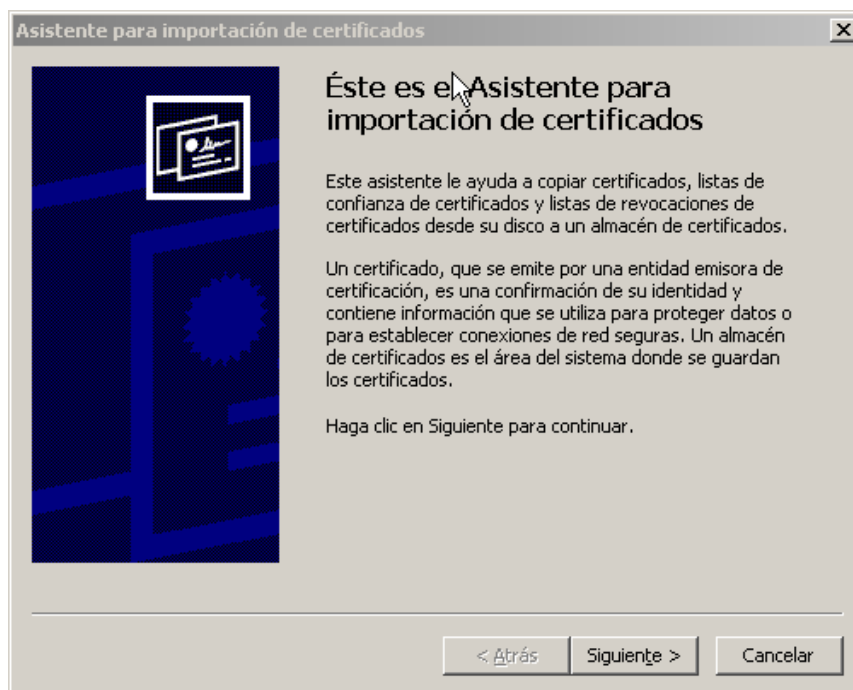


Ilustración 2-4 Asistente para importación de certificados.

El asistente permitirá seleccionar el **Almacén de certificados** donde se desea realizar la instalación, para ello hay que utilizar la opción **Colocar todos los certificados en el siguiente almacén**, como se aprecia en la Ilustración 2-5.

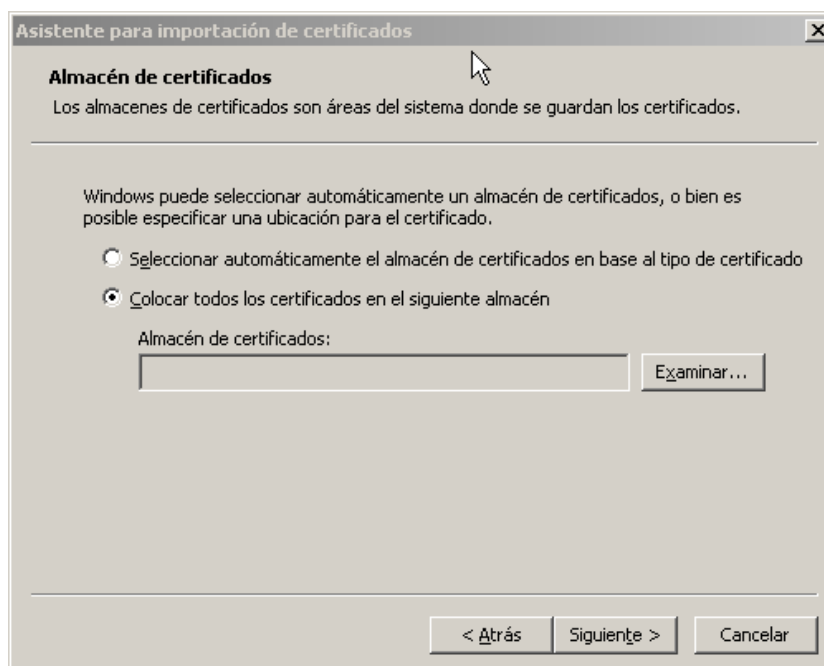


Ilustración 2-5 Cuadro de diálogo para la selección del Almacén de certificados.

El almacén de certificados que debe ser seleccionado es: “**Entidades emisoras raíz de confianza**”, para ello hay que hacer un clic en el botón Examinar (Ilustración 2-5) con lo cual aparecerá el cuadro de diálogo **Seleccionar almacén de certificados** (Ilustración 2-6).

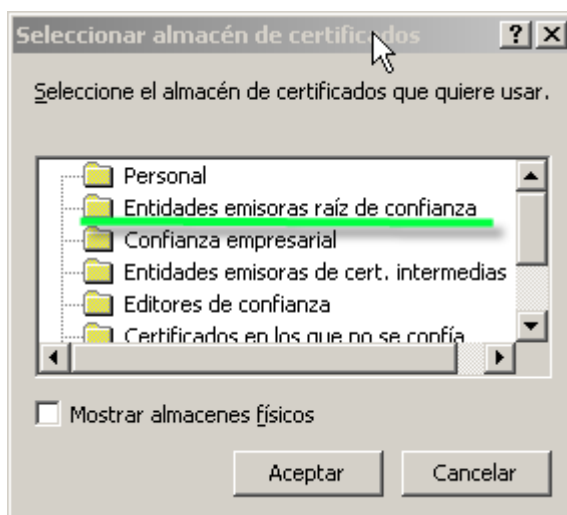


Ilustración 2-6 Cuadro de diálogo Seleccionar almacén de certificados.

Finalmente se debe dar clic en el botón **Siguiente** (Ilustración 2-5), una vez finalizada la instalación (importación) del certificado digital aparecerá el siguiente mensaje (Ilustración 2-7).

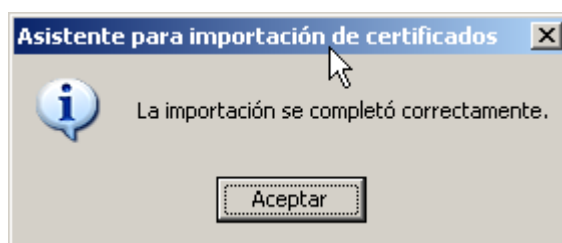


Ilustración 2-7 Mensaje final de importación del certificado.

2.3. Configuración del Nombre del Servidor a ser Utilizado

Antes de realizar la instalación del documento Inteligente es necesario configurar el nombre del Servidor de la Base de Datos que será utilizado, para ello en el CD de instalación bajo el Directorio **INSTALADOR**→**ARCHIVOS**→**CONFIGURACIÓN** se encontrará el archivo **ConfigDB.xml** tal como se puede apreciar en la Ilustración 2-8.



Ilustración 2-8 Directorio para encontrar el archivo de configuración de la Base de Datos.

Al editar el archivo **ConfigDB.xml** con el Bloc de notas se podrá observar el texto presentado a continuación (Ilustración 2-9).

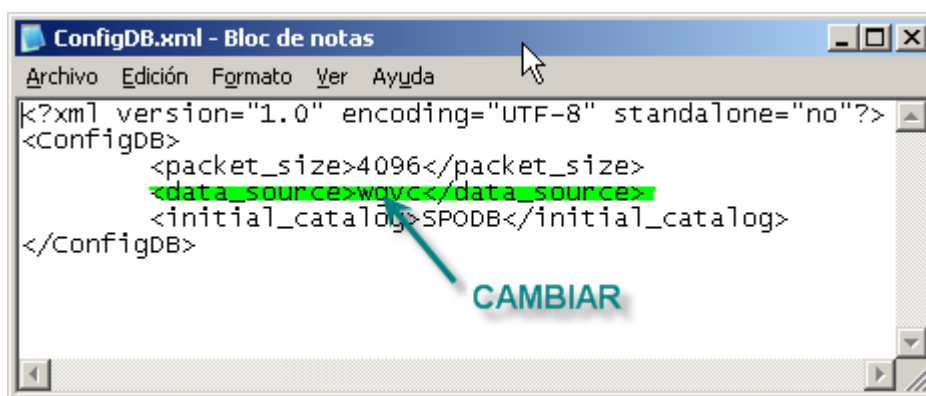


Ilustración 2-9 Texto del Archivo de Configuración de la Base de Datos

El elemento a ser editado es **<data_source></data_source>** que para el caso del ejemplo contiene el valor **wgvc** (Ilustración 2-9), este elemento debe contener el nombre del Servidor donde se encuentra alojada la base de datos SPODB.

En caso de no saber el nombre del servidor, se puede recurrir al Administrador de Servicios (*Service Manager*) de SQL Server, al que se accede por el menú **Inicio**→**Todos los programas**→**Microsoft SQL Server**→**Service Manager** ahí se encontrará un combo box (Ilustración 2-10) que contiene el nombre del Servidor de base de datos que se encuentra en ejecución.

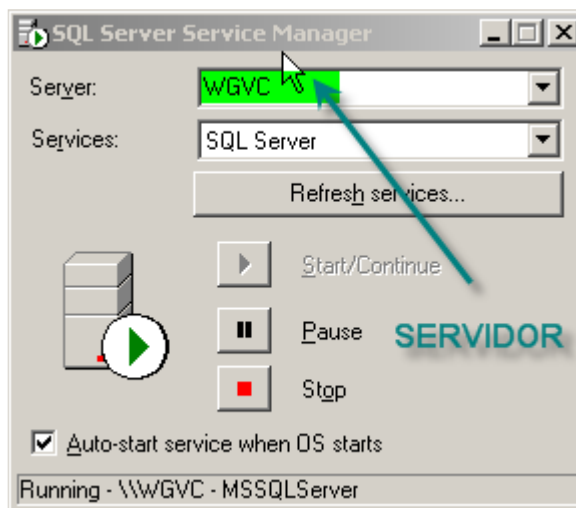


Ilustración 2-10 Pantalla del administrador del Servicio de SQL Server.

2.4. Configuración de las Directivas de Seguridad .NET

En el CD de instalación se ha incluido el archivo **ConfigurarSeguridad.cmd** el mismo que deberá ser ejecutado (doble clic) para configurar las directivas de seguridad .NET.

Aunque no es indispensable, se puede verificar que la directiva de seguridad ha sido agregada correctamente ingresando por **Panel de Control**→**Herramientas Administrativas**→**Configuración de Microsoft .NET Framework 1.1**

Se abrirá la consola de configuración de .NET, en donde se debe revisar **Directivas de seguridad en tiempo de ejecución** → **Usuario** → **Grupos de código** → **All_Code** → **DOCUMENTO INTELIGENTE EPN** como se puede apreciar en la Ilustración 2-11, si este elemento existe, es porque la directiva fue agregada correctamente.

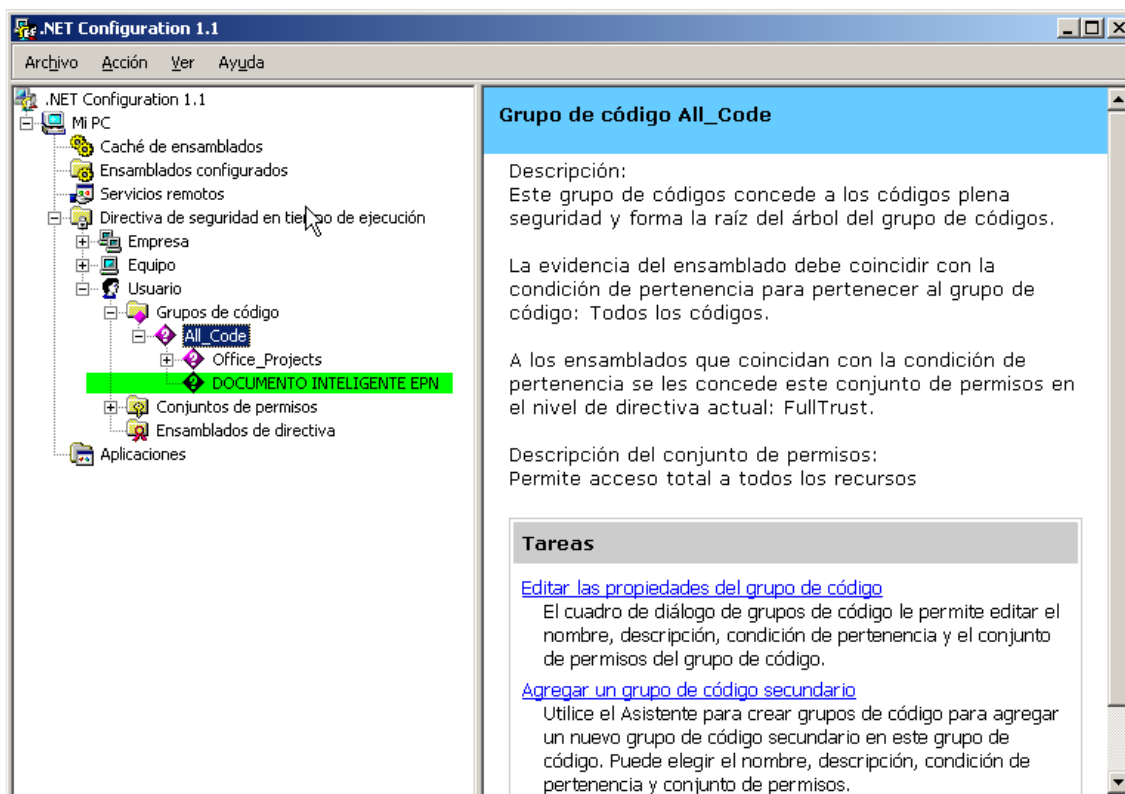


Ilustración 2-11 Pantalla de configuración del .NET Framework.

2.5. Instalación del Paquete de Expansión

- Abrir un nuevo documento en blanco de Microsoft Office Word 2003.
- En el menú **Herramientas**, seleccionar la opción **Plantillas y complementos** (Ilustración 2-12).
- Aparecerá el cuadro de diálogo **Plantillas y complementos**, en donde se debe seleccionar la opción **Paquetes de expansión XML** (Ilustración 2-13).
- Hacer clic en el botón Agregar (Ilustración 2-13), y seleccionar el archivo **ProyectosTitElectrónica.xml** del CD de instalación.
- Una vez seleccionado el archivo, comenzará el proceso de instalación y se presentará un cuadro de diálogo similar al mostrado en la Ilustración 2-15.

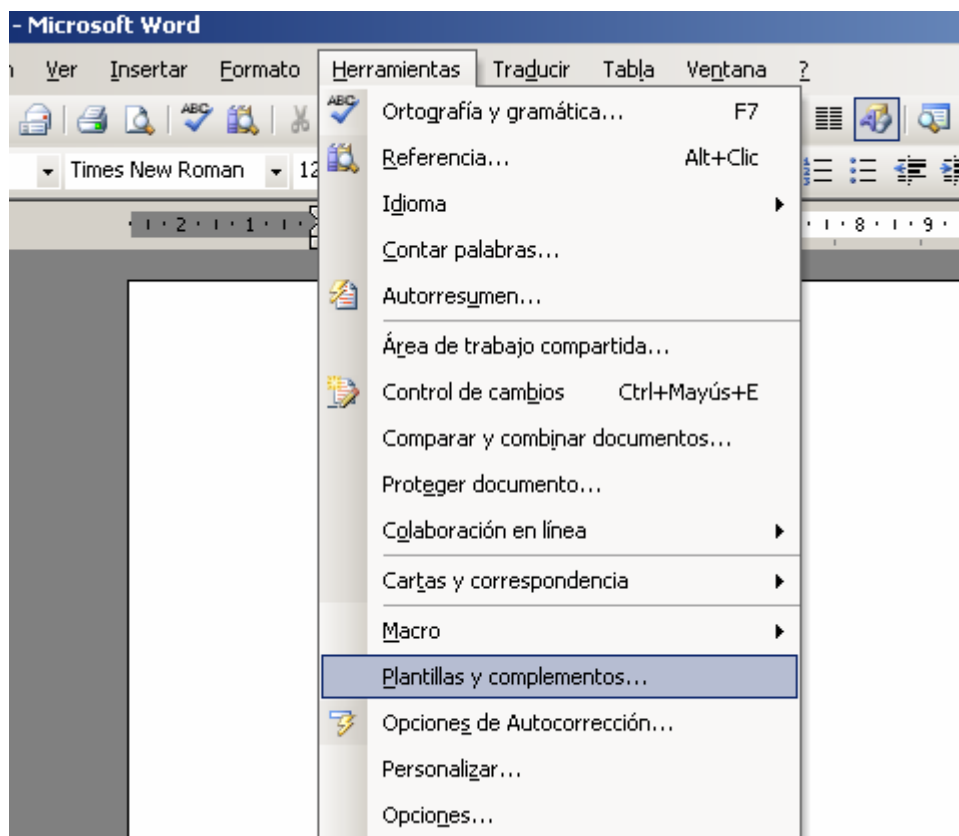


Ilustración 2-12 Menú Herramientas de MS Word 2003.

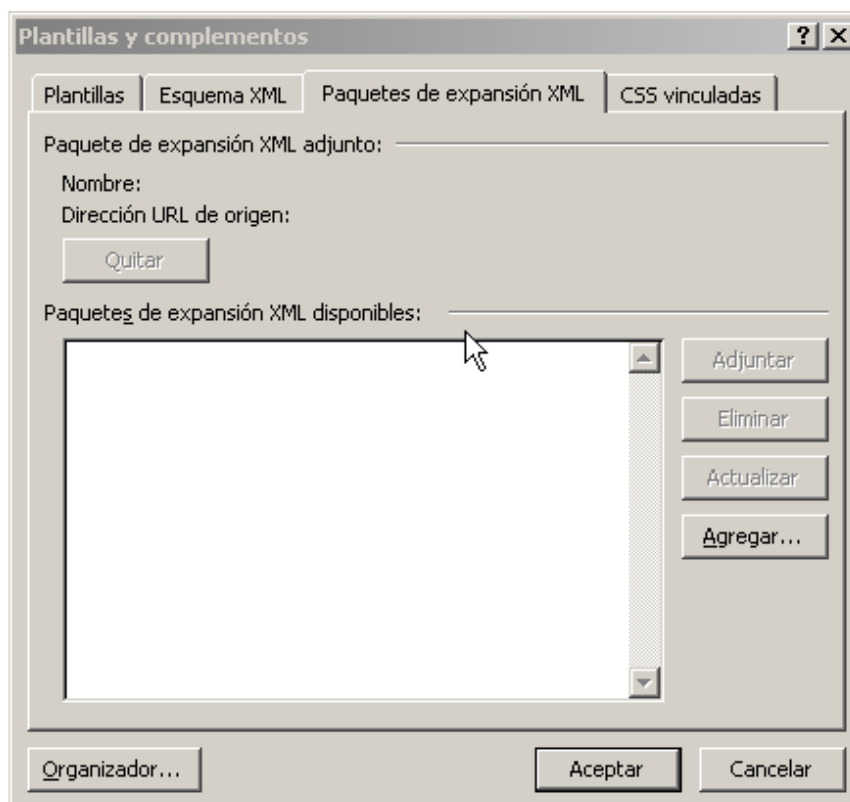


Ilustración 2-13 Cuadro de diálogo Plantillas y complementos opción Paquetes de Expansión XML.

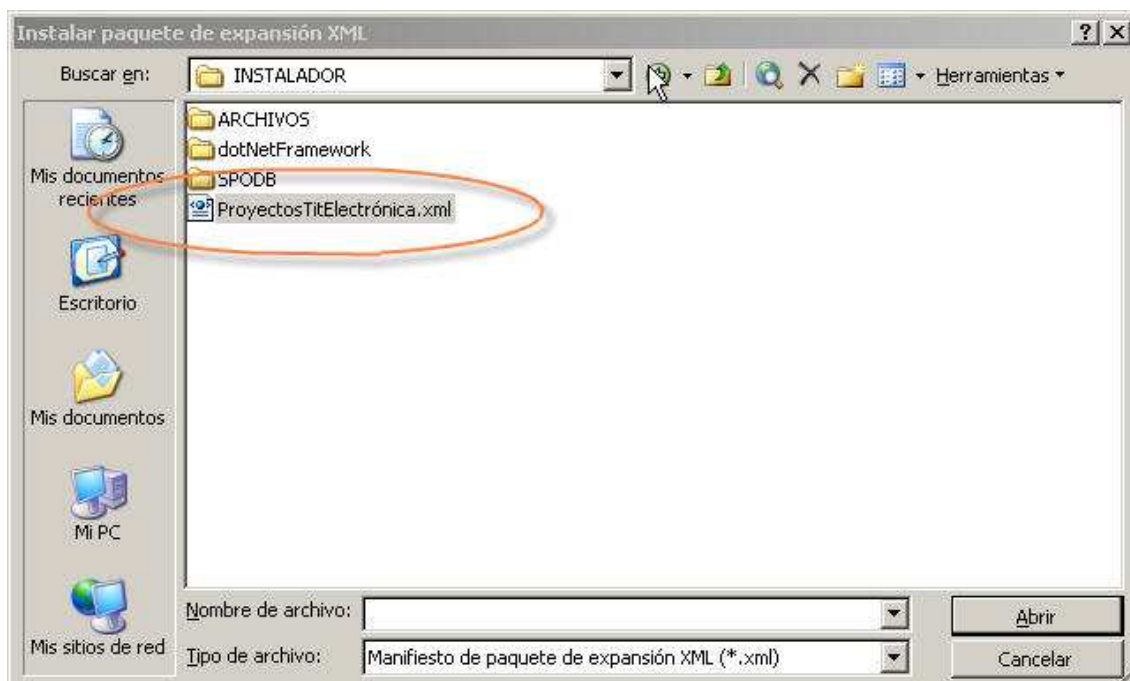


Ilustración 2-14 Archivo de Manifiesto XML para el Paquete de Expansión XML.

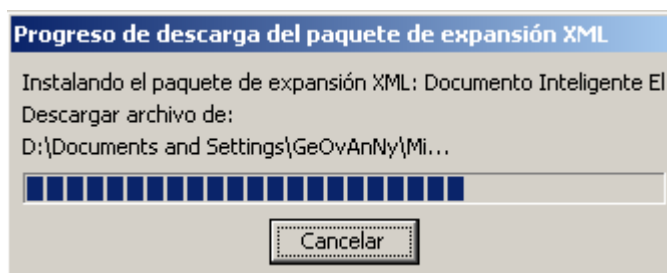


Ilustración 2-15 Cuadro de diálogo del Proceso de descarga del paquete de expansión.

- Si el proceso de instalación se completó exitosamente, se deben verificar tres cambios fundamentales:
 - Entre los paquetes de expansión disponibles debe encontrarse **Documento Inteligente Electrónica EPN** (Ilustración 2-16).
 - Se debe haber adjuntado como Esquema XML: **Esquema Documentos Electrónica EPN** (Ilustración 2-17)
 - Se debe haber cambiado la plantilla del documento (Ilustración 2-18)

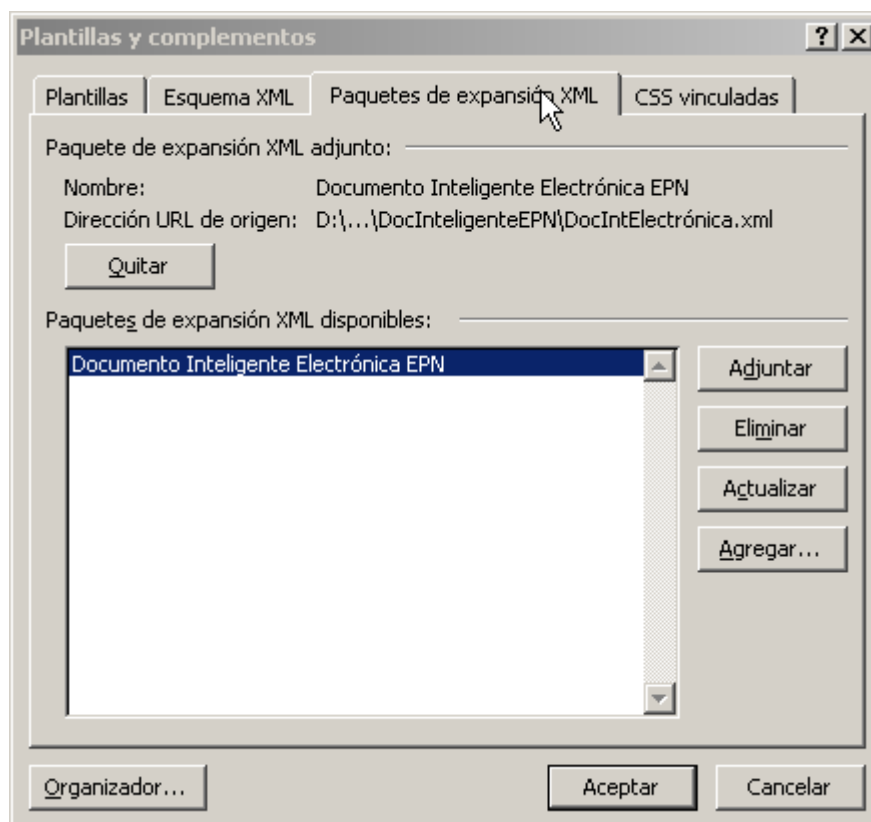


Ilustración 2-16 Cuadro de diálogo Plantillas y complementos luego de la descarga del Paquete de Expansión.

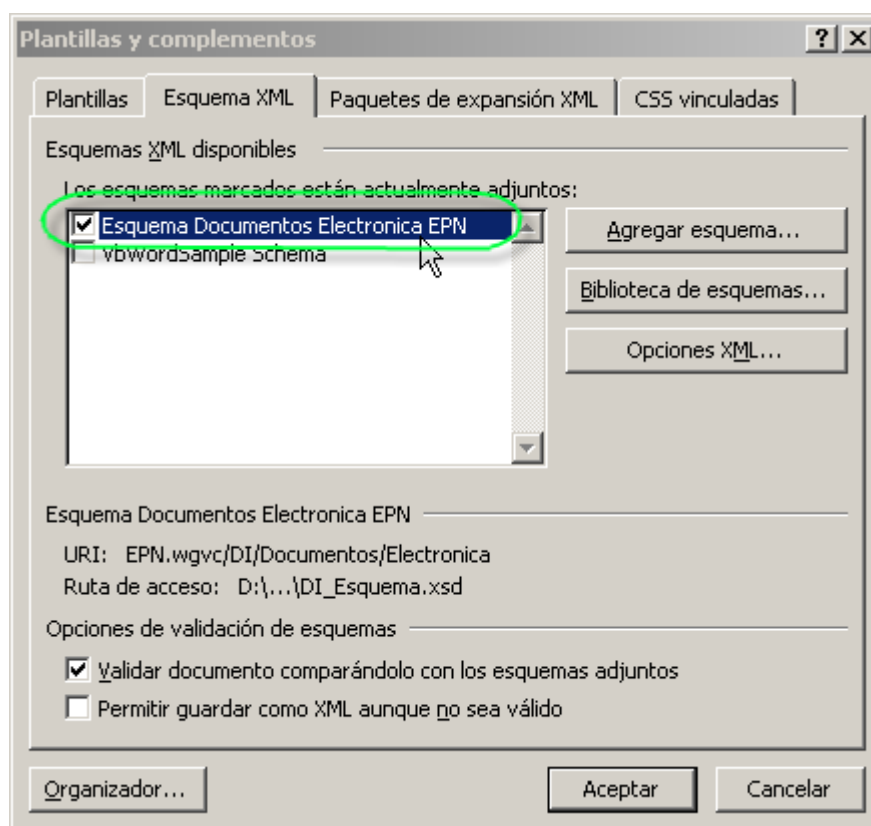


Ilustración 2-17 Esquema XML adjuntado al documento.

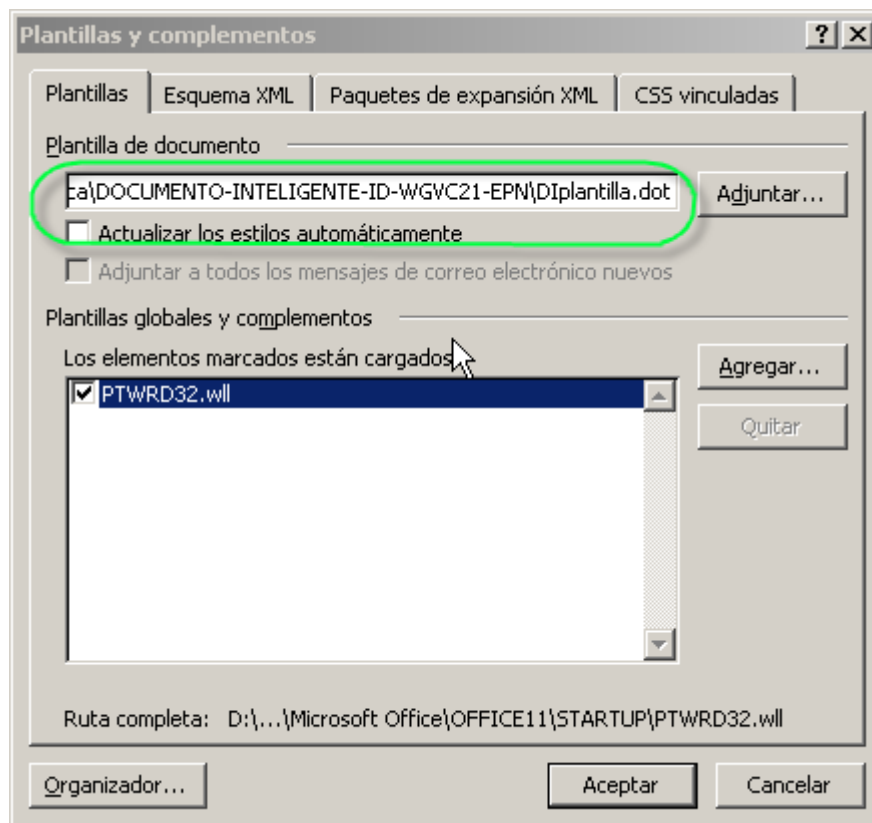


Ilustración 2-18 La plantilla (archivo .dot) asociado al documento.

- En caso de que el certificado digital no se haya instalado o exista algún problema de reconocimiento del mismo, aparecerá un mensaje como el de la Ilustración 2-19, por lo cual habrá que repetir el proceso de instalación del certificado.

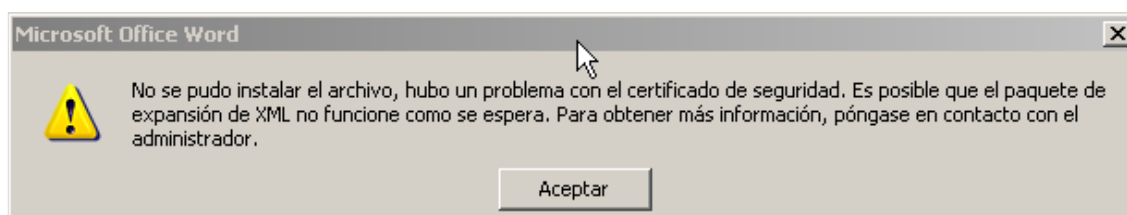


Ilustración 2-19 Mensaje de aviso para errores con el certificado digital.

- Luego de completado el proceso, el Documento Inteligente estará listo para ser utilizado, y aparecerán el Panel de Tareas de Acción y la Barra de Herramientas personalizada como se puede apreciar a en la Ilustración 2-20.

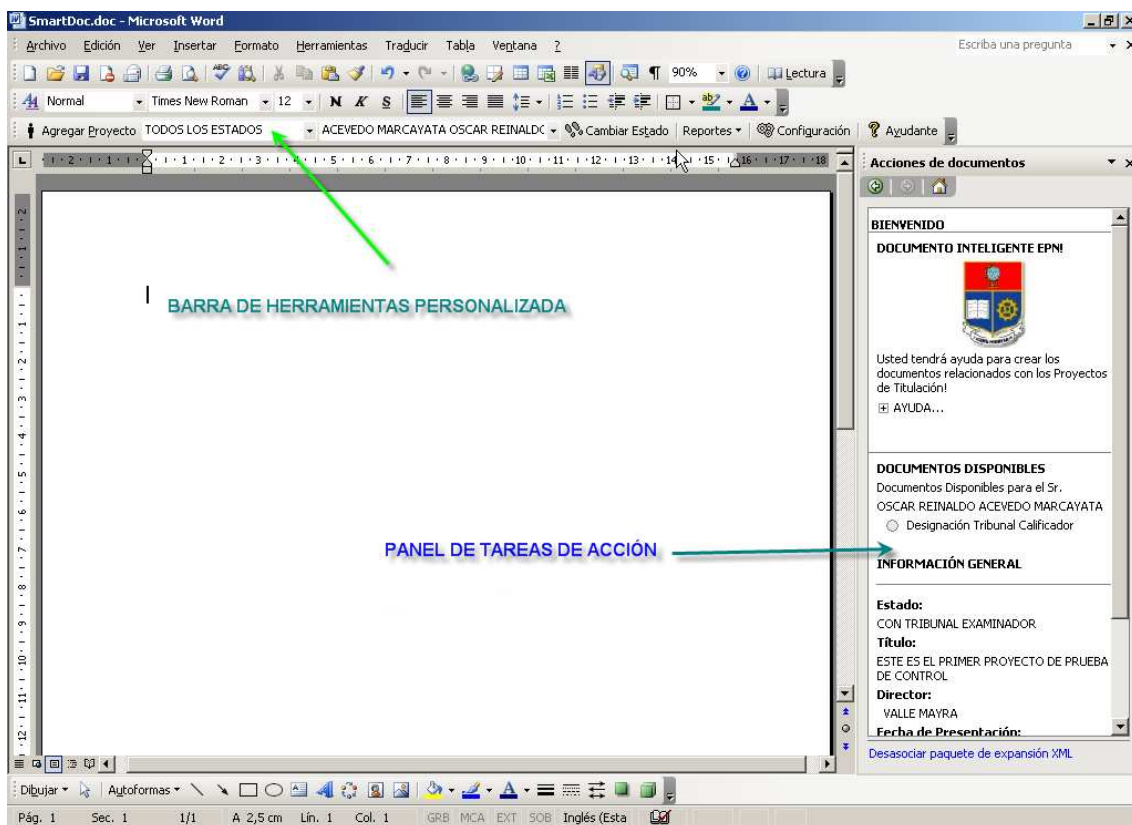


Ilustración 2-20 Interfaz de usuario del Documento Inteligente.

ANEXO G

**Manual de
Usuario**

**SOLUCIÓN
PERSONALIZADA DE
OFFICE**

**PARA LAS CARRERAS DE
ELECTRÓNICA**

E.P.N 2006

1. ADMINISTRACIÓN DE LOS PPTs

La Administración de los PPTs es una función que engloba varias otras funciones, las mismas que podrán ser ejecutadas desde el Documento Inteligente, y como un adicional si las secretarías así lo desean, para agregar las observaciones realizadas en una sesión de Subcomisión Académica podrán emplear el Formulario de InfoPath como se indica más adelante.

Todas las funciones de la SPO utilizarán una conexión directa a la Base de Datos Local con sus respectivos permisos, por esta razón a menos que explícitamente se indique algo diferente, su utilización deberá estar sobrentendida.

1.1. Sincronización con la Información del SAE

Esta función estará disponible a través de un formulario de Windows denominado **Formulario de Configuración** el mismo que aparecerá al presionar el botón de la barra de herramientas personalizada del Documento Inteligente, que se muestra en la Ilustración 1-1.

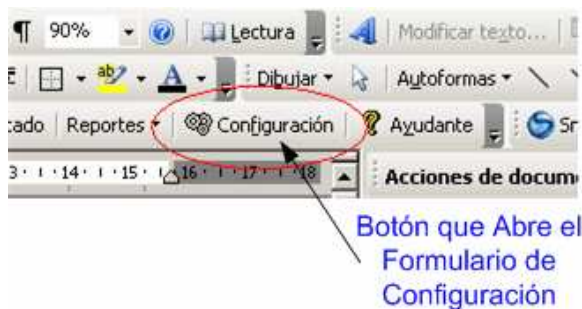


Ilustración 1-1 Barra de herramientas personalizada Documento Inteligente.

En el **Formulario de Configuración**, existirá una pestaña denominada **Sincronizar Datos SAE**, la misma que contendrá un solo botón, tal como se muestra en la Ilustración 1-2; su ejecución resulta en la sincronización de las bases de datos SAE y SPODB.

Ésta debería ser la primera función a ser ejecutada por los usuarios para garantizar la disponibilidad de todos los datos; al presionar el botón

SINCRONIZAR de la Ilustración 1-2, se copiarán los datos necesarios que contenga la base de datos SAE y que no contenga la SPODB.



Ilustración 1-2 "Formulario de Configuración" opción "Sincronizar Datos SAE".

La función de sincronización requerirá el ingreso de un nombre de usuario y contraseña. En condiciones normales esta función se realizaría una sola vez por semestre luego de reinscripciones; sin embargo, los usuarios no tienen restricciones para ejecutarla siempre que lo consideren necesario.

1.2. Ingreso de Nuevos PPTs

Al igual que la función anterior, ésta también se realizará utilizando un formulario de Windows a manera de cuadro de diálogo; el botón de la barra de herramientas personalizada que permitirá presentar el mencionado formulario, tendrá el nombre de **Agregar Proyecto** y se puede apreciar a continuación.

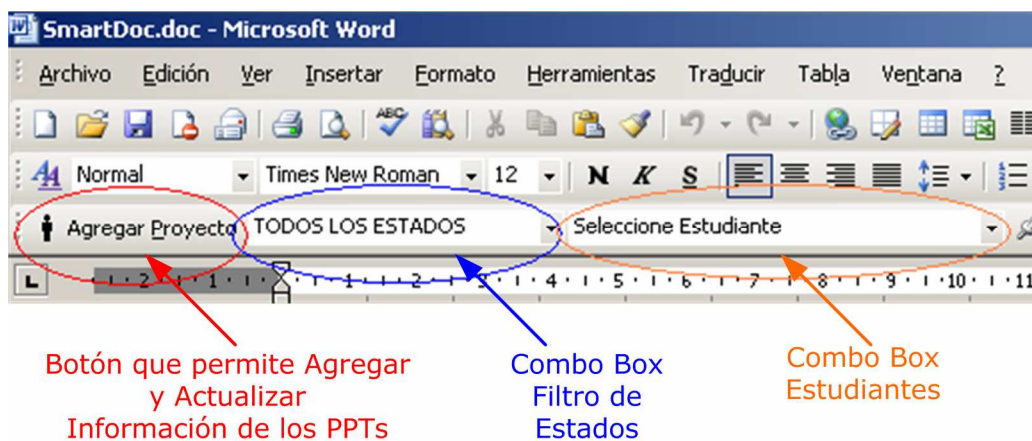


Ilustración 1-3 Barra de herramientas personalizada Documento Inteligente.

- **Validación General:** existen varios datos que no pueden ser nulos, y esto es lo que se comprueba en una validación general, que se realizará en el momento que se desee almacenar la información, en este caso los errores se presentarán en un mensaje de error como el que se presenta en la siguiente ilustración.

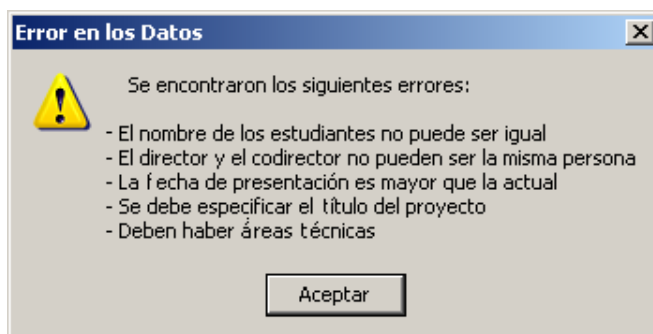


Ilustración 1-4 Mensaje de error en los datos.

- **Validación instantánea:** comprueba que la información ingresada cumpla con los siguientes requerimientos elementales: el nombre del director y del codirector del proyecto deben ser diferentes; en caso de ser un proyecto de dos estudiante, los nombres de los mismos también deben ser distintos; la fecha de presentación del plan de proyecto debe ser menor o igual que la fecha en la que se está creando el registro. Los errores de validación instantánea se presentarán inmediatamente luego de ingresado un valor, y serán identificados por un icono junto al control en el que se presenta el inconveniente, como se puede apreciar en la Ilustración 1-5.

Agregar Proyecto

PLAN DE PROYECTO

Agregar Nuevo Plan Actualizar Existente

PROPUESTO POR
 Un estudiante Dos Estudiantes
 DAQUI SOLANO OSWALDO FERNANI ok
 DAQUI SOLANO OSWALDO FERNANI !

DIRECTOR Y CODIRECTOR DEL PROYECTO
 Proyecto con Codirector
 ALDAZ OSWALDO ok
 ALDAZ OSWALDO !

TEMA O TITULO DEL PROYECTO
 El nombre del director no puede ser igual al del codirector

AREAS TECNICAS DEL TEMA
 Buscar: redes
 REDES DE AREA EXTENDIDA
 REDES DE AREA LOCAL
 REDES DE ÁREA LOCAL INALAMBRIC
 BASES DE DATOS
 El nombre de los estudiantes no puede ser igual

FECHA PRESENTACION
 jueves , 09 de marzo de 2006 !
 La Fecha de Presentación no puede ser mayor que la actual

Reset **Grabar** **Salir**

Borra el contenido actual de los controles Almacena la información en la SPODB Cierra el formulario

Ilustración 1-5 Validación inmediata de información.

1.3. Actualización de la Información de los PPTs

Para actualizar la información de un PPT se accederá de la misma manera que se utiliza para el ingreso del mismo, sólo que en este caso se utilizará la opción **Actualizar Existente**.

Al seleccionar un estudiante del *Combo Box*, en el formulario se presentará toda la información asociada al plan de proyecto de titulación presentado por dicho estudiante; si el plan es de un proyecto conjunto, seleccionando a cualquiera de los dos estudiantes sería suficiente para la actualización de la información.

Para este caso se aplican las mismas validaciones que en el ingreso, y cambian algunos controles de la interfaz de usuario como se puede observar en la Ilustración 1-6.

El título del formulario cambia al seleccionar la opción Actualizar Existente

Actualizar Proyecto

PLAN DE PROYECTO

Agregar Nuevo Plan Actualizar Existente

PROPUESTO POR

Un estudiante Dos Estudiantes

MOYA LEIMBERG FERNANDO ANDRE

REA GUALOTO VICTOR HUGO

DIRECTOR Y CODIRECTOR DEL PROYECTO

Proyecto con Codirector

SINCHE MAITA SORAYA LUCIA

ÁREAS TÉCNICAS DEL TEMA

Buscar: _____

REDES DE COMPUTADORAS

REDES E INTRANET

REDES TCP/IP

REDES DE ÁREA LOCAL INALAMBRICAS
REDES DE COMPUTADORAS
REDES E INTRANET
REDES TCP/IP
SEGURIDAD EN REDES

FECHA PRESENTACION

viernes , 03 de junio de 2005

TEMA O TITULO DEL PROYECTO

DISEÑO Y SIMULACIÓN DE LA RED DE ACCESO INALÁMBRICO DE BANDA ANCHA PARA PANCHONET S.A.EN LA ZONA DE LOS VALLES CIRCUNDANTES A LA CIUDAD DE QUITO UTILIZANDO LAS TECNOLOGÍAS WIMAX (IEEE 802.16) Y MOBILEFI (802.20)

Eliminar **Actualizar** **Salir**

Elimina el registro de la SPODB

Actualiza la información en la SPODB

Cierra el formulario

Ilustración 1-6 Actualización de información.

También está contemplada la eliminación de los planes de proyecto, en este caso se presentará un cuadro de diálogo de confirmación, como el que se muestra en la Ilustración 1-7.

ELIMINAR PROYECTO

Seguro que desea eliminar este registro?

Ilustración 1-7 Confirmación de eliminación.

1.4. ADMINISTRACIÓN DEL ESTADO DE LOS PPTs Y PTs

Para poder ingresar ciertos datos asociados a un proyecto, éste primero tiene que encontrarse en un determinado estado, la barra de herramientas personalizada contendrá dos controles del tipo *Combo Box*, el uno permitirá seleccionar el estado de los proyectos, y el otro permitirá seleccionar el nombre de los estudiantes.

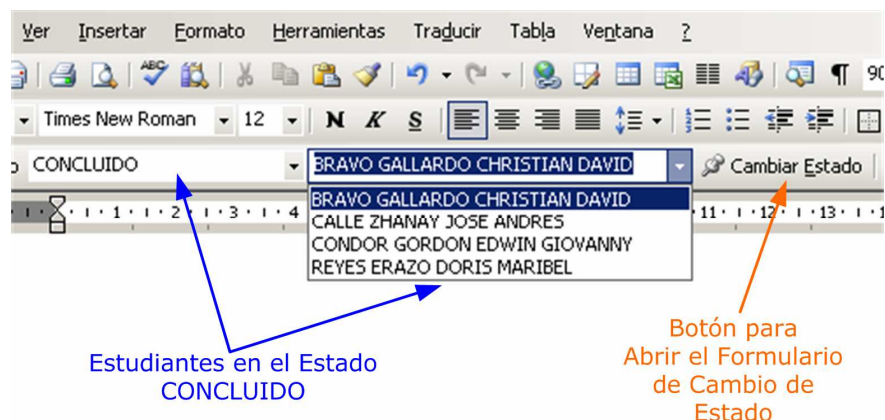


Ilustración 1-8 Barra de herramienta personalizada Documento Inteligente.


El *Combo Box* para los estados es simplemente una ayuda (filtro), que permite listar en el *Combo Box* de estudiantes únicamente aquellos cuyo proyecto de titulación se encuentre en el estado seleccionado, un ejemplo se muestra en la Ilustración 1-8.

En el *Combo Box* para los estados también existirá la opción **Todos los Estados**, con la cual no se aplica ningún filtro y se listan todos los estudiantes en orden alfabético.

Al utilizar la opción **Todos los Estados**, el usuario no necesita recordar el estado en el que se encuentra el proyecto de un determinado estudiante, pero en el caso de recordarlo el filtro de estados le permitirá localizarlo más rápidamente.

Una vez seleccionado un estudiante, se puede presionar el botón **Cambiar Estado** de la barra de herramientas [Ilustración 1-8], con lo cual aparecerá un formulario de Windows denominado **Cambio de Estados**; este formulario contendrá diferentes combinaciones de controles que permitirán que el usuario

únicamente ingrese los datos requeridos según el estado del proyecto. En las figuras de la siguiente Tabla se presentan todas las posibles combinaciones de controles indicando el estado al que pertenecen.

	<p>Del estado “En Aprobación” es posible cambiar a los estados “Negado”, “Aprobado” o “Delegado”. En el caso de seleccionar el estado “Aprobado” se solicitará la fecha de aprobación, en el caso del estado “Delegado” se solicitará la fecha de la delegación, y para el estado “Negado” solo se solicita una confirmación.</p>
<p>En Aprobación</p>	
	<p>Del estado “Delegado a Coordinador” se puede cambiar al estado “Negado” o al estado “Aprobado”. Para el estado “Aprobado” se solicitará la fecha de la aprobación, y para el estado negado al igual que el estado anterior solo será necesaria una confirmación en el momento de presionar el botón Cambiar.</p>
<p>Delegado a Coordinador</p>	

Cambio de Estados

CAMBIO DEL ESTADO DE LOS PROYECTOS

Estado Actual: APROBADO/EN DESARROLLO

Estudiantes: CADENA PINARGOTE ALVARO SANTIAGO

Estados Posibles:

- NEGADO/ABANDONADO
- CON TRIBUNAL EXAMINADOR

Datos Requeridos: Tribunal Examinador

Fecha Delegación Tribunal: 08 de mar de 2006

ALMEIDA CARLOS

ANDRADE JAIME

DISEÑO Y PRUEBAS DE CAMPO DE UNA RED LAN INALÁMBRICA PARA ELÉCTRICA QUITO S.A EMPLEADO EL ESTÁNDAR IEEE802.11g

PRESENTADO POR:
ALVARO SANTIAGO CADENA PINARGOTE

marzo de 2006

lun	mar	mié	jue	vie	sáb	dom
27	28	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Hoy: 08/03/2006

Reset Cambiar Salir

Aprobado

Del estado "Aprobado" todavía es posible cambiar al estado "Negado", y también se podrá cambiar al estado "Con Tribunal Examinador", los datos que se necesitan para esta transición son los nombres de los miembros del Tribunal Examinador y la fecha de su delegación.

Cambio de Estados

CAMBIO DEL ESTADO DE LOS PROYECTOS

Estado Actual: CON TRIBUNAL EXAMINADOR

Estudiantes: ARROBA PEÑAFIEL JOSE RAUL

Estados Posibles:

- APTO PARA DEFENSA ORAL

Datos Requeridos: Tribunal Notas Trabajo Escrito

Nota Ing. Xavier Armendariz: 10,00

Nota Ing. Oswaldo Aldaz: 10,00

Nota Ing. Renato Andrade: 09,50

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE UNA INTRANET/EXTRANET PARA EL CENTRO MÉDICO VITALSERVIS

PRESENTADO POR:
JOSE RAUL ARROBA PEÑAFIEL

Reset Cambiar Salir

Con Tribunal Examinador

Del estado "Con Tribunal Examinador", se puede cambiar al estado "Apto para Defensa Oral", y el requisito son las notas del trabajo escrito, tanto del director como de los miembros del Tribunal Examinador. En este caso en el formulario ya se presentan los nombres de los profesores.

Cambio de Estados

CAMBIO DEL ESTADO DE LOS PROYECTOS

Estado Actual: APTO PARA DEFENSA ORAL

Estudiantes: Seleccione Estudiante

Estados Posibles:

- CON TRIBUNAL DEFENSA ORAL

Datos Requeridos:

Seleccione Presidente

Fecha de la Defensa Oral: 08/03/2006

Hora de la Defensa Oral: 0:00:00

Datos Requeridos: Tribunal Defensa Oral

Seleccione Miembro1 Tribunal Defensa Oral

Seleccione Miembro2 Tribunal Defensa Oral

Reset Cambiar Salir

Apto para Defensa Oral

Del estado "Apto para Defensa Oral" se puede cambiar al estado "Con Tribunal Defensa Oral", se requiere el nombre del presidente, la fecha y hora de la defensa y los nombres de los miembros del Tribunal para la Defensa Oral. De manera predeterminada se presentarán los nombres de los miembros del Tribunal Examinador del trabajo Escrito.

Cambio de Estados	
CAMBIO DEL ESTADO DE LOS PROYECTOS	
Estado Actual	Estudiantes
CON TRIBUNAL DEFENSA ORAL	MEJIA NAVARRETE RAUL DAVID
Estados Posibles	Datos Requeridos: Grado Oral
<input checked="" type="radio"/> CONCLUIDO	Número de Folio IERI-10
	Nota de la Defensa Oral 09,90
	Nota Carrera Estudiantil 08,35
CLUSTER DE COMPUTADORES PERSONALES CON LINUX	
PRESENTADO POR: DIEGO ALEJANDRO FERNANDEZ AYALA RAUL DAVID MEJIA NAVARRETE	
<input type="button" value="Reset"/> <input type="button" value="Cambiar"/> <input type="button" value="Salir"/>	

Con Tribunal Defensa Oral

Del estado “Con Tribunal Defensa Oral” únicamente se podrá cambiar al estado “Concluido”, para lo cual es necesario que el usuario ingrese el número del archivo del estudiante, la nota promedio de la defensa oral y la nota promedio de la carrera estudiantil.

Tabla 1-1 Posibles pantallas que presenta el formulario “Cambio de Estados”.

Para el caso de los proyectos conjuntos, es importante señalar que hasta la transición: del estado “Con Tribunal Examinador” al estado “Apto para defensa Oral”, los dos estudiantes cambian su estado juntos (automáticamente) ya que hasta ese momento la información es común; por ejemplo, igual fecha de aprobación, iguales miembros de Tribunal Examinador, etc., de ahí en adelante hay que hacer una transición de estados individual, ya que sus datos comienzan a variar; por ejemplo, diferente hora de defensa, diferentes notas, etc.

Durante los estados “En Aprobación”, y “Delegado a Coordinador” se podrán agregar las observaciones realizadas durante una sesión de Subcomisión Académica, utilizando el botón Observaciones que se puede apreciar en las figuras de la Tabla correspondientes a cada estado. Sin embargo, ésta es la función principal para la que fue desarrollado el Formulario de InfoPath, en donde se podrán agregar las observaciones a varios Proyectos de una sola vez.

La interfaz de usuario del Formulario de InfoPath cuenta con un encabezado en el que se debe seleccionar la fecha de la Sesión de Subcomisión Académica, y escribir el nombre de la secretaria responsable de tomar las notas, como se puede observar en la Ilustración 1-9.

Sesión de Subcomisión Académica
Electrónica y Redes de Información

FECHA DE LA SESIÓN: 08/03/2006

SECRETARIA:

DE TITULACIÓN

Título:

lun	mar	mié	jue	vie	sáb	dom
27	28	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Hoy: 08/03/2006

Ilustración 1-9 Formulario de InfoPath (1/3).

En el Formulario de Infopath, el usuario no seleccionara el nombre de un estudiante sino el título del proyecto, una vez seleccionado un proyecto se presentarán los datos generales asociados (ej. nombre del director, estudiante o estudiantes, etc.), y adicionalmente se podrán ingresar las observaciones para el proyecto con controles muy amigables (a manera de texto con viñetas) permitiendo además indicar si son observaciones de delegación o no.

El elemento principal de la interfaz de usuario del Formulario de Infopath, es una sección de repetición, que contendrá la información de cada uno de los Planes de Proyectos de Titulación tratados en la fecha de la sesión especificada. En la Ilustración 1-10 se presentan dos secciones de repetición.

PLAN DE PROYECTO DE TITULACIÓN
 IMPLEMENTACIÓN DE UN SISTEMA DISTRIBUIDO DE AUTENTICACIÓN BIOMÉTRICA

Título:
 IMPLEMENTACIÓN DE UN SISTEMA DISTRIBUIDO DE AUTENTICACIÓN BIOMÉTRICA BASADO EN LA DETECCIÓN E IDENTIFICACIÓN DE ROSTROS HUMANOS EN IMÁGENES EXTRAÍDAS DE SECUENCIAS DE VIDEO

Director:
 JIMENEZ JIMENEZ MARIA SOLEDAD

Propuesto Por:
 GUALLE CALAHORRANO DIEGO XAVIER
 ROSALES YEPEZ JOSE FRANCISCO

Delegación a Coordinador:
 Sí
 No

OBSERVACIONES:

- Observación 1 para el Primer Plan de proyecto tratado
- Observación 2 para el Primer Plan de proyecto tratado
- Observación 3 para el Primer Plan de proyecto tratado

PLAN DE PROYECTO DE TITULACIÓN
 DISEÑO E IMPLEMENTACIÓN DE UN CLIENTE RADIUS EN LINUX

Título:
 DISEÑO E IMPLEMENTACIÓN DE UN CLIENTE RADIUS EN LINUX

Director:
 SINCHE MAITA SORAYA LUCIA

Propuesto Por:
 CEPEDA CARVAJAL CESAR IVAN
 PROAÑO SANCHEZ PABLO ARTURO

Delegación a Coordinador:
 Sí
 No

OBSERVACIONES:

- Observación 1 para el Segundo Plan de proyecto tratado
- Observación 2 para el Segundo Plan de proyecto tratado
- Observación 3 para el Segundo Plan de proyecto tratado

Control para agregar otro proyecto

Las Secciones se pueden repetir n veces

Combo Box Selección Tema

Sección de Repetición

Controles para Agregar Observaciones

Controles para especificar si las Observaciones son de Delegación o No

Control para agregar otro proyecto

Datos Generales del Proyecto

Se pueden agregar todas las Observaciones que sean necesarias, no hay límite

Agregar Plan de Proyecto de Titulación

Ilustración 1-10 Formulario de InfoPath (2/3).

En la Ilustración 1-10 se muestra un ejemplo de la Interfaz del Formulario de InfoPath, en este caso el Formulario está compuesto por dos secciones de repetición (PPTs), y como se puede ver existen controles (botones) que permiten agregar⁸¹ más secciones de repetición; el control que se encuentra en la parte inferior de la figura, agrega la sección de repetición justo bajo la anterior, mientras que el control que se encuentra en la parte superior izquierda de la figura, despliega un menú contextual [Ilustración 1-11].

⁸¹ Los controles para agregar, en realidad aumentan una sección de repetición al Formulario de Infopath, y se los puede identificar fácilmente porque se presentan con un ícono de punta de flecha hacia abajo.

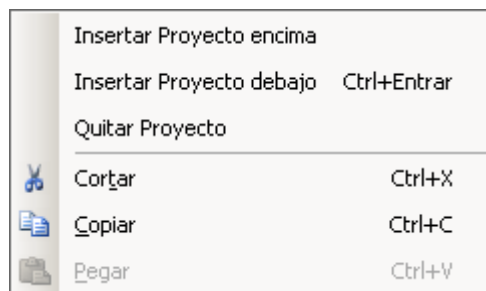


Ilustración 1-11 Menú contextual para el Formulario de InfoPath.

Como se puede apreciar en la Ilustración 1-11, el menú contextual del Formulario de Infopath permite seleccionar en donde insertar la nueva sección de repetición (encima o debajo), así como también, eliminar, cortar o copiar la información contenida en una determinada sección.

En la parte final del Formulario de InfoPath se encuentra el botón **Guardar en la Base de Datos** [Ilustración 1-12], que como su nombre lo indica permite agregar todos los comentarios de cada uno de los proyectos ingresados en el Formulario a la SPODB con un solo *clic*.

Esta operación podrá ser ejecutada una sola vez ya que de lo contrario generaría registros duplicados en la base de datos, luego de invocado el comando, el botón desaparece y el Formulario de InfoPath podrá cerrarse, ser guardado o impreso desde el menú **Archivo** de InfoPath, como una constancia de la sesión de Subcomisión Académica de esa fecha.

ESTA OPERACIÓN SE EJECUTARÁ UNA SOLA VEZ

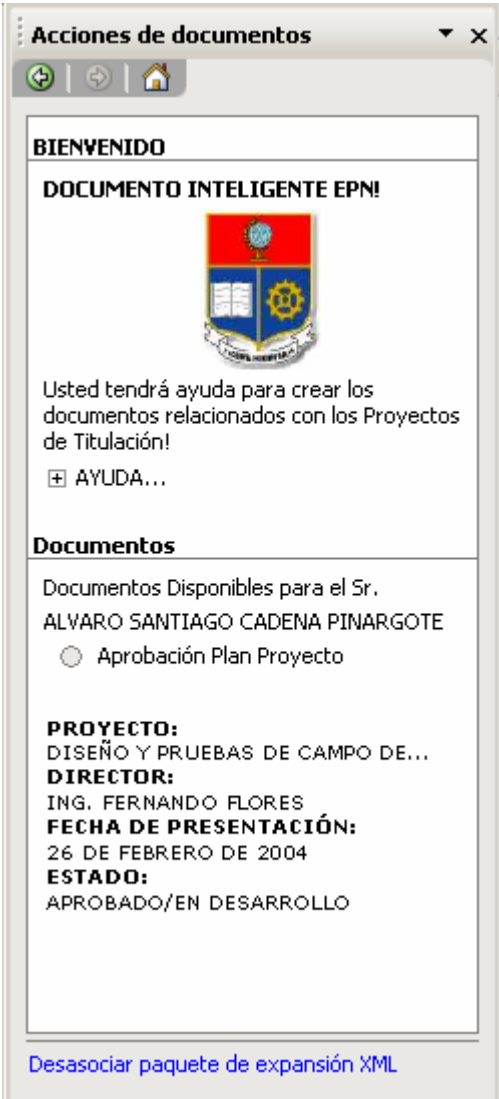
Guardar en la Base de Datos

Ilustración 1-12 Formulario de InfoPath (3/3).

2. ELABORACIÓN DE CARTAS Y MEMORANDOS

Ésta es la función principal que cumplirá el Documento Inteligente y se podría decir que la SPO en general. A continuación se presentarán ejemplos que permitan visualizar las características más importantes que reflejan su correcto funcionamiento.


1.5. Generación de Documentos



Acciones de documentos

BIENVENIDO

DOCUMENTO INTELIGENTE EPN!



Usted tendrá ayuda para crear los documentos relacionados con los Proyectos de Titulación!

[AYUDA...](#)

Documentos

Documentos Disponibles para el Sr.
ALVARO SANTIAGO CADENA PINARGOTE

- Aprobación Plan Proyecto

PROYECTO:
DISEÑO Y PRUEBAS DE CAMPO DE...

DIRECTOR:
ING. FERNANDO FLORES

FECHA DE PRESENTACIÓN:
26 DE FEBRERO DE 2004

ESTADO:
APROBADO/EN DESARROLLO

[Desasociar paquete de expansión XML](#)

Documentos Disponibles

Documentos

Documentos Disponibles para el Sr.
ALVARO SANTIAGO CADENA PINARGOTE

- Lista de Cambios Solicitados

En Aprobación

Documentos

Documentos Disponibles para el Sr.
ALVARO SANTIAGO CADENA PINARGOTE

- Lista Cambios Finales

Delegado a Coordinador

Documentos

Documentos Disponibles para el Sr.
ALVARO SANTIAGO CADENA PINARGOTE

- Aprobación Plan Proyecto

Aprobado/ En Desarrollo

Documentos

Documentos Disponibles para el Sr.
ALVARO SANTIAGO CADENA PINARGOTE

- Designación Tribunal Calificador

Con Tribunal Examinador

Documentos

Documentos Disponibles para el Sr.
ALVARO SANTIAGO CADENA PINARGOTE

- Declaración Aptitud Defensa Oral
- Proveido Calificaciones Escrito

Apto para Defensa Oral

Documentos

Documentos Disponibles para el Sr.
ALVARO SANTIAGO CADENA PINARGOTE

- Designación Tribunal Grado Oral
- Formulario para la Defensa Oral

Con Tribunal Defensa Oral

Documentos

Documentos Disponibles para el Sr.
ALVARO SANTIAGO CADENA PINARGOTE

- Proveido Defensa Oral
- Proveido Notas y Promedio

Concluido

Ilustración 2-1 Panel de Tareas de Acción para la generación de documentos.

La generación de documentos es un proceso fácil para el usuario, básicamente hay que seleccionar el estudiante en la barra de herramientas personalizada del Documento Inteligente [Ilustración 2-1], en ese instante aparecerán en el panel de tareas de acción los datos generales del estudiante, del proyecto y los documentos que se pueden generar como se puede apreciar en el ejemplo del lado izquierdo de la Ilustración 2-1.

Los documentos a generarse también dependen del estado del proyecto presentado por el estudiante, y deben concordar con el diagrama de estados presentado en la Ilustración 3-1; en la parte derecha de la Ilustración 2-1 se presenta la parte del panel de tareas correspondiente a las opciones de los documentos que se pueden generar según el estado del proyecto.

Al seleccionar la opción en el panel de tareas, del documento que se desea visualizar, se genera el documento y se presenta en la ventana de trabajo principal de Word tal como se muestra en la Ilustración 2-2.

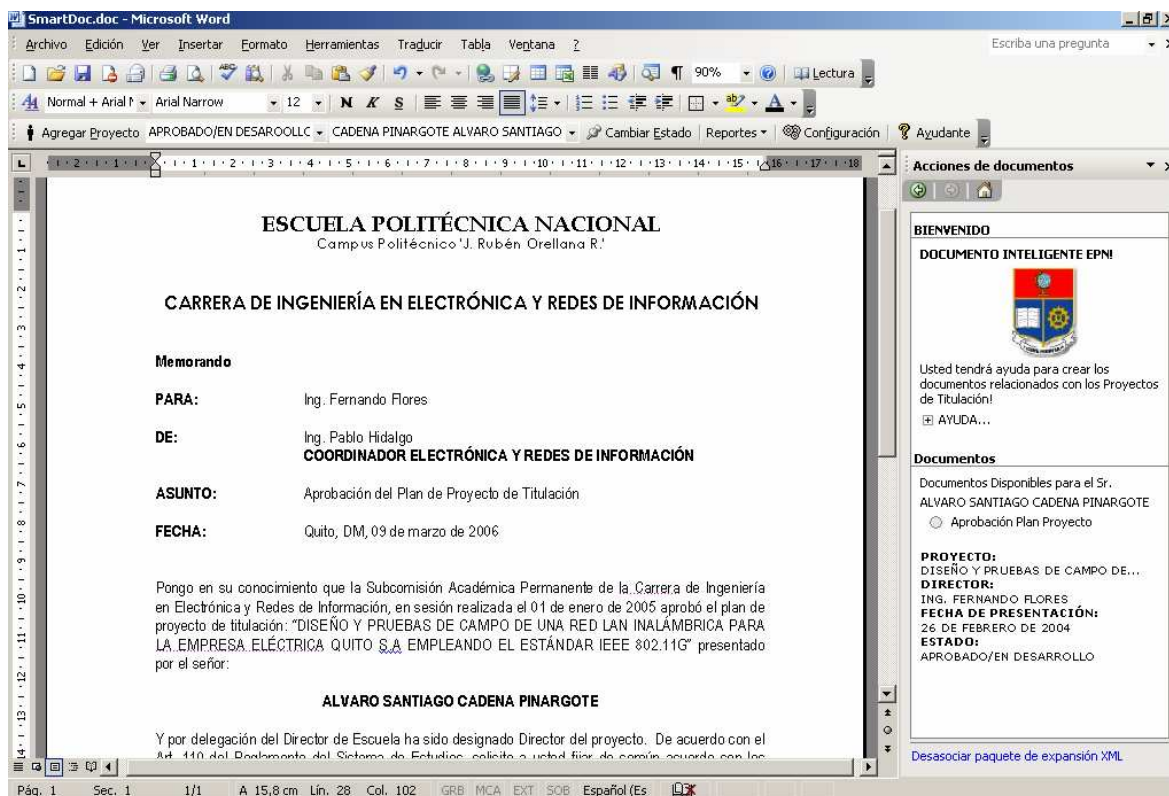


Ilustración 2-2 Memorando generado por el Documento Inteligente.

Algunas de las características más importantes que cumplen los documentos generados son las siguientes:

- Los artículos y calificativos personales son concordantes con el sexo de los estudiantes y de los profesores [Ilustración 2-3].

PARA:	Ing. Ivan Bernal
DE:	Ing. Pablo Hidalgo COORDINADOR ELECTRÓNICA Y REDES DE INFORMACIÓN
ASUNTO:	Aprobación del Plan de Proyecto de Titulación
FECHA:	Quito, DM, 09 de marzo de 2006

Pongo en su conocimiento que la Subcomisión Académica Permanente de la Carrera de Ingeniería en Electrónica y Redes de Información, en sesión realizada el 15 de febrero de 2006 aprobó el plan de proyecto de titulación: "PROYECTO DE PRUEBA DE DOS ESTUDIANTES DE SEXOS DIFERENTES" presentado por la señorita y el señor:

CRISTINA ELIZABETH TORRES JARAMILLO
SANTIAGO DAMIAN GUDINO CARRILLO

Y por delegación del Director de Escuela ha sido designado Director del proyecto. De acuerdo con el Art. 110 del Reglamento del Sistema de Estudios, solicito a usted fijar de común acuerdo con los

Ilustración 2-3 Comprobación de artículos y calificativos personales.

- El remitente del memorando siempre es el Director de Escuela cuando el director del proyecto es el Coordinador de Carrera [Ilustración 2-4].

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE INFORMACIÓN	
Memorando	
PARA:	Ing. Pablo Hidalgo
DE:	Ing. Jorge Molina DIRECTOR ESCUELA DE INGENIERÍA
ASUNTO:	Aprobación del Plan de Proyecto de Titulación
FECHA:	Quito, DM, 09 de marzo de 2006

Pongo en su conocimiento que la Subcomisión Académica Permanente de la Carrera de Ingeniería en Electrónica y Redes de Información, en sesión realizada el 1 de enero de 2006 aprobó el plan de

Ilustración 2-4 Comprobación de remitente.

- Absolutamente todo el texto de los documentos es generado automáticamente inclusive los números escritos en letras, para los documentos que requieran esta información [Ilustración 2-5].

UNIDAD ACADÉMICA: INGENIERÍA EN ELECTRÓNICA Y REDES DE INFORMACIÓN. De conformidad con lo dispuesto en el Art. 119 del Reglamento de Sistemas de Estudios, y por disposición del Director de la Escuela de Ingeniería, se establece la nota final de titulación, según la suma de lo siguiente:

ITEM DE CALIFICACIÓN	PUNTAJE EN LETRAS	PUNTAJE
Valor del promedio de su carrera estudiantil	siete puntos veintidós centésimas sobre diez (7,22/10)	7,22/10
Calificación promedio del proyecto de titulación escrito	ocho puntos ochenta y tres centésimas sobre diez (8,83/10)	8,83/10
Calificación promedio de la prueba oral	diez puntos sobre diez (10,00/10)	10,00/10
TOTAL	veintiséis puntos cinco centésimas sobre treinta (26,05/30)	26,05/30

En consecuencia se declara que el señor EDWIN GIOVANNY CONDOR GORDON ha obtenido la nota de VEINTISÉIS PUNTOS CINCO CENTÉSIMAS SOBRE TREINTA (26,05/30) redondeada a VEINTISÉIS PUNTOS SOBRE TREINTA (26/30), por lo que se le confiere el título de Ingeniero en Electrónica y Redes de Información, con la calificación de

Ilustración 2-5 Escritura automática de notas en palabras.

- Todos los formatos de las plantillas utilizadas por las secretarías fueron respetados, los documentos que requieren formatos especiales como sangrías francesas o párrafos numerados se generan con todas sus características; por ejemplo, en la Ilustración 2-6 se presenta el formato del documento “Formulario para la Defensa Oral de Ingeniero”.

FORMULARIO PARA LA DEFENSA ORAL DE INGENIERO

TEMA: → CLUSTER DE COMPUTADORES PERSONALES CON LINUX

DIRECTOR: → Ing. Iván Bernal

FECHA GRADO ORAL: → martes, 07 de febrero del 2006 a las 00:00

GRADUADO: → RAUL DAVID MEJÍA NAVARRETE

LUGAR DE TRABAJO: → No especificado

DIRECCIÓN PERMANENTE: → Coop. Unión Naval 2 casa 45

CORREO ELECTRÓNICO: → daviddam@hotmail.com

TELEFONO: → 095028295

Señores Miembros del Tribunal favor indicar su recomendación para la difusión del Proyecto de Titulación.

Párrafo numerado

1. → JORNADAS DE INGENIERIA ELECTRICA Y ELECTRÓNICA →

2. → CURSOS DE EDUCACIÓN CONTINUA →

3. → SEMINARIO A ESTUDIANTES →

4. → CONFERENCIA A ESTUDIANTES →

5. → OTROS (ESPECIFIQUE) →

Sangría Francesa

Ilustración 2-6 Formato especial de los documentos generados.

1.6. Visualización de Información Sensitiva al Contexto

El elemento fundamental que hará posible tener esta funcionalidad en cada uno de los documentos que se auto generarán, es el esquema XML de la solución.

Los documentos generados son etiquetados adecuadamente para que el usuario pueda gozar de la característica principal de la lógica inteligente del documento, que es la ayuda relativa al contexto. Un ejemplo de uno de los documentos generados con sus respectivas etiquetas XML se presenta en la Ilustración 2-7.

The screenshot displays a document editor window with the following content and XML tags:

- Document: `<Documento>` (containing `<Campus>` ESCUELA POLITÉCNICA NACIONAL)
- Carrera: `<Carrera>` CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE INFORMACIÓN
- Memorando: `<Etiqueta1>` (containing `<Memorando>`)
- Para: `<Para>` (containing `<Director>` Ing. Xavier Armendariz)
- Profesor: `<Profesor>` Ing. Oswaldo Aldaz
- MiembroTribunal1: `<MiembroTribunal1>` Ing. Renato Andrade
- De: `<De>` (containing `<Profesor>` Ing. Pablo Hidalgo)

The right-hand panel, 'Estructura XML', shows a tree view of the document structure. The selected 'Profesor' element is expanded to show its attributes:

```

URI=EPN.wgvc/DI/Documentos/Electronica
codigo="200320010"
dedicacion="N/E"
fecnac="21 de septiembre de 2000"
cedula="1711337830"
Asuntititacac="INGENIERO EN SISTEMAS"
direccion="IBERIA 737 Y CALLE B I"
email="xarmendariz@hotmail.com"
telefono1="2524271"
telefono2="telefono2"
nombre="XAVIER SALVATORE ARMENDARIZ CORRALES"
path_foto="G:\Diseños SCORE LISTOS\colgante1.jpg"
rol_proyecto="2"
  
```

Ilustración 2-7 Documento generado y sus etiquetas.

Como se puede ver en la Ilustración 2-7 las etiquetas XML colocadas en los documentos generados tiene sus atributos en concordancia con el valor de su contenido, el usuario en general no visualizará los documentos tal como se presentan en la figura ya que para ello tiene que hacer visibles las etiquetas, pero si tendrá varias formas de ayuda en el panel de tareas que se van presentando según su posición en el documento, algunos ejemplos se presentan a continuación.

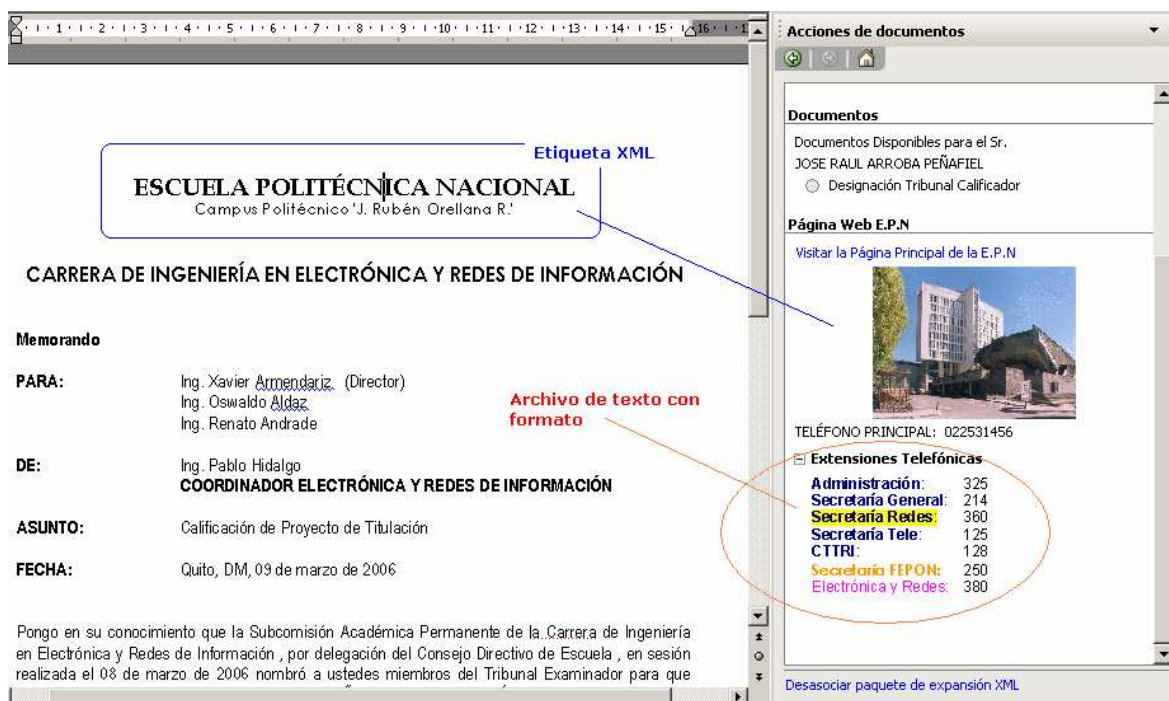


Ilustración 2-8 Ayuda relativa al contexto.

En la Ilustración 2-8 se presenta el panel de tareas que aparece cuando el usuario se coloca sobre el encabezado “ESCUELA POLITÉCNICA NACIONAL”, en este caso el panel muestra un hipervínculo a la página Web de la universidad, y ayuda sobre las extensiones telefónicas de uso común para las secretarías.

Para mostrar las extensiones telefónicas que se ven en este ejemplo, se está utilizando uno de los archivos XML con formato, mencionados en la Sección 3.2.2.2, y se puede ver claramente que se conservan sin problema los formatos especificados (fuentes, colores, tamaños, etc.).

1.7. Envío Directo de Documentos Vía e-mail

Esta función estará disponible sin abandonar el Documento Inteligente, se manejará el modelo de objetos de Outlook 2003, para que el usuario no necesite abrir una instancia independiente del mismo; sin embargo, cuando lo haga podrá visualizar los documentos enviados desde el Documento Inteligente en la carpeta de correos enviados, o en la lista de envío.

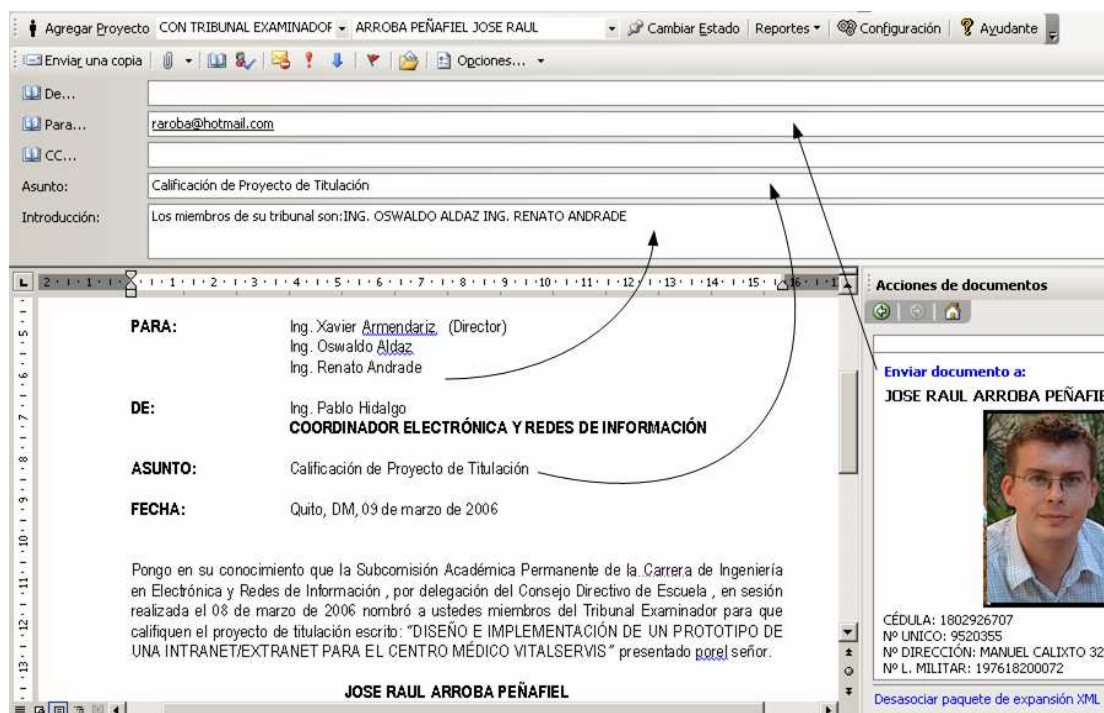


Ilustración 2-9 Envío de documentos por correo electrónico..

En la Ilustración 2-9 se puede apreciar otro ejemplo del panel de tareas de acción, en este caso, la ayuda presentada son los datos personales de un estudiante, todas las personas (profesores o estudiantes) cuyo nombre aparezca en el documento, presentarán un panel de tareas similar al de este ejemplo.



Outlook es utilizado para configurar todos los parámetros del correo electrónico a ser enviado, en este caso se llena automáticamente el asunto, la introducción y el destinatario del correo.



1.8. Edición del Contexto de los Documentos

El usuario podrá configurar la seguridad de edición de secciones en cada uno de los documentos auto generados, mediante el **Formulario de Configuración** en la pestaña **Seguridad de Escritura** [Ilustración 2-10]. La base de datos local contendrá el nombre de usuario y la contraseña que permitirán establecer esta política de seguridad, los mismos que podrán ser configurados desde la pestaña **Cambio de Contraseña** [Ilustración 2-11].

Formulario de Configuración

CONFIGURACIONES

 Direcciones URL
 Sincronizar Datos SAE

 Seguridad de Escritura
 Cambio de Contraseña
 Archivos de Texto

DOCUMENTOS SIN PERMISO DE ESCRITURA

- Aprobación del Plan de Proyecto de Titulación
- Nombramiento del Tribunal Examinador Proyecto Escrito
- Proveído de las Calificaciones del Proyecto Escrito
- Declaración de Aptitud para Rendir la Defensa Oral
- Designación para Integrar el Tribunal de la Defensa Oral
- Formulario para la Defensa Oral
- Proveido de las Calificaciones Defensa Oral
- Proveido sobre Notas y Promedio de Graduación

Ilustración 2-10 “Formulario de Configuración” opción: “Seguridad de Escritura”.

Formulario de Configuración

CONFIGURACIONES

 Direcciones URL
 Sincronizar Datos SAE

 Seguridad de Escritura
 Cambio de Contraseña
 Archivos de Texto

Cambio de contraseña para wgv

Contraseña Actual

Nueva Contraseña

Confirmar Nueva Contraseña

Ilustración 2-11 “Formulario de Configuración” opción: “Cambio de Contraseña”.

Proyecto CON TRIBUNAL EXAMINADOR ARROBA PEÑAFIEL JOSE RAUL Cambiar Estado Reportes Configuración Ayudante

ESCUELA POLITÉCNICA NACIONAL
Campus Politécnico 'J. Rubén Orellana R.'

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE INFORMACIÓN

Memorando

PARA: Ing. Xavier Armendariz (Director)
Ing. Oswaldo Alda
Ing. Renato Andrade

DE: Ing. Pablo Hidalgo
COORDINADOR ELECTRÓNICA Y REDES DE INFORMACIÓN

ASUNTO: Calificación de Proyecto de Titulación

FECHA: Quito, DM, 09 de marzo de 2006

Pongo en su conocimiento que la Subcomisión Académica Permanente de la Carrera de Ingeniería en Electrónica y Redes de Información, por delegación del Consejo Directivo de Escuela, en sesión realizada el 08 de marzo de 2006 nombró a ustedes miembros del Tribunal Examinador para que califiquen el proyecto de titulación escrito: "DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE UNA INTRANET/EXTRANET PARA EL CENTRO MÉDICO VITALSERVIS" presentado por el señor.

JOSE RAUL ARROBA PEÑAFIEL

Acciones de documentos

DOCUMENTO INTELIGENTE EPNI

Usted tendrá ayuda para crear los documentos relacionados con los Proyectos de Titulación

AYUDA...

Documentos

Documentos Disponibles para el Sr. JOSE RAUL ARROBA PEÑAFIEL

Designación Tribunal Calificador

mar 2006

lun	mar	mié	jue	vie	sáb	dom
27	28	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Desasociar paquete de expansión XML

Ilustración 2-12 Protección de párrafos y secciones.

Cuando un documento esté protegido contra la edición de párrafos y secciones, existirán ciertas partes que si tendrán permisos de manera predeterminada, las mismas que serán resaltadas tal como se puede ver en la Ilustración 2-12, el resto del documento no podrá ser modificado ni en contenido ni en formato.

La Ilustración 2-12 muestra un ejemplo distinto de la ayuda prestada por el panel de tareas, que en este caso permite insertar la fecha de emisión del memorando con la ayuda visual de un calendario.

1.9. Edición del Texto Común (repetitivo)

El usuario tendrá acceso a los archivos de texto XML con formato o sin formato desde el **Formulario de Configuración** opción **Archivos de Texto** [Ilustración 2-13], en donde el usuario podrá elegir el archivo que desea editar. Todos los archivos se abren en Word en una ventana diferente a la del Documento Inteligente.

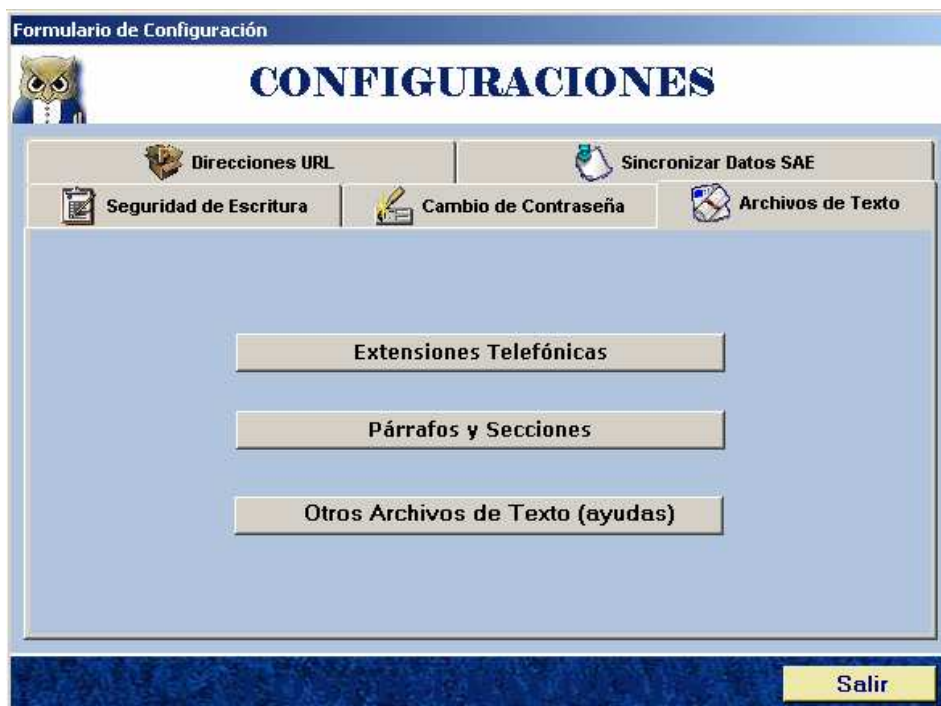
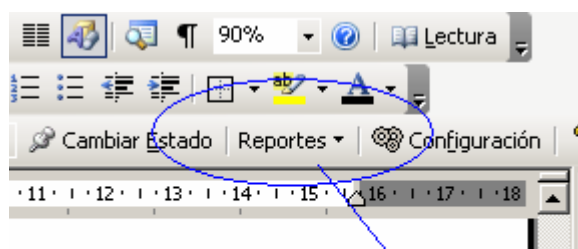


Ilustración 2-13 Formulario de Configuración opción Archivos de Texto.

3. VISUALIZACIÓN DE REPORTES

En el Documento Inteligente existirá la opción de generación de reportes, y únicamente cuando el usuario elija esta opción, entrarán en funcionamiento los Servicios Web XML que como se dijo anteriormente tendrán una conexión a la base de datos local con permiso de lectura. Para acceder a la funcionalidad de los reportes se utiliza un botón en la barra de herramientas personalizada como se muestra en la Ilustración 3-1.



**Botón que permite
seleccionar el tipo de
reporte**

Ilustración 3-1 Barra de herramientas personalizada del Documento Inteligente.

1.10. Utilización de Filtros de Información

Los Servicios Web XML proveerán todos los métodos necesarios para obtener la información que esté en relación a varios filtros de búsqueda que podrá aplicar el usuario antes de generar el reporte.

Estos filtros permitirán seleccionar las carreras, el estado de los proyectos, el director, o algún texto incluido en el título del proyecto, y estarán disponibles a través del panel de tareas de acción como se puede apreciar en la Ilustración 3-2.

Si el usuario decide que la información de alguna de las carrera no es útil para un reporte determinado, entonces la aplicación simplemente no consumirá el Servicio Web que proporcione dicha información.

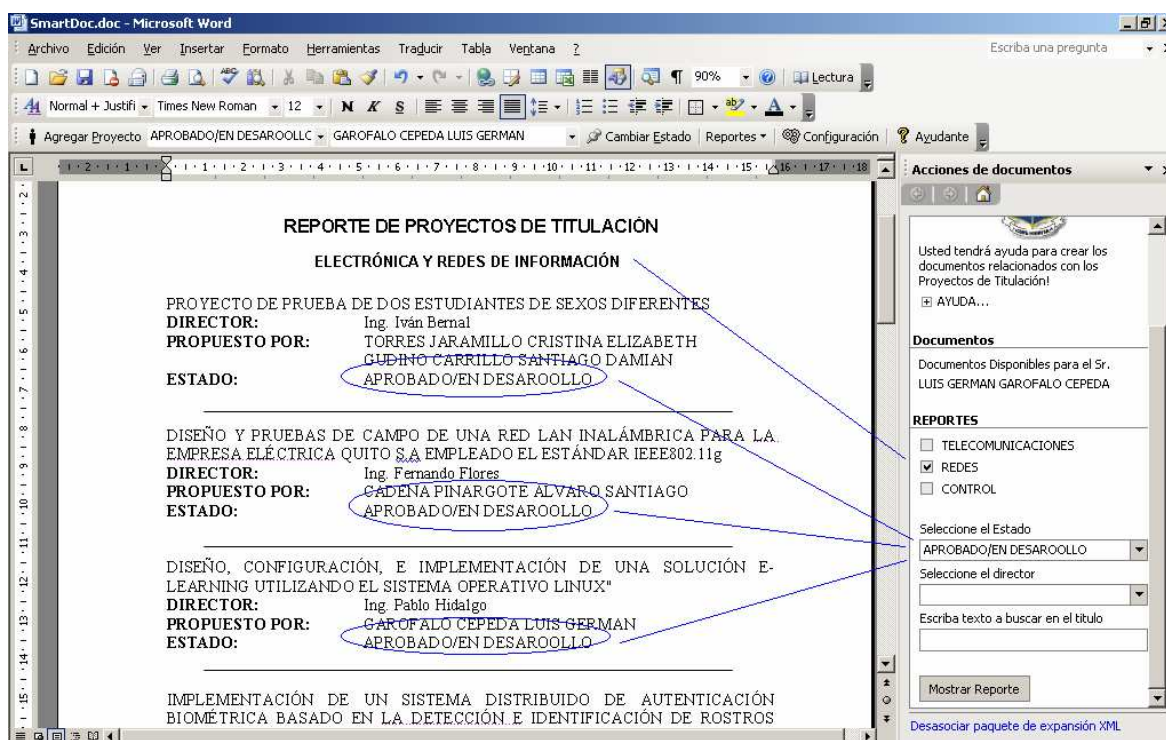


Ilustración 3-2 Generación de Reportes.

SmartDoc.doc - Microsoft Word

Archivo Edición Ver Insertar Formato Herramientas Traducir Tabla Vengana 2 Escriba una pregunta

Normal + Justifi Times New Roman 12

Agregar Proyecto APROBADO/EN DESAROOOLL C GAROFALO CEPEDA LUIS GERMAN Cambiar Estado Reportes Configuración Ayudante

REPORTE DE PROYECTOS DE TITULACIÓN

ELECTRÓNICA Y REDES DE INFORMACIÓN

PROYECTO DE PRUEBA DE DOS ESTUDIANTES DE SEXOS DIFERENTES
DIRECTOR: Ing. Iván Bernal
PROPUESTO POR: TORRES JARAMILLO CRISTINA ELIZABETH
 GUDINO CARRILLO SANTIAGO DAMIAN
ESTADO: APROBADO/EN DESAROOOLL

DISEÑO Y PRUEBAS DE CAMPO DE UNA RED LAN INALÁMBRICA PARA LA
 EMPRESA ELÉCTRICA QUITO S.A EMPLEADO EL ESTÁNDAR IEEE802.11g
DIRECTOR: Ing. Fernando Flores
PROPUESTO POR: CADENA PINARGOTE ALVARO SANTIAGO
ESTADO: APROBADO/EN DESAROOOLL

DISEÑO, CONFIGURACIÓN, E IMPLEMENTACIÓN DE UNA SOLUCIÓN E-
 LEARNING UTILIZANDO EL SISTEMA OPERATIVO LINUX"
DIRECTOR: Ing. Pablo Hidalgo
PROPUESTO POR: GAROFALO CEPEDA LUIS GERMAN
ESTADO: APROBADO/EN DESAROOOLL

IMPLEMENTACIÓN DE UN SISTEMA DISTRIBUIDO DE AUTENTICACIÓN
 BIOMÉTRICA BASADO EN LA DETECCIÓN E IDENTIFICACIÓN DE ROSTROS

Acciones de documentos

Usted tendrá ayuda para crear los documentos relacionados con los Proyectos de Titulación!
 AYUDA...

Documentos
 Documentos Disponibles para el Sr. LUIS GERMAN GAROFALO CEPEDA

REPORTES

TELECOMUNICACIONES
 REDES
 CONTROL

Seleccione el Estado
 APROBADO/EN DESAROOOLL

Seleccione el director

Escriba texto a buscar en el título

Mostrar Reporte

Desasociar paquete de expansión XML

Ilustración 3-3 Generación de Reportes 2.

1.11. Configuración de Direcciones URL

Esta función también estará disponible a través del **Formulario de Configuración**, opción **Direcciones URL** [Ilustración 3-4], y en vista de que es otra de las funciones críticas, ya que de la correcta configuración de estas direcciones depende que se pueda establecer o no la interconexión lógica con cada una de las Carreras; se requerirá nombre de usuario y contraseña.

Formulario de Configuración

CONFIGURACIONES

Seguridad de Escritura | Cambio de Contraseña | Archivos de Texto

Direcciones URL | Sincronizar Datos SAE

Dirección del Servidor de Electrónica y Control
 Probar Conexión

Dirección del Servidor de Electrónica y Redes
 Probar Conexión

Dirección del Servidor de Electrónica y Telecomunicaciones
 Probar Conexión

Salir

Ilustración 3-4 “Formulario de Configuración” opción: “Direcciones URL”.

ANEXO H

ANÁLISIS COSTO BENEFICIO

Se ha decidido incluir el análisis Costo Beneficio de la Solución Personalizada de Office como un Anexo y no como una parte sustancial de este Proyecto de Titulación, en vista de que como ya se mencionó en la Sección 3.3.4 al no ser un proyecto comercial no se escatimaron recursos para el diseño e implementación y tampoco se pueden considerar exactamente factores importantes como el tiempo de vida útil del sistema, el porcentaje de utilidad deseado, y principalmente los costos de implementación comerciales, haciendo que éste análisis esté de cierta forma un tanto alejado de la realidad, no tanto en valores sino más bien en las condiciones de su aplicación, como se podrá ver a continuación.

Personal y Tiempo necesario para el Desarrollo

Este es un proyecto de desarrollo unipersonal, y el tiempo necesario para la construcción y entrega fue considerado desde la etapa de investigación o aprendizaje de la tecnología como se puede apreciar en la siguiente tabla.

Actividad	D	HD	TH	CH	CT	TOTAL
Investigación	60	8	480	\$ 4,00	\$ 1.920,00	\$ 3.560,00
Diseño y Planificación	20	8	160	\$ 4,00	\$ 640,00	
Desarrollo	25	10	250	\$ 4,00	\$ 1.000,00	

Tabla 1 Costo de desarrollo

El significado de la nomenclatura utilizada se especifica a continuación:

SIGLAS	SIGNIFICADO
D	Días
HD	Horas por Día
TH	Total de Horas
CH	Costo por Hora
CT	Costo Total

Tabla 2 Nomenclatura de costos.

Costos de Desarrollo

Costo de Materiales

MATERIALES	COSTO
Papel (6000 hojas)	\$ 36.00
Memoria Flash	\$ 42.00
Tinta (4 cartuchos)	\$ 140.00
Encuadernación	\$ 60.00
Útiles de Oficina	\$ 10.00
Copias (300 hojas)	\$ 10.00
TOTAL:	\$ 298.00

Tabla 3 Costo de materiales.

Costo Generales de Producción y Servicios

SERVICIO/OTROS	COSTO
Servicios Básicos (\$20.00 mensuales x 6)	\$ 120.00
Seguro de equipos (\$30.00 anuales)	\$ 15.00
Servicio de Internet (\$30.00 mensualesx6)	\$ 180.00
Depreciación de bienes (monto \$2000.00)	\$ 330.00
Movilización (\$40.00 mensuales x 3)	\$ 120.00
Pérdida por daños de equipos (1 disco)	\$ 80.00
TOTAL:	\$ 845.00

Tabla 3 Costo de producción y servicios.

El costo total del desarrollo es de **\$4703.00**, en vista de que el sistema fue diseñado para la utilización en tres de las Carreras de la Escuela Politécnica Nacional, el costo de la solución por cada una de ellas sería de **\$1567.00**.

Costo de Implementación

Se considera un costo de implementación por Carrera.

SOFTWARE/OTROS	COSTO
Sistema Operativo Windows XP profesional	\$ 177.00
SQL Server 2000	\$ 1200.00
Microsoft Office 2003	\$ 360.00
Microsoft Office Infopath 2003	\$ 256.00
Movilización	\$ 40.00
Configuración e instalación del sistema	\$ 25.00
Capacitación (10 horas)	\$ 50.00
TOTAL:	\$ 2108.00

Tabla 3 Costo de implementación.

Relación Costo Beneficio considerando la Producción de Documentos

Aproximadamente las tres Carreras de Electrónica generan documentos para 210 estudiantes al año, para cada estudiante se generan 12 documentos, por lo tanto en un año las secretarías generarán 2520 documentos relacionados con los proyectos de titulación de los estudiantes.

En promedio la generación de un documento sin el sistema le toma a cada secretaria 12 minutos, mientras que utilizando el sistema en óptimas condiciones 2 minutos.

Con esta información se puede concluir que en la actualidad se utilizan 504 horas por año para elaborar documentos relacionados con los proyectos de titulación:

$$\frac{2520 \text{ documentos}}{\text{año}} \times \frac{12 \text{ minutos}}{1 \text{ documento}} \times \frac{1 \text{ hora}}{60 \text{ minutos}} \approx \frac{504 \text{ horas}}{\text{año}}$$

Mientras que utilizando el sistema se utilizarían sólo 84 horas por año:

$$\frac{2520 \text{ documentos}}{\text{año}} \times \frac{2 \text{ minutos}}{1 \text{ documento}} \times \frac{1 \text{ hora}}{60 \text{ minutos}} \approx \frac{84 \text{ horas}}{\text{año}}$$

El costo por hora de una secretaria es de \$3.00; en el año existe un ahorro de 420 horas (504 - 84), por lo que existiría un ahorro de $420 * \$3.00 = \1260.00 en las tres Carreras; es decir, \$420,00 en cada una.

Si el sistema tiene un tiempo de vida útil de tres años, considerando un incremento de salarios del 10% anual, se tendría un ahorro total de **\$1390.00** en cada Carrera.

El costo de la implementación del sistema por Carrera es de **\$3675.00**, por lo que a simple vista se puede concluir que la implementación del sistema no es rentable bajo las condiciones planteadas.

$$\frac{\text{costo } 3675.00}{\text{beneficio } 1390.00} \gg 1$$

La relación costo beneficio es mucho mayor que la unidad, por lo tanto el sistema no debe ser implementado por ningún motivo bajo las condiciones planteadas.

Análisis de la Relación Costo Beneficio

La razón fundamental por la cual el proyecto no resulta rentable, se encuentra en que los costos de implementación son más elevados que los costos del sistema en sí, con lo cual se puede ver muy fácilmente que si la institución carece de las licencias del software base necesario, la implementación del sistema por si solo no es rentable, (las licencias estarían subutilizadas).

Sin embargo considerando un caso más práctico y real, en el cual la Escuela Politécnica Nacional cuente con las licencias necesarias, el costo sería el siguiente por cada una de las Carreras:

TOTAL SOLUCIÓN:	\$1567.00
SOFTWARE/OTROS	COSTO
Movilización	\$ 40.00
Configuración e instalación del sistema	\$ 25.00
Capacitación (10 horas)	\$ 50.00
TOTAL IMPLEMENTACIÓN:	\$ 115.00
TOTAL POR CARRERA:	\$1682.00

Como podemos apreciar aún así el costo de la solución (\$1682.00) sigue siendo mayor que el costo del beneficio (\$1390.00), por esta razón sería necesario implementar la solución por lo menos en una Carrera mas de las existentes en la universidad; por ejemplo, si la solución también es implementada en la Carrera de Ingeniería Eléctrica, los costos variarían de la siguiente manera:

El costo total de la solución por Carrera sería de \$4703.00 / 4= **\$1175.75**; en Ingeniería Eléctrica la secretaria realiza en promedio los documentos para 60 estudiantes al año; es decir, 720 documentos, los cuales deberán ser agregados a los otros 2520 de las otras Carreras.

$$\frac{3240 \text{ documentos}}{\text{año}} \times \frac{12 \text{ minutos}}{1 \text{ documento}} \times \frac{1 \text{ hora}}{60 \text{ minutos}} \approx \frac{648 \text{ horas}}{\text{año}}$$

Utilizando el sistema el tiempo a ser utilizado será:

$$\frac{3240 \text{ documentos}}{\text{año}} \times \frac{2 \text{ minutos}}{1 \text{ documento}} \times \frac{1 \text{ hora}}{60 \text{ minutos}} \approx \frac{108 \text{ horas}}{\text{año}}$$

Por lo tanto el ahorro de tiempo será de 540 horas y económicamente de $540 \times \$3.00 = \mathbf{\$1620.00}$, por las cuatro carreras; es decir, \$405.00 por cada una.

De igual forma considerando un tiempo de vida del sistema de tres años y con un incremento de sueldo anual de 10%, el total de ahorro será de **\$1340.55** por cada carrera.

Como se puede apreciar el ahorro se mantiene sobre los \$1300.00), pero el costo de la solución menor ($\$1175.75 + \$115.00 = \$1290.75$), ya es factible realizar la implementación, pero no ofrece un beneficio atractivo.

Sin embargo hay que hacer notar que la productividad del sistema se incrementa con la cantidad de personas que la utilizan, si la solución personalizada de Office fuera implementada en todas las facultades de la Escuela, el sistema resultaría muy beneficioso, ya que costo de la solución por carrera bajaría significativamente y el volumen de documentos se incrementaría notablemente principalmente con las carreras de Mecánica y Sistemas.

En conclusión para que el sistema pueda ser implementado con beneficio, debe ser utilizado por lo menos en cuatro de las Carreras de la Escuela Politécnica Nacional, lo cual gracias al diseño flexible y completamente configurable de la SPO es completamente factible, el único requisito para implementar el sistema en cualquiera de las Carreras de la EPN radica en la utilización de una copia del SAE correspondiente.