



**TESIS DE MAGÍSTER**  
**EN INGENIERÍA DEL SOFTWARE**  
**HERRAMIENTA DE INTEGRACION DE**  
**ALGORITMOS DE INDUCCION Y MAPAS**  
**AUTO-ORGANIZADOS**

**AUTOR: ING. ESTEBAN RICARDO SAL**

**DIRECTORES: M. ING. PAOLA BRITOS**

**M. ING. CLAUDIO RANCAN**

**BUENOS AIRES, 2007**

Agradecimientos:

*A toda mi familia.*

---

## ÍNDICE

	Página
CAPÍTULO 1 – PLAN DE SISTEMAS DE INFORMACIÓN _____	17
1.1. Introducción _____	17
1.2. Aproximación inicial al problema _____	18
1.3. Situación actual del Dominio _____	18
1.4. Análisis Comparativo entre las herramientas de explotación de datos disponibles en el mercado _____	21
1.5. Identificación de requisitos iniciales _____	23
1.5.1. Análisis de las Necesidades de Información _____	23
1.6. Solución propuesta _____	24
1.6.1. Metodología para la construcción del sistema _____	25
1.6.2 Selección del Ciclo de Vida _____	26
CAPÍTULO 2 – ESTUDIO DE VIABILIDAD DEL SISTEMA _____	28
2.1. Introducción _____	28
2.2. Análisis de la Situación y Restricciones _____	28
2.3. Descripción General del Sistema _____	29
2.4. Objetivo del proyecto _____	30
2.5. Estudio de la situación actual _____	31
2.6. Definición de requisitos del sistema _____	31
2.6.1. Requisitos funcionales _____	31
2.6.1.1. Requisitos funcionales Generales _____	31
2.6.1.2. Requisitos funcionales relacionados con Usuarios _____	31

---

	Página
2.6.1.3. Requisitos funcionales relacionados con Proyectos _____	32
2.6.1.4. Requisitos funcionales relacionados con Versiones _____	32
2.6.1.5. Requisitos funcionales relacionados con la Explotación de Datos _____	32
2.6.2. Requisitos de rendimiento _____	32
2.6.2.1. Estáticos _____	32
2.6.2.2. Dinámicos _____	32
2.6.3. Atributos del sistema software _____	33
2.6.3.1. Fiabilidad _____	33
2.6.3.2. Disponibilidad _____	33
2.6.3.3. Seguridad _____	33
2.6.3.4. Mantenimiento _____	33
2.6.3.5. Portabilidad _____	33
2.6.4. Catalogación de requisitos _____	34
2.6.4.1. Requerimientos de hardware y software _____	34
2.6.4.1.1. Estación de trabajo _____	34
2.6.4.2. Requisitos no funcionales – Normas y Estándares _____	35
2.6.4.3. Requisitos no funcionales – Seguridad _____	36
2.6.4.4. Requisitos no funcionales – Organización _____	36
2.6.4.5. Requisitos no funcionales – Backup _____	37
2.7. Catálogo de Usuarios _____	37
2.8. Tamaño, Esfuerzo y Duración del Proyecto _____	38
2.8.1. Características del Proyecto “DM” _____	38
2.8.2. Técnica de Estimación elegida _____	38
2.8.3. Desarrollo de la Técnica _____	38

---

	Página
2.8.3.1. Criterios para Clasificación y Asignación de Valores Numéricos _____	39
2.8.3.2. Estimación del Total de Puntos de Función _____	41
2.8.3.2.1. Caso de Uso: Ingreso al Sistema _____	41
2.8.3.2.2. Caso de Uso: Salir del Sistema _____	42
2.8.3.2.3. Caso de Uso: Agregar Usuario _____	42
2.8.3.2.4. Caso de Uso: Modificar Usuario _____	43
2.8.3.2.5. Caso de Uso: Crear Proyecto _____	44
2.8.3.2.6. Caso de Uso: Eliminar Proyecto _____	45
2.8.3.2.7. Caso de Uso: Listar Proyectos _____	45
2.8.3.2.8. Caso de Uso: Crear Versiones de un Proyecto _____	46
2.8.3.2.9. Caso de Uso: Abrir Versión _____	47
2.8.3.2.10. Caso de Uso: Eliminar Versión _____	48
2.8.3.2.11. Caso de Uso: Listar Versiones _____	49
2.8.3.2.12. Caso de Uso: Establecer Parámetros de SOM _____	49
2.8.3.2.13. Caso de Uso: Establecer Parámetros de C5.0 _____	50
2.8.3.2.14. Caso de Uso: Ejecutar Proceso de Explotación de Datos _____	51
2.8.3.2.15. Caso de Uso: Total General de Puntos de Función _____	52
2.8.3.3. Esfuerzo y Duración _____	53
2.9. Planificación _____	60
2.10. Gestión de Configuración _____	61
2.10.1. Identificación de la Configuración _____	61
2.10.2. Control de la Configuración _____	63
2.11. Gestión de Aseguramiento de Calidad _____	63
2.11.1. Introducción _____	63

---

	Página
2.11.2. Identificación de las Propiedades de Calidad _____	63
2.11.3. Revisiones _____	64
2.11.3.1. Revisiones definidas para la etapa de Análisis _____	64
2.11.3.2. Revisiones definidas para la etapa de Diseño _____	65
2.11.3.3. Revisiones definidas para la etapa de las Pruebas _____	65
2.11.4. Pruebas de Software _____	66
2.12. Gestión de Riesgos _____	66
2.12.1. Metodología para la Gestión de los Riesgos _____	66
2.12.2. Desarrollo de la Metodología para la Gestión de los Riesgos _____	67
CAPÍTULO 3 – MODELO DE ANÁLISIS _____	72
3.1. Definición del Sistema _____	72
3.1.1. Determinación del Alcance del Sistema _____	72
3.1.1.1. Modelo de Negocio _____	72
3.1.2. Identificación de los Usuarios Participantes y Finales _____	74
3.1.2.1. Catalogo de Usuarios _____	74
3.2. Especificación de Requisitos _____	75
3.2.1. Introducción _____	75
3.2.1.1. Objetivos y Alcance del Sistema _____	75
3.2.1.2. Referencias _____	76
3.2.1.3. Visión General _____	76
3.2.2. Descripción General _____	76
3.2.2.1. Funciones del Sistema _____	76
3.2.2.1.1. Gestionar Proyectos _____	76

---

---

	Página
3.2.2.1.2. Aplicar algoritmo SOM sobre los datos _____	77
3.2.2.1.3. Aplicar algoritmo C5.0 sobre la salida de SOM _____	77
3.2.2.2. Diagrama de Contexto _____	77
3.2.2.3. Suposiciones y dependencias _____	78
3.2.2.3.1. Suposiciones _____	78
3.2.2.3.2. Dependencias _____	78
3.2.3. Requisitos Específicos _____	78
3.2.3.1. Requisitos Funcionales _____	78
3.2.3.1.1. Requisitos funcionales Generales _____	78
3.2.3.1.1.1. Ingreso al Sistema (RF01) _____	78
3.2.3.1.1.2. Salir del Sistema (RF02) _____	79
3.2.3.1.2. Requisitos funcionales relacionados con Usuarios _____	79
3.2.3.1.2.1. Agregar Usuario (RF03) _____	79
3.2.3.1.2.2. Modificar Usuario (RF04) _____	80
3.2.3.1.3. Requisitos funcionales relacionados con Proyectos _____	81
3.2.3.1.3.1. Crear Proyecto (RF05) _____	81
3.2.3.1.3.2. Eliminar Proyecto (RF06) _____	81
3.2.3.1.3.3. Listar Proyectos (RF07) _____	82
3.2.3.1.4. Requisitos funcionales relacionados con Versiones _____	82
3.2.3.1.4.1. Crear Versiones de un Proyecto (RF08) _____	82
3.2.3.1.4.2. Abrir Versión (RF09) _____	83
3.2.3.1.4.3. Eliminar Versión (RF10) _____	84
3.2.3.1.4.4. Listar Versiones (RF11) _____	84
3.2.3.1.5. Requisitos funcionales relacionados con la Explotación de Datos _____	85

---

	Página
3.2.3.1.5.1. Establecer Parámetros de SOM (RF12) _____	85
3.2.3.1.5.2. Establecer Parámetros de C5.0 (RF13) _____	86
3.2.3.1.5.3. Ejecutar Proceso de Explotación de Datos (RF14) _____	86
3.2.3.2. Requisitos de Usuario _____	87
3.2.3.3. Requisitos Tecnológicos _____	87
3.2.3.4. Requisitos de Interfaces Externos _____	88
3.2.3.4.1. Interfaces de Usuario _____	88
3.2.3.4.2. Interfaces Hardware _____	88
3.2.3.4.3. Interfaces Software _____	88
3.2.3.4.4. Interfaces de Comunicación _____	88
3.2.3.5. Requisitos de Rendimiento _____	88
3.2.3.6. Requisitos de Desarrollo _____	88
3.2.3.7. Restricciones de Diseño _____	88
3.2.3.7.1. Ajuste a estándares _____	88
3.2.3.7.2. Limitaciones Hardware _____	89
3.2.3.8. Atributos _____	89
3.2.3.8.1. Seguridad _____	89
3.2.3.8.2. Mantenimiento _____	89
3.2.3.9. Otros requisitos _____	90
3.2.3.9.1. Base de datos _____	90
3.2.3.9.2. Operaciones _____	90
3.2.3.9.3. Adaptación al entorno _____	90
3.2.3.9.4. Políticas de Borrado _____	90
3.3. Identificación de Subsistemas de Análisis _____	90

---

---

	Página
3.4. Establecimiento de Requisitos _____	91
3.4.1. Especificación de Casos de Uso _____	91
3.4.1.1. Casos de Uso _____	92
3.4.2. Análisis de los Casos de Uso _____	111
3.4.2.1. Identificación de Clases Asociadas a un Caso de Uso _____	111
3.4.2.1.1. Identificación de clases _____	111
3.4.2.2. Descripción de la Interacción de Objetos _____	111
3.4.2.2.1. Identificación de la Relación entre Objetos _____	112
3.4.3. Análisis de Clases _____	117
3.4.3.1. Identificación de Responsabilidades y Atributos _____	117
3.4.3.1.1. Definición de Responsabilidades y Atributos _____	118
3.4.3.2. Identificación de Asociaciones y Agregaciones _____	120
3.4.3.2.1. Diagrama de Clases donde se identifican Asociaciones y Agregaciones _____	120
3.5. Especificación de Interfaz de Usuario _____	121
3.5.1. Especificación de Principios Generales de la Interfaz _____	123
3.5.1.1. Principios Generales de la Interfaz _____	124
3.5.2. Especificación de Formatos Individuales de la Interfaz de Pantalla _____	124
3.5.2.1. Especificación del Comportamiento Dinámico de la Interfaz _____	125
3.5.2.1.1. Descripción de las características generales de cada pantalla _____	126
3.5.3. Prototipos de Interfaz _____	129
3.5.4. Especificación de Formatos de Impresión _____	145
3.5.4.1. Formatos de Impresión _____	146
3.6. Análisis de Consistencia y Especificación de Requisitos _____	151
3.6.1. Verificación de los modelos _____	151

---

---

	Página
3.6.2. Verificación de Modelos _____	151
3.6.3. Análisis de Consistencia entre métodos _____	152
3.6.3.1. Análisis de Consistencia _____	155
3.7. Aprobación del Análisis del Sistema de Información _____	167
3.7.1. Presentación y Aprobación del Análisis del Sistema de Información _____	167
CAPÍTULO 4 – DISEÑO DEL SISTEMA DE INFORMACIÓN _____	168
4.1. Definición de la Arquitectura del Sistema _____	172
4.1.1. Definición de Niveles de Arquitectura _____	174
4.1.1.1. Descripción de los Niveles de Arquitectura del Sistema _____	174
4.1.2. Identificación de Requisitos de Diseño y Construcción _____	178
4.1.2.1. Descripción de los requisitos adicionales _____	178
4.1.3. Especificaciones de Excepción _____	179
4.1.3.1. Catalogo de Excepciones _____	180
4.1.4. Identificación de Subsistemas de Diseño _____	183
4.1.4.1. Subsistemas de Diseño _____	185
4.1.5. Especificación del Entorno Tecnológico _____	185
4.1.5.1. Especificación de Hardware _____	186
4.1.5.2. Especificación de Software _____	187
4.1.5.3. Especificación de Comunicación _____	187
4.1.6. Especificación de Requisitos de Operación y Seguridad _____	187
4.1.6.1. Acceso al sistema y a sus recursos _____	188
4.1.6.2. Mantenimiento de la integridad y confidencialidad de los datos _____	189
4.1.6.3. Control y registro de acceso al sistema _____	189

---

---

	Página
4.1.6.4. Copia de seguridad y recuperación de datos y su periodicidad _____	189
4.1.6.5. Recuperación y reanudación de trabajos _____	190
4.2. Diseño de la Arquitectura de Soporte _____	190
4.2.1. Diseño de Subsistemas de Soporte _____	191
4.3. Diseño de Casos de Uso Reales _____	192
4.3.1. Identificación de Clases Asociadas a un Caso de Uso _____	193
4.3.1.1. Clases Asociadas a los Casos de Usos _____	193
4.3.2. Revisión de la Interfaz de Usuario _____	198
4.3.2.1. Resultado de la revisión de la Interfaz de Usuario _____	199
4.4. Diseño de Clases _____	199
4.4.1. Identificación de Atributos de las Clases _____	200
4.4.1.1. Atributos de Clases _____	201
4.4.2. Identificación de Operaciones de las Clases _____	204
4.4.2.1. Operaciones de Clases _____	204
4.4.3. Diagrama de Clases de Diseño _____	211
4.4.4. Especificación de Necesidades de Migración y Carga Inicial de Datos _____	215
4.4.4.1. Carga Inicial de Datos _____	215
4.5. Diseño Físico de Datos _____	215
4.5.1. Diseño del Modelo Físico de Datos _____	216
4.5.1.1. Diseño del modelo físico de datos _____	217
4.6. Verificación y Aceptación de la Arquitectura del Sistema _____	219
4.6.1. Verificación de la Especificación de Diseño _____	220
4.6.1.1. Resultado de la Verificación de la Especificación de Diseño _____	220
4.6.2. Análisis de Consistencia de las Especificaciones de Diseño _____	220

---

---

	Página
4.6.2.1. Consistencia de las Especificaciones de Diseño _____	221
4.7. Aceptación de la Arquitectura del Sistema _____	224
4.8. Generación de Especificaciones de Construcción _____	224
4.8.1. Especificación del Entorno de Construcción _____	225
4.8.2. Definición de Componentes y Subsistemas de Construcción _____	226
4.8.2.1. Componentes y Subsistemas de Construcción _____	227
4.9. Especificación Técnica del Plan de Pruebas _____	228
4.9.1. Especificación del Entorno de Pruebas _____	230
4.9.1.1. Entorno de Pruebas _____	230
4.9.2. Revisión de la Planificación de Pruebas _____	231
4.9.2.1. Planificación de Pruebas _____	231
4.9.2.2. Pruebas Unitarias _____	232
4.9.2.3. Pruebas de Integración _____	233
4.9.2.4. Pruebas del Sistema _____	234
4.10. Aprobación del Diseño del Sistema de Información _____	255
4.10.1. Presentación y Aprobación del Diseño del Sistema de Información _____	255
CAPÍTULO 5 – CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN _____	256
5.1. Preparación del Entorno de Generación y Construcción _____	257
5.1.1. Implantación de la Base de Datos Física o Ficheros _____	258
5.1.1.1. Creación de la Base de Datos _____	258
5.1.2. Preparación del Entorno de Construcción _____	259
5.1.2.1. Generación del Entorno de trabajo _____	259
5.2. Generación del Código de los Componentes y Procedimientos _____	259

---

	Página
5.2.1. Generación del Código de Componentes _____	260
5.2.2. Generación del Código de los Procedimientos de Operación y Seguridad ____	260
5.2.2.1. Generación de Procedimientos de Operación y Seguridad _____	260
5.3. Ejecución de las Pruebas Unitarias _____	261
5.3.1. Realización y Evaluación de las Pruebas Unitarias _____	261
5.3.1.1. Resultado de la Realización de las Pruebas Unitarias _____	262
5.4. Ejecución de las Pruebas del Sistema _____	262
5.4.1. Realización de las Pruebas del Sistema _____	262
5.4.1.1. Resultado de la Realización de las Pruebas de Sistema _____	263
5.4.2. Evaluación del Resultado de las Pruebas del Sistema _____	264
5.4.2.1. Resultado de la Evaluación de los Resultados de las Pruebas de Sistema ____	264
5.5. Aprobación del Sistema de Información _____	264
CAPÍTULO 6 – IMPLANTACIÓN Y ACEPTACIÓN DEL SISTEMA _____	265
6.1. Establecimiento del Plan de Implantación _____	266
6.1.1. Definición del Plan de Implantación _____	267
6.1.1.1. Formación de usuarios finales y equipo de pruebas _____	268
6.1.1.2. Preparación de la infraestructura necesaria para la incorporación del sistema al entorno de operación _____	268
6.1.1.3. Ejecución de carga inicial _____	269
6.1.1.4. Realización de las pruebas de implementación y aceptación del sistema ____	269
6.1.1.5. Formalización del plan de mantenimiento _____	269
6.1.2. Especificación del Equipo de Implantación _____	270
6.1.2.1. Equipo de Implantación _____	270
6.2. Incorporación del Sistema al Entorno de Operación _____	271

---

---

	Página
6.2.1. Preparación de la Instalación _____	272
6.2.1.1. Descripción de la Instalación _____	272
6.2.2. Realización de la Instalación _____	273
6.2.2.1. Instalación del sistema _____	274
6.3. Carga de Datos al Entorno de Operación _____	274
6.3.1. Migración y Carga inicial de Datos _____	274
6.3.1.1. Instalación del sistema _____	274
6.4. Pruebas de Implantación del Sistema _____	275
6.4.1. Preparación de las Pruebas de Implantación _____	275
6.4.1.1. Pruebas de Implantación _____	275
6.4.2. Realización de las Pruebas de implantación _____	276
6.4.2.1. Prueba de Implantación _____	276
6.4.3. Evaluación del Resultado de las Pruebas de Implantación _____	276
6.4.3.1. Evaluación de la Prueba de Implantación _____	276
6.5. Pruebas de Aceptación del Sistema _____	276
6.5.1. Realización de las Pruebas de Aceptación _____	277
6.5.1.1. Pruebas de Aceptación _____	277
6.6. Presentación y Aprobación del Sistema _____	277
CAPÍTULO 7 – CONCLUSIÓN Y FUTURAS LÍNEAS DE TRABAJO _____	278
7.1. Conclusión _____	278
7.2. Futuras Líneas de Trabajo _____	279
CAPÍTULO 8 – BIBLIOGRAFÍA Y GLOSARIO DE TÉRMINOS _____	280

---

---

	Página
8.1. Bibliografía _____	280
8.2. Glosario _____	280
ANEXO I – CONTROL DE CONFIGURACIÓN _____	282
ANEXO II – PLAN DE ACCIÓN Y CONTINGENCIA FRENTE AL RIESGO ____	284
II.1. Riesgo R01 Falla de seguridad en la aplicación _____	284
II.1.1 Aspectos a considerar _____	284
II.1.2 Plan de Acción _____	284
II.1.3 Plan de Contingencia _____	285
ANEXO III – ASEGURAMIENTO DE CALIDAD _____	286
III.1. Listas de verificación _____	286
III.2. Plan de Pruebas de Sistema _____	288
III.2.1. Plan de Pruebas _____	289
III.2.1.1. Introducción _____	289
III.2.1.2. Alcance _____	290
III.2.1.3. Ítems y características a probar _____	290
III.2.1.4. Características que no van a ser probadas _____	290
III.2.2. Actividades a Realizar _____	290
III.2.2.1. Pruebas unitarias _____	291
III.2.2.2. Pruebas de sistema _____	291
III.2.2.3. Pruebas de aceptación _____	291
III.2.2.4. Pruebas unitarias _____	291

---

---

	Página
III.2.2.5. Pruebas de sistema _____	292
III.2.2.6. Amplitud de las pruebas _____	292
III.2.3. Reporte de fallas de las pruebas _____	292
III.2.4. Trazabilidad de requerimientos _____	293
III.2.4.1. Disponibilidad de ítems de prueba _____	293
III.2.4.2. Disponibilidad de recursos para las pruebas _____	293
III.2.5. Criterio de Paso/Falla _____	293
III.2.5.1. Ítems _____	293
III.2.5.2. Aplicación _____	294
III.2.6. Criterio de suspensión y reiniciación de pruebas _____	294
III.2.7. Artefactos de las pruebas _____	295
III.2.8. Actividades de prueba _____	295
III.2.9. Procedimiento de pruebas _____	295
III.2.10. Necesidades de ambiente _____	297
III.2.10.1. Hardware _____	297
III.2.10.2. Software _____	297
III.2.10.3. Seguridad _____	297
III.2.10.4. Modo de uso _____	297
III.2.10.5. Certificación de entorno _____	297
III.2.11. Responsabilidades _____	297
III.2.12. Riesgos y contingencias de pruebas _____	299

## CAPÍTULO 1 – PLAN DE SISTEMAS DE INFORMACIÓN

### 1.1. Introducción

La Explotación de datos es el proceso de extraer información previamente **desconocida**, **valida** y **realizable** de grandes bases de datos y entonces utilizar la información para efectuar decisiones de negocio cruciales. Esta definición nos llevan a la naturaleza esencial de la explotación de datos y ayuda a explicar las diferencias fundamentales entre ella y el análisis de datos tradicional, como las consultas, los reportes y el procesamiento analítico en línea (OLAP). Esencialmente la explotación de datos se distingue del análisis tradicional por el hecho que su objetivo es descubrir información, sin una hipótesis previamente formulada.

Primero, la información descubierta debe ser previamente **desconocida**. Aunque esto suena obvio, el asunto principal aquí es que debe ser poco probable que la información pueda haber sido hipotetizada de entrada, esto es, el que realiza explotación de datos esta buscando algo que no es intuitivo, o tal vez, es aun contraintuitivo.

Segundo, la información nueva debe ser **valida**, esto es que los datos deben estar libres de errores de codificación, de medida, de tipos, etc. Este elemento de la definición se relaciona con el problema de sobre optimismo en explotación de datos. Esto es, si el que efectúa la minería de datos realiza arduas búsquedas en grandes bases de datos, esta destinado a obtener algo de interesante tarde o temprano. La posibilidad de resultados espurios aplica a todas las minería de datos y resalta la constante necesidad de validación post minería y chequeo de sanidad.

Tercero, y en forma más crítica, la información debe ser **realizable**, esto es, debe ser posible de traducirla en alguna ventaja de negocio. En muchos casos, sin embargo, el criterio **realizable**, no es tan simple. Igualmente, la explotación de aparentes oportunidades pueden requerir de uso de datos que no están disponibles o que no se pueden usar por una cuestión legal. No hace falta decir, una organización debe tener la voluntad política para llevar a cabo la acción implicada por la minería. [Cabena, P., *et al.*; 1998]

Con la creciente capacidad y disponibilidad de almacenamiento en las organizaciones, para realizar tareas de explotación de datos es necesario contar con herramientas que tengan la capacidad de aplicar a los datos ciertos algoritmos. Este trabajo trata de desarrollar, siguiendo la metodología de Métrica V3, una herramienta que provea de los mecanismos necesarios en conjunción con el criterio del usuario para intentar que se cumplan las tres premisas anteriores.

## **1.2. Aproximación inicial al problema**

Existen en la actualidad herramientas que realizan operaciones de explotación de datos aplicando técnicas de mapas auto-organizados y algoritmos de inducción como puede ser C5.0, como las que se listan en la tabla PSI 1.1, siendo estas bastantes representativas de este mercado en particular, pero no se pudo encontrar una que pudiera ejecutar C5.0 a partir de los datos de mapas auto-organizados en un solo paso.

Fundamentalmente lo que se busca es tratar de poder resolver problemas de explotación de datos con mapas auto-organizados o el algoritmo de inducción C5.0 en forma independiente o conjunta. Obteniendo los datos de diferentes fuentes de datos, como ser una tabla en una base de datos o bien un archivo plano en nuestra PC. Es importante que las interfaces de usuarios sean lo mas amigables posible, además debe contener un esquema de seguridad para resguardar a cada usuario, los datos del proyecto en que se encuentra trabajando.

## **1.3. Situación actual del Dominio**

Las redes neuronales, Perceptrón, Adaline, Backpropagation, se comportan mejor en situaciones en las cuales la salida no es lineal o con problemas complejos en que no requieren una solución de absoluta precisión en comparación con otras metodologías. Las redes neuronales también son muy adecuadas en situaciones donde una gran cantidad de datos históricos están disponibles para entrenamiento y donde se efectuará una solución de “caja negra” y no es requerida una explicación. En 1981

Teuvo Kohonen propuso una nueva arquitectura de redes neuronales, llamada mapas auto organizados (Self Organizing Maps – SOM). Una red SOM se configura como una colección de neuronas, como un cerebro humano, cada una conectada a su vecino. A medida que SOM es expuesto a muestras, comienza a crear clusters autoorganizados. Gradualmente, una red de Kohonen organizara clusters y puede de esta manera ser usada en situaciones donde se desconoce la salida y la variable dependiente. SOM involucra al aprendizaje no supervisado, una muestra de entrenamiento para la cual no se conoce la salida, y es comúnmente usada para descubrir relaciones en un conjunto de datos.

El uso de SOM para descubrir clusters en sus datos puedes a menudo llevarnos a un problema al tratar de extrapolar un significado para cada agrupamiento. Hay dos maneras de solucionar este problema. La primera se refiere a la obtención del valor medio para cada característica en los datos. El otro método requiere el paso de cada cluster a través de un algoritmo de aprendizaje automático (como C5.0) con el fin de generar un conjunto de reglas que describan las características de ese agrupamiento. *[Mena, J.; 1999]*

El algoritmo de inducción C5.0 forma parte de la familia de los TDIDT (Top Down Induction Trees), junto con sus antecesores el ID3 y C4.5 (el primero en la década de los ochenta y el segundo en 1993), los mismos fueron desarrollados por Quinlan, son sistemas de aprendizaje supervisado que construyen árboles de decisión a partir de un conjunto de ejemplos. Estos ejemplos o tuplas están constituidos por un conjunto de atributos y un clasificador o clase. *[García Martínez et al, 2003]*

Este sistema tiene una buena performance con respecto a otros algoritmos en un amplio rango de aplicaciones de diversos dominios, como el dominio médico, el artificial y el análisis de juegos de ajedrez. El nivel de precisión en la clasificación generalmente es alto. A veces, los árboles son demasiado frondosos, lo cual conlleva una difícil interpretación. En esos casos pueden ser transformados en reglas de decisión para hacerlos más comprensibles.

El algoritmo de inducción C5.0 utiliza la estrategia de "divide y reinarás" para

generar el árbol de decisión inicial a partir de un conjunto de datos de entrenamiento. El método "divide y reinarás" realiza en cada paso una partición de los datos del nodo según una prueba realizada sobre el "mejor" atributo. Cualquier prueba que divida a  $T$  en una manera no trivial, tal que al menos dos subconjuntos distintos  $\{T_i\}$  no estén vacíos, eventualmente resultará en una partición de subconjuntos de una única clase, aún cuando la mayoría de los subconjuntos contengan un solo ejemplo. Sin embargo, el proceso de construcción del árbol no apunta meramente a encontrar cualquier partición de ese tipo, sino a encontrar un árbol que revele una estructura del dominio y, por lo tanto, tenga poder predictivo. Para ello, necesitamos un número importante de casos en cada hoja y cada partición debe tener la menor cantidad de clases posibles. En el caso ideal, nos gustaría elegir en cada paso la prueba que genere el árbol más pequeño.

Los árboles de decisión demasiado grandes son difíciles de entender porque cada nodo debe ser interpretado dentro del contexto fijado por las ramas anteriores. En cualquier árbol de decisión, las condiciones que deben satisfacerse cuando un caso se clasifica por una hoja pueden encontrarse analizando los resultados de las pruebas en el camino recorrido desde la raíz. Es más, si el camino fuese transformado directamente en una regla de producción, dicha regla podría ser expresada como una conjunción de todas las condiciones que deben satisfacerse para llegar a la hoja. Consecuentemente, todos los antecedentes de las reglas generadas de esta manera son mutuamente excluyentes y exhaustivos.

Para pasar a reglas de decisión, se recorre el árbol desde la raíz hasta las hojas en preorden (de raíz a hojas, de izquierda a derecha) y genera una regla por cada camino recorrido. El antecedente de cada regla estará compuesto por la conjunción de las pruebas de valor de cada nodo visitado y la clase será la correspondiente a la hoja. *[García Martínez et al; 2003]*

Los algoritmos previamente nombrados, existen actualmente en forma separada en herramientas de explotación de datos disponibles en el mercado, pero no de forma integrada, de tal manera que puedan ser ejecutados uno a partir de la salida de otro, utilizando la salida de SOM para servir de entrada para el algoritmo de C5.0.

#### 1.4. Análisis Comparativo entre las herramientas de explotación de datos disponibles en el mercado

A continuación se presenta una comparación de herramientas de distintas herramientas existentes en el mercado (tabla PSI 1.1):

Herramienta	Fabricante	Algoritmos soportados	Web Site	Breve descripción
Knowledge SEEKER	AGNOSS	ID3	www.angoss.com	El corazón de KnowledgeSEEKER es su de proceso de inducción del árbol de decisión que tiene el poder para determinar las dependencias entre las variables y mostrar los resultados en un árbol de decisión visual y muy interactivo. Usted no sólo podrá identificar las relaciones significativas sino que descubrirá a menudo nuevas conexiones permitiendo formar conclusiones profundas y determinar acciones significativas para mejorar su negocio.
DataEngine	MIT	SOM	www.dataengine.de	DataEngine es la herramienta de software para análisis inteligente de datos que une métodos estadísticos con redes neuronales y con tecnología fuzzy.
DecisionHouse	Quadstone	ID3	www.quadstone.com/system/decision	Decisionhouse™ es el producto de software insignia de Quadstone el cual proporciona un ambiente único para una rápida exploración de las oportunidades del cliente. Decisionhouse proporciona un lenguaje compartido para permitir la fusión del conocimiento del negocio con habilidades analíticas y los datos. Una solución empaquetada fácil de utilizar para analizar y predecir comportamiento, Decisionhouse se ha diseñado y desarrollado desde el principio específicamente para el análisis del cliente. Integrando modelado interactivo de alta productividad con la mejor visualización de cuentas rápidas, Decisionhouse es único en velocidad y en escala en su soporte para la exploración y predicciones de comportamiento del cliente.

Tabla PSI 1.1: Comparación de herramientas

Herramienta	Fabricante	Algoritmos soportados	Web Site	Breve descripción
Clementine	SPSS	CHAID, Exhaustive CHAID, QUEST, Kohonen, K-Means, etc.	<a href="http://www.spss.com/clementine">www.spss.com/clementine</a>	<p>Con Clementine de SPSS Inc descubra patrones y asociaciones en los datos de su organización y tome las decisiones que lo lleven a resultados significativos y mejoras mensurables. Clementine es el líder en workbench, reconocido en todo el mundo igualmente entre los que realizan datamining y los analistas del negocio, y es el único que soporta el proceso entero de datamining. Con Clementine y sus productos asociados de SPSS, los usuarios pueden fácilmente tener acceso y elaborar datos numéricos, de texto, y de Web para modelar; construya y compare rápidamente los modelos; y desarrolle eficientemente, en tiempo real, para personas y sistemas que realizan la toma de decisiones y recomendaciones. Porque le permite fácilmente integrar resultados de datamining con otros sistemas y procesos del negocio, Clementine ayuda a su organización a tomar decisiones más rápidas, mejores, para toda la empresa. Las personas que realizan el datamining puede realizar un análisis natural del pensamiento, gracias a la poderosa interfaz visual de Clementine. Clementine produce "corrientes", mapas visuales de cada paso en el proceso de datamining.</p> <p>Interactuando con una corriente, los analistas pueden agregar conocimientos claves del negocio para un datamining exitoso en cualquier punto en el proceso. Porque la interfaz permite que los analistas se centren en descubrimiento del conocimiento más bien que en la ejecución de tareas técnicas tales como la escritura de código, su proceso datamining es mucho mas eficiente. Con Cementine 9.0, puede mejorar la productividad en datamining, gracias a las capacidades de SPSS Model Manager. Integrado completamente con la Clementine, el Model Manager le permite mejorar el conocimiento del negocio de toda su organización. Los que realizan el datamining guardan corrientes, modelos, y archivos de salida en un repositorio central. Los usuarios autorizados pueden entonces acceder y reutilizar a las corrientes y modelos más eficaces.</p> <p>Los administradores pueden asegurarse fácilmente de que solamente aquellos con los permisos suficientes puedan ver e interactuar con las corrientes y los modelos. El SPSS Model Manager también incluye una característica versioning, así que usted puede confiar que las versiones de producción no serán sobrescritas. Con el SPSS Model Manager, usted tiene completo control de sus activos de datamining.</p>

Tabla PSI 1.1 (continuación): Comparación de herramientas

Herramienta	Fabricante	Algoritmos soportados	Web Site	Breve descripción
Intelligent Miner	IBM	SOM, ID3	www.software.ibm.com/d ata/intelli-mine	Las capacidades de datamining de IBM le ayudan a detectar fraudes, a dividir a sus clientes en segmentos, y a simplificar análisis de canasta de mercado. Las capacidades datamining de IBM se integran con sus sistemas existentes para proporcionar escalabilidad, análisis de alta performance sin migrar sus datos a una plataforma propietaria de datamining. Utilice SQL, Web Services, o Java para acceder a las capacidades de datamining de DB2's desde sus propias aplicaciones o desde las herramientas de inteligencia de negocio de los partners de IBM.

Tabla PSI 1.1 (continuación): Comparación de herramientas

## 1.5. Identificación de requisitos iniciales

### 1.5.1. Análisis de las Necesidades de Información

La información que el analista del negocio debe ubicar a partir de los datos de entrada, son o bien cluster con agrupamientos de datos o un árbol de decisión, por lo cual a las columnas de las tablas a analizar debo pasarlas por uno de los algoritmos propuestos, SOM y acto seguido exponer el resultado o si no, esa salida debe ser la entrada de C5.0, y obtener un árbol de decisión, como lo muestra la figura PSI 1.1.

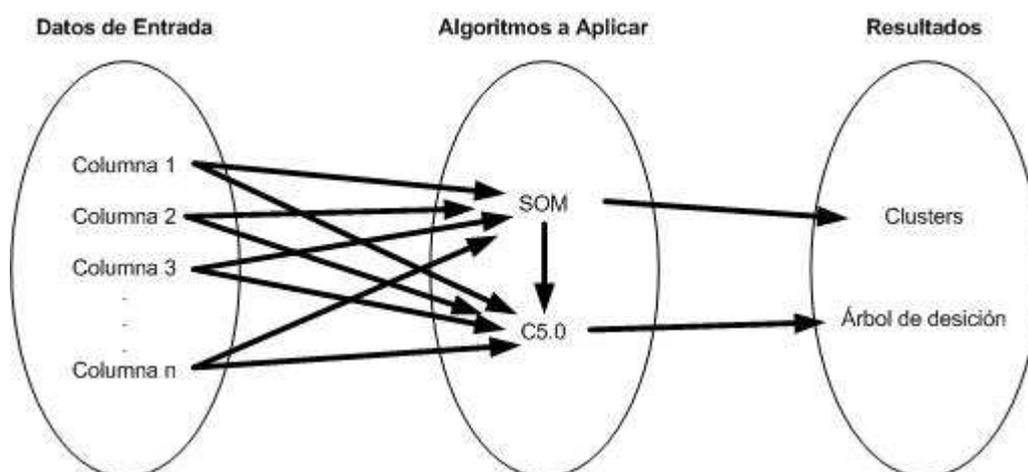


Figura PSI 1.1: Modelo de información de las principales entidades y relaciones existentes entre ellas

## 1.6. Solución propuesta

La herramienta a desarrollar como objetivo del trabajo de tesis se diferenciará de las existentes en el mercado (ver tabla PSI 1.1) en que si bien ya existen algunas que aplican algoritmos de SOM y C5.0, lo realizan de manera independiente. Esta herramienta a crear los implementará en forma continua una metodología a partir de los datos de la otra (aplicar C5.0 a la salida de SOM), partiendo de los datos de entrada, pasando por SOM, hasta llegar a reglas de inducción, sin que esto se realice en varias etapas. En un principio se establecen las fuentes de datos, los algoritmos a aplicar, pudiéndose elegir si se desea solo uno, y los parámetros de los mismos y a continuación la herramienta realiza el procesamiento en forma conjunta y continua.

Será una aplicación del tipo standalone, ejecutable bajo el sistema operativo Microsoft Windows y será testeado para las versiones 2000 y XP. La misma será desarrollada con la herramienta de programación Microsoft Visual FoxPro Versión 9.0, lenguaje que soporta orientación a objetos y que además posee una gran ductilidad en el manejo de datos, elemento fundamental en este proyecto. Para la construcción de la herramienta se utilizará la metodología Métrica Versión 3.

El producto final permitirá al usuario aplicar SOM y C5.0 a un conjunto de datos, ubicado en algún repositorio de datos (local o remoto), y el resultado ubicarlo en una tabla en el motor de bases de datos o bien en un archivo plano en su estación de trabajo.

El mismo contará con una interfaz gráfica que permita:

- a) Elegir las fuentes de datos, como un archivo plano al cual se accederá vía ODBC, utilizando SQL ANSI,
- b) Visualizar las entradas y salidas de datos en forma tabular
- c) Aplicar a los datos las técnicas de SOM y C5.0 ya sea en forma conjunta o separada.
- d) Graficar los resultados obtenidos con SOM y C5.0.

- e) Obtener reportes tanto por pantalla como por impresora,
- f) Permitir la exportación de datos a archivos planos y Microsoft Excel.

### **1.6.1. Metodología para la construcción del sistema**

A los efectos de desarrollar el presente proyecto se ha resuelto adoptar la metodología Métrica Versión 3 promovida por el Consejo Superior de Informática del Ministerio para las Administraciones Públicas del gobierno del Reino de España (órgano interministerial responsable de la política informática del gobierno español).

Esta metodología ofrece a las Organizaciones un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software dentro del marco que permite alcanzar los siguientes objetivos:

- Proporcionar o definir Sistemas de Información que ayuden a conseguir los fines de la Organización mediante la definición de un marco estratégico para el desarrollo de los mismos.
- Dotar a la Organización de productos software que satisfagan las necesidades de los usuarios dando una mayor importancia al análisis de requisitos.
- Mejorar la productividad de los departamentos de Sistemas y Tecnologías de la Información y las Comunicaciones, permitiendo una mayor capacidad de adaptación a los cambios y teniendo en cuenta la reutilización en la medida de lo posible.
- Facilitar la comunicación y entendimiento entre los distintos participantes en la producción de software a lo largo del ciclo de vida del proyecto, teniendo en cuenta su papel y responsabilidad, así como las necesidades de todos y cada uno de ellos.
- Facilitar la operación, mantenimiento y uso de los productos software obtenidos.

La nueva versión de MÉTRICA contempla el desarrollo de Sistemas de Información para las distintas tecnologías que actualmente están conviviendo y los aspectos de gestión que aseguran que un Proyecto cumple sus objetivos en términos de calidad, coste y plazos.

En la elaboración de MÉTRICA Versión 3 se han tenido en cuenta los métodos de desarrollo más extendidos, así como los últimos estándares de ingeniería del software y calidad, además de referencias específicas en cuanto a seguridad y gestión de proyectos. También se ha tenido en cuenta la experiencia de los usuarios de las versiones anteriores para solventar los problemas o deficiencias detectados.

En una única estructura la metodología MÉTRICA Versión 3 cubre distintos tipos de desarrollo: estructurado y orientado a objetos, facilitando a través de interfaces la realización de los procesos de apoyo u organizativos: Gestión de Proyectos, Gestión de Configuración, Aseguramiento de Calidad y Seguridad.

La automatización de las actividades propuestas en la estructura de MÉTRICA Versión 3 es posible ya que sus técnicas están soportadas por una amplia variedad de herramientas de ayuda al desarrollo.

### **1.6.2 Selección del Ciclo de Vida**

Para la selección del ciclo de vida del producto software se han tenido en cuenta los siguientes factores:

- No se conocen desde el principio todos los requisitos, y se necesita experimentar con un sistema operacional para aprender cuáles son.
- Los requisitos de fiabilidad y rendimiento se deben implementar desde el principio.
- Centrarse en la funcionalidad y el contenido.
- Evolución continua hacia una solución.
- Establecer un primer prototipo como un marco para el resto del desarrollo.

- Refinar las funciones de la versión anterior, con el fin de acercarse cada vez más a las necesidades del usuario.
- Necesidad de obtener resultados tangibles en un tiempo reducido.

Debido a los factores expuestos previamente se elige como ciclo de vida a, “Prototipado Evolutivo” pues es que se acerca mas a nuestras aspiraciones en este proyecto.

## **CAPÍTULO 2 – ESTUDIO DE VIABILIDAD DEL SISTEMA**

### **2.1. Introducción**

El objetivo de este proceso es el análisis del conjunto concreto de necesidades para proponer una solución que tenga en cuenta las restricciones económicas, técnicas, legales y operativas aplicables a la solución. La solución que se obtendrá de este estudio podrá constituir la definición de un nuevo producto software que realizará operaciones con datos a fin de obtener resultado a través de técnicas de minería de datos, para un posterior análisis e interpretación de resultados.

### **2.2. Análisis de la Situación y Restricciones**

Existen en la actualidad herramientas que realizan operaciones de explotación de datos aplicando técnicas de mapas auto-organizados y algoritmos de inducción como puede ser C5.0, como las que se listan en la tabla PSI 1.1, siendo estas bastantes representativas de este mercado en particular, pero no se pudo encontrar una que pudiera ejecutar C5.0 a partir de los datos obtenidos de mapas auto-organizados en un solo paso.

Fundamentalmente lo que se busca es tratar de poder resolver problemas de explotación de datos con mapas auto-organizados o el algoritmo de inducción C5.0 en forma independiente o conjunta. Obteniendo los datos de diferentes fuentes de datos, como puede ser una tabla en una base de datos o bien un archivo plano en nuestra PC. Es importante que las interfaces de usuarios sean lo mas amigables posible, además debe contener un esquema de seguridad para resguardar a cada usuario, los datos del proyecto en que se encuentra trabajando.

Se tratará de estudiar el alcance de la necesidad planteada por el proyecto, analizando las posibles restricciones que pueden afectar al nuevo producto. Deberán considerarse como antecedentes las aproximaciones iniciales establecidas en el Capítulo 1 Plan de Sistemas de Información.

En el proyecto se han presentado los siguientes requerimientos de información:

- 1) Elegir las fuentes de datos, como un archivo plano al cual se accederá vía ODBC, utilizando SQL ANSI,
- 2) Visualizar las entradas y salidas de datos en forma tabular
- 3) Aplicar a los datos las técnicas de SOM y C5.0 ya sea en forma conjunta o separada.
- 4) Graficar los resultados obtenidos con SOM y C5.0.
- 5) Obtener reportes tanto por pantalla como por impresora,
- 6) Permitir la exportación de datos a archivos planos y Microsoft Excel

### **2.3. Descripción General del Sistema**

El sistema fundamentalmente permitirá a los usuarios aplicar las técnicas de explotación de datos SOM y C5.0, en forma independiente o una a partir de la anterior. Esta herramienta a crear los implementará en forma continua una metodología a partir de los datos de la otra (aplicar C5.0 a la salida de SOM), partiendo de los datos de entrada, pasando por SOM, hasta llegar a reglas de inducción, sin que esto se realice en varias etapas. En un principio se establecen las fuentes de datos, los algoritmos a aplicar, pudiéndose elegir si se desea solo uno, y los parámetros de los mismos y a continuación la herramienta realiza el procesamiento en forma conjunta y continua.

Será una aplicación del tipo standalone, ejecutable bajo el sistema operativo Microsoft Windows y será testeado para las versiones 2000 y XP. La misma será desarrollada con la herramienta de programación Microsoft Visual FoxPro Versión 9.0, lenguaje que soporta orientación a objetos y que además posee una gran ductilidad en el manejo de datos, elemento fundamental en este proyecto. Para la construcción de la herramienta se utilizará la metodología Métrica Versión 3.

El producto final permitirá al usuario aplicar SOM y C5.0 a un conjunto de datos, ubicado en algún repositorio de datos (local o remoto), y el resultado ubicarlo en una tabla en el motor de bases de datos o bien en un archivo plano en su estación de trabajo. Además se proveerá de un esquema que facilitará la administración y versionado de proyectos. El sistema contará con ayudas en línea y un asistente para la creación de un proyecto desde cero.

#### **2.4. Objetivo del proyecto**

El objetivo del proyecto es desarrollar una herramienta que se diferenciará de las existentes en el mercado en que si bien ya existen algunas que aplican algoritmos de SOM y C5.0, ésta los implementará en forma continua uno tras otro, partiendo de los datos de entrada, pasando por SOM, hasta llegar a reglas de inducción, sin que esto se realice en varias etapas. En un principio se establecen las fuentes de datos, los algoritmos a aplicar (pudiéndose elegir solo uno) y los parámetros y a continuación la herramienta realiza el procesamiento en forma conjunta y continua.

Será una aplicación del tipo “standalone”, ejecutable bajo el sistema operativo Microsoft Windows y será testeado para las versiones 2000 y XP. La misma será desarrollada con la herramienta de programación Microsoft Visual FoxPro Versión 9.0, lenguaje que soporta orientación a objetos y que además posee una gran ductilidad en el manejo de datos, elemento fundamental en este proyecto.

El producto final permitirá al usuario aplicar SOM y C5.0 a un conjunto de datos, ubicado en algún repositorio de datos (local o remoto), y el resultado ubicarlo en una tabla en el motor de bases de datos o bien en un archivo plano en su estación de trabajo.

La misma contará con una interfaz gráfica que permita elegir las fuentes de datos, mostrar las entradas y salidas de datos en forma tabular y graficar los resultados obtenidos con SOM y el árbol de C5.0. Permitirá obtener reportes tanto por pantalla como por impresora, y la exportación de datos a archivos planos y Microsoft Excel.

En primera instancia los datos a leer o a escribir podrán estar ubicados, en un

archivo plano como se dijo anteriormente, o bien en un motor de base de datos, al cual se accederá vía ODBC, utilizando SQL ANSI, y será comprobado este esquema con el motor de bases de datos Microsoft SQL Server 2000. Cabe destacar que los datos a introducir en el sistema deben estar previamente chequeados en su limpieza y validez, por el analista del negocio.

## **2.5. Estudio de la situación actual**

Actualmente existe una amplia gama de proveedores que venden herramientas de minería de datos que utilizan las técnicas de SOM y C5.0, pero ninguna de ellas aplica los dos métodos en forma consecutiva, alimentándose es segundo con los datos del primero. Ver capítulo 1, punto 1.6 “Análisis Comparativo entre las herramientas de explotación de datos disponibles en el mercado”.

## **2.6. Definición de requisitos del sistema**

A continuación se realizará una descripción de los requisitos del sistema. Para la descripción de los requisitos funcionales se utilizará el estándar IEEE 830-1998, que se ampliará en el punto 3.2 de éste texto.

### **2.6.1. Requisitos funcionales**

#### **2.6.1.1. Requisitos funcionales Generales**

RF01 - Ingreso al Sistema

RF02 - Salir del Sistema

#### **2.6.1.2. Requisitos funcionales relacionados con Usuarios**

RF03 - Agregar Usuario

RF04 - Modificar Usuario

### **2.6.1.3. Requisitos funcionales relacionados con Proyectos**

RF05 - Crear Proyecto

RF06 - Eliminar Proyecto

RF07 - Listar Proyectos

### **2.6.1.4. Requisitos funcionales relacionados con Versiones**

RF08 - Crear Versiones de un Proyecto

RF09 - Abrir Versión

RF10 - Eliminar Versión

RF11 - Listar Versiones

### **2.6.1.5. Requisitos funcionales relacionados con la Explotación de Datos**

RF12 - Establecer Parámetros SOM

RF13 - Establecer Parámetros C5

RF14 - Ejecutar Proceso de Explotación de Datos

## **2.6.2. Requisitos de rendimiento**

### **2.6.2.1. Estáticos**

Se prevé la instalación de la herramienta en una estación de trabajo. Además se puede tanto consultar como guardar datos en tablas externas para lo cual se requiere la instalación en un servidor de base de datos y una red LAN. El acceso a datos dependerá del perfil del usuario del sistema.

### **2.6.2.2. Dinámicos**

No se prevén requisitos específicos dinámicos.

### **2.6.3. Atributos del sistema software**

#### **2.6.3.1. Fiabilidad**

Por tratarse de un sistema no-crítico se someterá al producto a un programa de pruebas empleando el método de caja negra, lo que asegurará una fiabilidad razonable. Poniendo énfasis en que los algoritmos utilizados estén correctamente implementados.

#### **2.6.3.2. Disponibilidad**

El sistema deberá asegurar una disponibilidad alta y cortos períodos de mantenimiento, en tal sentido se debe lograr un diseño modular que permita el reemplazo de módulos sin afectar el funcionamiento general del sistema.

#### **2.6.3.3. Seguridad**

La información que accederá y generará es sistema una vez en funcionamiento será de carácter sensible y poseerá gran importancia comercial. En tal sentido, será por parte del usuario el disponer de medidas de seguridad adecuadas con las salidas generadas y el tener conocimiento de los niveles de permisos asignados en las tablas de la base de datos que utilizará.

#### **2.6.3.4. Mantenimiento**

El mantenimiento de este sistema, será realizado de acuerdo a solicitudes de los usuarios, respecto de correcciones sobre eventuales fallas o bien actualización e incorporación de algoritmos de minería de datos a la herramienta.

#### **2.6.3.5. Portabilidad**

En esta etapa no se prevén requerimientos de portabilidad para este Sistema.

## 2.6.4. Catalogación de requisitos

### 2.6.4.1. Requerimientos de hardware y software

#### 2.6.4.1.1. Estación de trabajo

Identif.	Descripción	Volatilidad	Prioridad	Necesidad	Estado	Creado
RHU01	PC con Procesador clase Pentium. 64 Mb de RAM. Disco Rígido con capacidad disponible de 20 MB para el Sistema – Unidad de disquetes de 3.5” y 1.44 MB de capacidad. CD – ROM.	Baja	Alta	Alta	Incorporado	
RHU02	Las listado se efectuarán sobre impresoras tanto locales como remotas	Baja	Media	Alta	Incorporado	
RHU03	La Información generada podrá ser transferida a través de correo electrónico, de no contarse con esa capacidad existirá un proceso manual de transferencia de información mediante un soporte magnético (disquete de 3.5”- 1.4 Mb) u óptico (CD - ROM)	Baja	Alta	Alta	Incorporado	
RSU01	Sistemas operativos: Microsoft Windows 2000 con Service Pack 3 o superior. Microsoft Windows XP o superior. Microsoft Windows Server 2003 o superior.	Baja	Alta	Alta	Incorporado	

Tabla EVS 2.1 - Requerimientos de hardware y software de la Estación de trabajo

### 2.6.4.2. Requisitos no funcionales – Normas y Estándares

Identif.	Descripción	Volatilidad	Prioridad	Necesidad	Estado	Creado
RNFNU01	El Subsistema debe ser desarrollado empleando la metodología de desarrollo Métrica V-3 Ciclo de vida Prototipado Evolutivo	Baja	Alta	Alta	Incorporado	
RNFNU02	El subsistema debe presentar una interfase con el usuario tipo Windows y una apariencia (combinación de Colores) similar a la "Estándar de Windows"	Baja	Alta	Alta	Incorporado	
RNFNU03	El subsistema correrá con una configuración de pantalla 1024 x 768	Baja	Alta	Alta	Incorporado	
RNFNU04	Las interfases con los usuarios se implementarán mediante entornos visuales propios de los lenguajes de programación que se utilicen.	Baja	Alta	Alta	Incorporado	

Tabla EVS 2.2 - Requisitos no funcionales – Normas y Estándares

### 2.6.4.3. Requisitos no funcionales – Seguridad

Identif.	Descripción	Volatilidad	Prioridad	Necesidad	Estado	Creado
RNFSU01	Los usuarios deben ingresar su identificación y contraseña para tener acceso al Sistema	Baja	Alta	Alta	Incorporado	
RNFSU02	Las claves de usuario deben ser almacenadas de manera encriptada	Baja	Alta	Alta	Incorporado	
RNFSU03	Los valores de Disponibilidad de Componentes o Subsistemas deben ser guardados de manera encriptada para que los usuarios no tengan acceso a esos valores.	Baja	Alta	Alta	Incorporado	
RNFSU04	Los operadores no podrán extraer listados de estados de disponibilidad del sistema	Baja	Alta	Alta	Incorporado	

Tabla EVS 2.3 - Requisitos no funcionales – Seguridad

### 2.6.4.4. Requisitos no funcionales – Organización

Identif.	Descripción	Volatilidad	Prioridad	Necesidad	Estado	Creado
RNFOU01	Las herramientas de desarrollo de software y soporte deben ser las existentes en la organización.	Baja	Alta	Alta	Incorporado	31/08/05
RNFO02	Se requieren operadores adiestrados	Baja	Media	Media	Incorporado	31/08/05

Tabla EVS 2.4 - Requisitos no funcionales – Organización

### 2.6.4.5. Requisitos no funcionales – Backup

Identif.	Descripción	Volatilidad	Prioridad	Necesidad	Estado	Creado
RNFBU01	El Subsistema deberá poseer facilidades para realizar copias de seguridad y, en caso de contingencias, poder restaurar la información existente al momento de realizar la última copia de seguridad.	Baja	Alta	Alta	Incorporado	31/08/05

Tabla EVS 2.5 - Requisitos no funcionales – Backup

## 2.7. Catálogo de Usuarios

Para esta versión (versión 1) de la herramienta se contemplarán dos tipos de usuarios, Usuario Administrador y Usuario Analista. El Administrador es el usuario con mas responsabilidades y poder de actuación dentro del sistema, ya que son los únicos facultados para poder gestionar cualquier proyecto existente. El usuario Analista involucra a los que estarán a cargo de la ejecución de la tarea en si misma de desarrollo de proyectos. El mismo tendrá atribuciones para dar de alta nuevos proyectos, consultar el estado de cada uno de los proyectos a su cargo. Cabe destacar que los usuarios del sistema deben tener en cuenta que en ellos recae la responsabilidad de interpretar los resultados del los algoritmos implementados, siendo por lo tanto, de vital importancia su conocimiento sobre el dominio del problema involucrado.

## **2.8. Tamaño, Esfuerzo y Duración del Proyecto**

### **2.8.1. Características del Proyecto “DM”**

Es conveniente antes de avanzar en las estimaciones destacar las características particulares que se presentaron al iniciar este proyecto:

- ✓ El trabajo para este proyecto es realizado solo por el autor. El grado de compromiso y conocimiento personal es muy alto.
- ✓ El autor ha realizado numerosos sistemas con el entorno de programación escogido, por lo que puede establecer una opinión aproximada sobre la duración del proyecto.

### **2.8.2. Técnica de Estimación elegida**

A efectos de estimar el esfuerzo necesario para desarrollar este proyecto, en el marco de las características apuntadas en el párrafo anterior, se ha elegido la aplicación de una técnica que combina los Casos de Uso con el análisis de Puntos de Función [RT-ITBA, 2004].

Los Casos de Uso ayudan a describir los requerimientos funcionales de la aplicación. A partir de los requerimientos, el análisis de los Puntos de Función, permite cuantificar el tamaño de un sistema.

Ambas técnicas son independientes del lenguaje de desarrollo, las metodologías, plataformas y/o tecnologías utilizadas.

Una vez conocido el tamaño usando los Puntos de Función sin ajustar a partir de los Casos de Uso, y dado que no se cuenta con registros propios históricos de productividad informática, se estima el esfuerzo considerando el método COCOMO II (Constructive Cost Model Version II).

### **2.8.3. Desarrollo de la Técnica**

Por lo expuesto, se usarán los casos de uso detectados para el proyecto. Se sabe

que esta estimación puede ser muy imprecisa al inicio, pero da una idea general del emprendimiento, al ir avanzando, se puede ir ajustando el cálculo. Los pasos a realizar serán los siguientes:

- a. Se toma cada Caso de Uso y para cada componente del Análisis de Función, transacción o archivo, se determina su cantidad y factor de complejidad.
- b. Se obtiene el total de puntos de función.
- c. Se calcula el esfuerzo y duración mediante COCOMO II.

### 2.8.3.1. Criterios para Clasificación y Asignación de Valores Numéricos

Para el Análisis de Puntos de Función se van a clasificar y cuantificar las transacciones y los archivos conforme a los criterios establecidos en las tablas identificadas como EVS 2.6 a EVS 2.10.

#### Transacciones

- Entradas Externas: Archivos Referenciados indican el número de archivos lógicos internos mantenidos por la Entrada Externa y Elementos de Datos indican la cantidad de elementos que componen la misma.

Archivos Referenciados	Elementos de Datos		
	1-4	5-15	>15
0-1	Baja	Baja	Media
2	Baja	Media	Alta
3 o más	Media	Alta	Alta

Tabla EVS 2.6 - Clasificación de Entradas Externas

- Salidas y Consultas Externas: Archivos Referenciados indican el número de archivos lógicos internos o archivos de interfaz externas vinculados con la salida externa o consulta externa y Elementos de Datos indican la cantidad combinada de elementos que componen la entrada y la salida en uno u otro caso.

Archivos Referenciados	Elementos de Datos		
	1-5	6-19	>19
0-1	Baja	Baja	Media
2	Baja	Media	Alta
3 o más	Media	Alta	Alta

Tabla EVS 2.7 - Clasificación de Salidas y Consultas Externas

- Valores Numéricos: Cada Factor de Complejidad (baja, media o alta) se le asigna el siguiente valor

Clasificación	Valores		
	Salidas Externas	Consultas Externas	Entradas Externas
Baja	4	3	3
Media	5	4	4
Alta	7	6	6

Tabla EVS 2.8 - Asignación de valores numéricos para transacciones

## Archivos

- Archivos Lógicos Internos y Archivos de Interfaz Externos: Tipos de Registro indica un subgrupo de elementos de datos reconocibles por el usuario y Elementos de Datos indica la cantidad de elementos de datos básicos (campos únicos) que componen el archivo (ver Tabla 2.5).

Tipos de Registro	Elementos de Datos		
	1-19	20-50	>50
1	Baja	Baja	Media
2-5	Baja	Media	Alta
5 o más	Media	Alta	Alta

Tabla EVS 2.9 - Clasificación de Archivos Lógicos Internos y de Interfaz Externos

- Valores Numéricos: Cada Factor de Complejidad (baja, media o alta) para cada tipo de archivo, Archivo Lógico Externo o Archivo de Interfaz Externo es indicado en la tabla 2.6.

Clasificación	Valores		
	Salidas Externas	Consultas Externas	Entradas Externas
Baja	4	3	3
Media	5	4	4
Alta	7	6	6

Tabla EVS 2.10 - Asignación de valores numéricos para archivos

### 2.8.3.2. Estimación del Total de Puntos de Función

A continuación se presenta por cada Caso de Uso una tabla (de tabla 2.11 a la 2.24) que identifica los parámetros obtenidos durante el análisis inicial, para cada uno se indica la cantidad y su factor de complejidad (simple, media o alta) y el total de puntos de función obtenidos. Debajo de las tablas se detallan las justificaciones pertinentes.

#### 2.8.3.2.1. Caso de Uso: Ingreso al Sistema

	Complejidad			Aporte
	Baja	Media	Alta	
Entradas Externas	1			1*3
Salidas Externas				
Consultas Externas				
Archivos Lógicos Internos	1			1*7
Archivos de Interfaces Externas				
Total				10

Tabla EVS 2.11 Función Ingreso al Sistema

- ✓ Entradas Externas: Corresponde a la entrada de datos del usuario para

identificarse ante el sistema. Se consideran menos de 4 ítems referenciados.

- ✓ Archivos Lógicos Internos: Usuarios. Se considera registro lógico único y con componentes entre 1 y 19 elementos.

### 2.8.3.2.2. Caso de Uso: Salir del Sistema

	Complejidad			Aporte
	Baja	Media	Alta	
Entradas Externas	1			1*3
Salidas Externas				
Consultas Externas				
Archivos Lógicos Internos				
Archivos de Interfaces Externas				
Total				3

Tabla EVS 2.12 - Función Salir del Sistema

- ✓ Entradas Externas: Corresponde a la acción del usuario para salir del sistema. Se consideran menos de 4 ítems referenciados.

### 2.8.3.2.3. Caso de Uso: Agregar Usuario

	Complejidad			Aporte
	Baja	Media	Alta	
Entradas Externas		1		1*4
Salidas Externas				
Consultas Externas				
Archivos Lógicos Internos	1	2		1*7 +2*10
Archivos de Interfaces Externas				
Total				31

Tabla EVS 2.13 - Función Agregar Usuario

- ✓ Entradas Externas: Corresponde a la entrada de datos del usuario para darse de alta en el sistema. Se consideran entre 5 y 15 ítems referenciados.
- ✓ Archivos Lógicos Internos:
  - Usuarios. Se considera registro lógico único y con componentes entre 1 y 19 elementos.
  - Mae\_provincias. Se consideran mas de 5 registros y con componentes entre 1 y 19 elementos.
  - Mae\_paises. Se consideran mas de 5 registros y con componentes entre 1 y 19 elementos.

#### 2.8.3.2.4. Caso de Uso: Modificar Usuario

	Complejidad			Aporte
	Baja	Media	Alta	
Entradas Externas		1		1*4
Salidas Externas				
Consultas Externas				
Archivos Lógicos Internos	1	2		1*7 +2*10
Archivos de Interfaces Externas				
Total				31

Tabla EVS 2.14 - Función Modificar Usuario

- ✓ Entradas Externas: Corresponde a la modificación de datos del usuario en el sistema. Se consideran entre 5 y 15 ítems referenciados.
- ✓ Archivos Lógicos Internos:
  - Usuarios. Se considera registro lógico único y con componentes entre 1 y 19 elementos.

- Mae\_provincias. Se consideran mas de 5 registros y con componentes entre 1 y 19 elementos.
- Mae\_paises. Se consideran mas de 5 registros y con componentes entre 1 y 19 elementos.

### 2.8.3.2.5. Caso de Uso: Crear Proyecto

	Complejidad			Aporte
	Baja	Media	Alta	
Entradas Externas	1			1*3
Salidas Externas				
Consultas Externas				
Archivos Lógicos Internos	2			2*7
Archivos de Interfaces Externas				
Total				17

Tabla EVS 2.15 - Función Crear Proyecto

- ✓ Entradas Externas: Corresponde a la entrada de datos por parte del usuario para crear un nuevo proyecto en el sistema. Se consideran menos de 4 ítems referenciados.
- ✓ Archivos Lógicos Internos:
  - Usuarios. Se considera registro lógico único y con componentes entre 1 y 19 elementos.
  - Proyectos. Se considera registro lógico único y con componentes entre 1 y 19 elementos.

### 2.8.3.2.6. Caso de Uso: Eliminar Proyecto

	Complejidad			Aporte
	Baja	Media	Alta	
Entradas Externas	1			1*3
Salidas Externas				
Consultas Externas				
Archivos Lógicos Internos	1			1*7
Archivos de Interfaces Externas				
Total				10

Tabla EVS 2.16 - Función Eliminar Proyecto

- ✓ Entradas Externas: Corresponde a la entrada de datos por parte del usuario para eliminar un proyecto en el sistema. Se consideran menos de 4 ítems referenciados.
- ✓ Archivos Lógicos Internos:
  - Usuarios. Se considera registro lógico único y con componentes entre 1 y 19 elementos.
  - Proyectos. Se considera registro lógico único y con componentes entre 1 y 19 elementos.

### 2.8.3.2.7. Caso de Uso: Listar Proyectos

	Complejidad			Aporte
	Baja	Media	Alta	
Entradas Externas				
Salidas Externas	1			1*4
Consultas Externas				
Archivos Lógicos Internos	2			2*7

	Complejidad			Aporte
	Baja	Media	Alta	
Archivos de Interfaces Externas				
Total				18

Tabla EVS 2.17 - Función Listar Proyectos

- ✓ Salidas Externas: Corresponde al reporte de salida de proyectos del usuario activo. Se consideran menos de 4 ítems referenciados.
- ✓ Archivos Lógicos Internos:
  - Usuarios. Se considera registro lógico único y con componentes entre 1 y 19 elementos.
  - Proyectos. Se considera registro lógico único y con componentes entre 1 y 19 elementos.

#### 2.8.3.2.8. Caso de Uso: Crear Versiones de un Proyecto

	Complejidad			Aporte
	Baja	Media	Alta	
Entradas Externas	1			1*3
Salidas Externas				
Consultas Externas				
Archivos Lógicos Internos	2	1		2*7+1*10
Archivos de Interfaces Externas				
Total				27

Tabla EVS 2.18 - Función Crear Versiones de un Proyecto

- ✓ Entradas Externas: Corresponde a la entrada de datos por parte del usuario para crear una nueva versión de un proyecto en el sistema. Se consideran menos de 4 ítems referenciados.

- ✓ Archivos Lógicos Internos:
  - Usuarios. Se considera registro lógico único y con componentes entre 1 y 19 elementos.
  - Proyectos. Se consideran mas de 5 registros y con componentes entre 1 y 19 elementos.
  - Versiones. Se considera registro lógico único y con componentes entre 1 y 19 elementos.

#### 2.8.3.2.9. Caso de Uso: Abrir Versión

	Complejidad			Aporte
	Baja	Media	Alta	
Entradas Externas	1			1*3
Salidas Externas				
Consultas Externas				
Archivos Lógicos Internos	1	2		1*7+2*10
Archivos de Interfaces Externas				
Total				30

Tabla EVS 2.19 - Función Abrir Versión

- ✓ Entradas Externas: Corresponde a la entrada de datos por parte del usuario para abrir una versión de un proyecto en el sistema. Se consideran menos de 4 ítems referenciados.
- ✓ Archivos Lógicos Internos:
  - Usuarios. Se considera registro lógico único y con componentes entre 1 y 19 elementos.
  - Proyectos. Se consideran mas de 5 registros y con componentes entre 1 y 19 elementos.

- Versiones. Se consideran mas de 5 registros y con componentes entre 1 y 19 elementos.

### 2.8.3.2.10. Caso de Uso: Eliminar Versión

	Complejidad			Aporte
	Baja	Media	Alta	
Entradas Externas	1			1*3
Salidas Externas				
Consultas Externas				
Archivos Lógicos Internos	1	2		1*7+2*10
Archivos de Interfaces Externas				
Total				30

Tabla EVS 2.20 - Función Eliminar Versión

- ✓ Entradas Externas: Corresponde a la entrada de datos por parte del usuario para eliminar una versión de un proyecto en el sistema. Se consideran menos de 4 ítems referenciados.
- ✓ Archivos Lógicos Internos:
  - Usuarios. Se considera registro lógico único y con componentes entre 1 y 19 elementos.
  - Proyectos. Se consideran mas de 5 registros y con componentes entre 1 y 19 elementos.
  - Versiones. Se consideran mas de 5 registros y con componentes entre 1 y 19 elementos.

### 2.8.3.2.11. Caso de Uso: Listar Versiones

	Complejidad			Aporte
	Baja	Media	Alta	
Entradas Externas				
Salidas Externas	1			4
Consultas Externas				
Archivos Lógicos Internos	1	2		$1*7+2*10$
Archivos de Interfaces Externas				
Total				31

Tabla EVS 2.21 - Función Listar Versiones

- ✓ Salidas Externas: Corresponde al reporte de salida de versiones de proyectos del usuario activo. Se consideran menos de 4 ítems referenciados.
- ✓ Archivos Lógicos Internos:
  - Usuarios. Se considera registro lógico único y con componentes entre 1 y 19 elementos.
  - Proyectos. Se consideran mas de 5 registros y con componentes entre 1 y 19 elementos.
  - Versiones. Se consideran mas de 5 registros y con componentes entre 1 y 19 elementos.

### 2.8.3.2.12. Caso de Uso: Establecer Parámetros de SOM

	Complejidad			Aporte
	Baja	Media	Alta	
Entradas Externas		1		$1*4$
Salidas Externas				

	Complejidad			Aporte
	Baja	Media	Alta	
Consultas Externas				
Archivos Lógicos Internos	3			3*7
Archivos de Interfaces Externas				
Total				25

Tabla EVS 2.22 - Función Establecer Parámetros de SOM

- ✓ Entradas Externas: Corresponde a la entrada de datos por parte del usuario para cargar los parámetros de 1 algoritmo SOM en el sistema. Se consideran entre 5 y 15 ítems referenciados.
- ✓ Archivos Lógicos Internos:
  - Usuarios. Se considera registro lógico único y con componentes entre 1 y 19 elementos.
  - Proyectos. Se considera registro lógico único y con componentes entre 1 y 19 elementos.
  - Versiones. Se considera registro lógico único y con componentes entre 1 y 19 elementos.

#### 2.8.3.2.13. Caso de Uso: Establecer Parámetros de C5.0

	Complejidad			Aporte
	Baja	Media	Alta	
Entradas Externas		1		1*4
Salidas Externas				
Consultas Externas				
Archivos Lógicos Internos	3			3*7
Archivos de Interfaces Externas				
Total				25

Tabla EVS 2.23 - Función Establecer Parámetros de C5.0

- ✓ Entradas Externas: Corresponde a la entrada de datos por parte del usuario para cargar los parámetros de 1 algoritmo C5.0 en el sistema. Se consideran entre 5 y 15 ítems referenciados.
- ✓ Archivos Lógicos Internos:
  - Usuarios. Se considera registro lógico único y con componentes entre 1 y 19 elementos.
  - Proyectos. Se considera registro lógico único y con componentes entre 1 y 19 elementos.
  - Versiones. Se considera registro lógico único y con componentes entre 1 y 19 elementos.

#### 2.8.3.2.14. Caso de Uso: Ejecutar Proceso de Explotación de Datos

	Complejidad			Aporte
	Baja	Media	Alta	
Entradas Externas		1		1*4
Salidas Externas		2		2*5
Consultas Externas				
Archivos Lógicos Internos	3			3*7
Archivos de Interfaces Externas				
Total				35

Tabla EVS 2.24 - Función Ejecutar Proceso de Explotación de Datos

- ✓ Entradas Externas: Corresponde a la entrada de datos por parte del usuario para cargar los parámetros de 1 algoritmo SOM en el sistema. Se consideran entre 5 y 15 ítems referenciados.
- ✓ Salidas Externas: Corresponde a las salidas luego de ejecutar el/los algoritmo/s SOM y C5.0. Se consideran más de archivos referenciados.

✓ Archivos Lógicos Internos:

- Usuarios. Se considera registro lógico único y con componentes entre 1 y 19 elementos.
- Proyectos. Se considera registro lógico único y con componentes entre 1 y 19 elementos.
- Versiones. Se considera registro lógico único y con componentes entre 1 y 19 elementos.

**2.8.3.2.15. Caso de Uso: Total General de Puntos de Función**

<b>Funciones</b>	<b>Totales</b>
01 - Ingreso al Sistema	10
02 - Salir del Sistema	3
03 - Agregar Usuario	31
04 - Modificar Usuario	31
05 - Crear Proyecto	17
06 - Eliminar Proyecto	10
07 - Listar Proyectos	18
08 - Crear Versiones de un Proyecto	27
09 - Abrir Versión	30
10 - Eliminar Versión	30
11 - Listar Versiones	31
12 - Establecer Parámetros de SOM	25
13 - Establecer Parámetros de C5.0	25
14 - Ejecutar Proceso de Explotación de Datos	35
<b>UFP (Puntos de Función sin ajustar) Total</b>	<b>323</b>

Tabla EVS 2.25 - Total General de Puntos de Función.

### 2.8.3.3. Esfuerzo y Duración

Una vez obtenido el total de Puntos de Función, en este caso 323, se utiliza COCOMO II para estimar el esfuerzo.

Este método se basa en la aplicación de ecuaciones matemáticas sea sobre los Puntos de Función sin ajustar o sobre la cantidad de líneas de código fuente estimadas. Las mencionadas ecuaciones se encuentran ponderadas por los denominados factores de costo. La ecuación base que se toma es la siguiente

$$PM_{\text{nominal}} = A * (\text{Size})^B$$

Donde PM nominal es el esfuerzo nominal requerido en meses-hombre.

Size es el tamaño que se estima tiene el software a desarrollar medido en miles de líneas de código fuente (SLOC- sigla en inglés comúnmente utilizada para este concepto) o Puntos de Función convertidos a éstas mediante un factor que depende de la tecnología utilizada.

Se considera el lenguaje Microsoft Visual FoxPro (SPR Programming Language Table – 2002), por lo que el factor de conversión es 29 por cada Punto de Función.

Haciendo el cálculo de miles de líneas de código fuentes es:

$$\text{Size} = 29 * 323 = 9.367$$

A es una constante que denota los efectos multiplicativos en el esfuerzo requerido de acuerdo al crecimiento del tamaño del software. Inicialmente se utiliza el valor 2.94.

B es una constante que afecta exponencialmente al esfuerzo y es determinada mediante la ponderación de aspectos positivos sobre los negativos que afectan al proyecto en cuanto a su complejidad y entorno de desarrollo. Las variables escalares que permiten determinar el factor B son las siguientes:

<b>VARIABLE</b>	<b>Descripción</b>
PREC	Experiencia en este tipo de desarrollos.
FLEX	Flexibilidad para el desarrollo.
RSEL	Relacionado con la arquitectura y la mitigación de riesgos.
TEAM	Cohesión y madurez del equipo de desarrollo.
PMAT	Proceso de madurez de desarrollo del software.

Tabla EVS 2.26 - Factores Escalares

Los criterios y valores posibles para cada una de estas variables se encuentran determinados en la siguiente tabla 2.24,

<b>Factor <math>W_i</math></b>	<b>Muy Bajo</b>	<b>Bajo</b>	<b>Nominal</b>	<b>Alto</b>	<b>Muy Alto</b>	<b>Extra Alto</b>
<b>PREC</b>	Completa	Completa	Algo	Familiar	Muy Familiar	Absolutamente Familiar
	6.20	4.96	3.72	2.48	1.24	0.00
<b>FLEX</b>	Rigurosa	Ocasional	Algo	Generalmente Conforme	Algo de Conformidad	Objetivos Generales
	5.07	4.05	3.04	2.03	1.01	0.00
<b>RESL</b>	Poco (20%)	Algo (40%)	A menudo (60%)	Generalmente (75%)	Mayormente (90%)	Totalmente (100%)
	7.07	5.65	4.24	2.83	1.41	0.00
<b>TEAM</b>	Interac. Muy difícil	Algo dificultosa	Algo Cooperativa	Cooperativa	Altamente Cooperativa	Interacción Total
	5.48	4.38	3.29	2.19	1.10	0.00
<b>PMAT</b>	Promedio de respuestas afirmativas en cuestionario CMM					
	7.60	6.24	4.68	3.12	1.56	0.00

Tabla EVS 2.27 - Valores para Factores Escalares

Para el proyecto se han determinado los siguientes valores,

<b>VARIABLE</b>	<b>Valor para este proyecto</b>	
PREC	Nominal	3.72
FLEX	Alto	1.01
RSEL	Alto	2.83
TEAM	Muy Alto	1.10
PMAT	Bajo	6.24
<b>TOTAL</b>	<b>14,9</b>	

Tabla EVS 2.28 - Factores Escalares para el proyecto

La justificación de cada uno de los valores elegidos es la siguiente,

- ✓ PREC: Nominal, dado que se tiene una idea básica sobre el tipo de sistema a desarrollar y experiencia en entornos de desarrollo similares.
- ✓ FLEX: Muy Alto, ya que se requiere una adaptabilidad considerable del software conforme a los requisitos exigidos.
- ✓ RSEL: Alto, se tiene una idea clara sobre la arquitectura que debe soportar este sistema.
- ✓ TEAM: Muy Alto, de acuerdo a la predisposición para el desarrollo del producto.
- ✓ PMAT: Bajo, por ser la primera vez que se intenta un desarrollo organizado bajo los lineamientos de un proceso determinado.

Entonces el factor escalar B se calcula con la siguiente ecuación:

$$B = 0,91 + 0,01 * \Sigma(W_i) = 1,059$$

El esfuerzo nominal resulta

$$PM_{\text{nominal}} = 2,94 * (9,367)^{1,059} = 31,42 \text{ Meses/Hombre}$$

Para completar la estimación se debe ajustar el esfuerzo nominal de acuerdo a las características del proyecto y los multiplicadores de esfuerzo que le corresponden. Se considera el modelo Post-Arquitectura ya que no hay exploración de arquitecturas alternativas y el concepto de operación ya está definido.

La fórmula a utilizar es:

$$PM_{\text{ajustado}} = PM_{\text{nominal}} * \Pi (\text{Multiplicadores de Esfuerzo})$$

Los valores de los multiplicadores para cada factor por nivel son los siguientes:

<b>Factores Producto</b>	<b>Muy Bajo</b>	<b>Bajo</b>	<b>Nominal</b>	<b>Alto</b>	<b>Muy Alto</b>	<b>Extra Alto</b>
RELY	0.82	0.92	1.00	1.10	1.26	xxxx
DATA	xxxx	0.90	1.00	1.14	1.28	xxxx
DOCU	0.81	0.91	1.00	1.11	1.23	xxxx
CPLX	0.73	0.87	1.00	1.17	1.34	1.74
RUSE	xxxx	0.95	1.00	1.07	1.15	1.24

Tabla EVS 2.29 - Valores para Factores Producto

<b>Factores Platafor.</b>	<b>Muy Bajo</b>	<b>Bajo</b>	<b>Nominal</b>	<b>Alto</b>	<b>Muy Alto</b>	<b>Extra Alto</b>
TIME	xxxx	xxxx	1.00	1.11	1.29	1.63
STOR	xxxx	xxxx	1.00	1.05	1.17	1.46
PVOL	xxxx	0.91	1.00	1.15	1.30	xxxx

Tabla EVS 2.30 - Valores para Factores Plataforma

<b>Factores Personal</b>	<b>Muy Bajo</b>	<b>Bajo</b>	<b>Nominal</b>	<b>Alto</b>	<b>Muy Alto</b>	<b>Extra Alto</b>
ACAP	1.42	1.19	1.00	0.85	0.71	xxxx
PCAP	1.34	1.15	1.00	0.88	0.76	xxxx
AEXP	1.22	1.10	1.00	0.88	0.81	xxxx

<b>Factores Personal</b>	<b>Muy Bajo</b>	<b>Bajo</b>	<b>Nominal</b>	<b>Alto</b>	<b>Muy Alto</b>	<b>Extra Alto</b>
PEXP	1.19	1.09	1.00	0.91	0.85	xxxx
LTEX	1.20	1.09	1.00	0.91	0.84	xxxx
PCON	1.29	1.12	1.00	0.90	0.81	xxxx

Tabla EVS 2.31 - Valores para Factores del Personal

<b>Factores Proyecto</b>	<b>Muy Bajo</b>	<b>Bajo</b>	<b>Nominal</b>	<b>Alto</b>	<b>Muy Alto</b>	<b>Extra Alto</b>
TOOL	1.17	1.09	1.00	0.90	0.78	xxxx
SITE	1.22	1.09	1.00	0.93	0.96	0.80
SCED	1.43	1.14	1.00	1.00	1.00	xxxx

Tabla EVS 2.32 - Valores para Factores del Proyecto

Teniendo en cuenta lo anterior para este proyecto se ha seleccionado lo que muestra la tabla 2.30.

<b>Multiplicador</b>	<b>Descripción</b>	<b>Ponderación</b>	<b>Valor</b>
<b>QUE AFECTAN AL PRODUCTO</b>			
RELY	Confiabilidad requerida del software	Alto	1.10
DATA	Tamaño de la base de datos.	Muy alto	1.28
CPLX	Complejidad del Producto	Nominal	1.0
RUSE	Reusabilidad del Código	Nominal	1.0
DOCU	Requerimiento de documentación	Nominal	1.0
<b>QUE AFECTAN A LA PLATAFORMA</b>			
TIME	Restricciones de tiempo de ejecución	Nominal	1.00
STOR	Restricciones de almacenamiento	Bajo	1.00
PVOL	Volatilidad de la plataforma	Bajo	0.87
<b>QUE AFECTAN AL PERSONAL</b>			
ACAP	Capacidad de los analistas	Muy Alto	0.71
PCAP	Capacidad de los programadores	Muy Alto	0.76

<b>Multiplicador</b>	<b>Descripción</b>	<b>Ponderación</b>	<b>Valor</b>
AEXP	Experiencia en aplicaciones	Nominal	1.00
PEXP	Experiencia en plataforma	Muy Alto	0.85
LTEX	Experiencia el lenguaje y herramienta de desarrollo	Muy Alto	0.84
PCON	Continuidad del personal	Muy Alto	0.81
<b>QUE AFECTAN AL PROYECTO</b>			
TOOL	Uso de la herramienta de software	Muy Alto	0.78
SITE	Desarrollo en múltiples ubicaciones	Nominal	1.0
SCED	Restricciones de calendario	Nominal	1.0
<b>II (Multiplicadores de Esfuerzo)</b>			<b>0.298</b>

Tabla EVS 2.33 - Totales para Multiplicadores del Proyecto

La justificación de cada uno de los valores elegidos es la siguiente,

- ✓ RELY: Alto, por el tipo de tareas que se desea que el sistema realice.
- ✓ DATA: Muy Alto, debido a que se estima muy probable la utilización de una gran cantidad de datos de entrada y de salida para este sistema.
- ✓ CPLX: Nominal, debido a que se implementan algoritmos ya conocidos al sistema.
- ✓ RUSE: Nominal, debido a que el mismo sistema utiliza esta facilidad de un entorno previamente desarrollado.
- ✓ DOCU: Nominal, por ser la primera vez que se intenta un desarrollo de este tipo se pretende cubrir la documentación básica necesaria.
- ✓ TIME: Nominal, debido a que es factible que el sistema utilice un gran volumen de datos y por ende los tiempos de ejecución pueden ser extensos.
- ✓ STOR: Bajo, no hay restricciones en cuanto al almacenamiento.
- ✓ PVOL: Bajo, no se estima cambio de plataforma en el mediano plazo.

- ✓ ACAP: Muy Alto, se considera la capacidad del analista muy superior a la del promedio en el mercado laboral.
- ✓ PCAP: Muy Alto, se considera la capacidad del programador muy superior a la del promedio en el mercado laboral.
- ✓ AEXP: Nominal, por haber desarrollado algunas aplicaciones de este tipo.
- ✓ PEXP: Muy Alto, por haber desarrollado gran cantidad de aplicaciones en esta plataforma.
- ✓ LTEX: Muy Alto, por haber desarrollado gran cantidad de aplicaciones en este lenguaje.
- ✓ PCON: Muy Alto, por considerar este factor como pilar fundamental para el desarrollo continuado y exitoso del sistema.
- ✓ TOOL: Muy Alto, se cuenta con amplia experiencia en la herramienta de desarrollo.
- ✓ SITE: Nominal, no aplica en este caso.
- ✓ SCED: Nominal, ya que no hay restricciones temporales extremadamente estrictas.

$PM \text{ ajustado} = 31.42 * 0.298 = 9,37 \text{ meses/hombre}$

Tomando en cuenta la cantidad de personal se calcula el tiempo de desarrollo:

$Duración = PM \text{ ajustado} / \text{Cantidad de personal} = 9,37 \text{ meses} / 1 = 9,37 \text{ meses.}$

## 2.9. Planificación

El diagrama de Gantt que muestra la discriminación de las actividades previstas para este proyecto, ver diagrama EVS 2.1:

ID	Nombre de tarea	Comienzo	Fin	Duración	2005					2006													
					Jul	Ago	Sep	Oct	Nov	Dic	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic	
1	<b>Plan de sistemas de información</b>	<b>01/07/2005</b>	<b>29/07/2005</b>	<b>21d</b>																			
2	Aproximación inicial al problema	01/07/2005	08/07/2005	6d																			
3	Identificación de requisitos iniciales	08/07/2005	14/07/2005	5d																			
4	Solución propuesta	15/07/2005	29/07/2005	11d																			
5	<b>Estudio de Viabilidad del Sistema</b>	<b>01/08/2005</b>	<b>31/08/2005</b>	<b>23d</b>																			
6	Análisis de la Situación y Restricciones	01/08/2005	05/08/2005	5d																			
7	Planificación	05/08/2005	12/08/2005	6d																			
8	Gestión de Configuración	12/08/2005	19/08/2005	6d																			
9	Gestión de Aseguramiento de Calidad	19/08/2005	26/08/2005	6d																			
10	Gestión de Riesgos	26/08/2005	31/08/2005	4d																			
11	<b>Modelo de análisis</b>	<b>01/09/2005</b>	<b>29/12/2005</b>	<b>86d</b>																			
12	Definición del Sistema	01/09/2005	09/09/2005	7d																			
13	Especificación de Requisitos	09/09/2005	14/10/2005	26d																			
14	Identificación de Subsistemas de Análisis	14/10/2005	28/10/2005	11d																			
15	Especificación de Interfaz de Usuario	28/10/2005	01/12/2005	25d																			
16	Análisis de Consistencia y Especificación de Requisitos	01/12/2005	29/12/2005	21d																			
17	<b>Diseño del sistema de información</b>	<b>02/01/2006</b>	<b>28/04/2006</b>	<b>85d</b>																			
18	Definición de la Arquitectura del Sistema	02/01/2006	13/01/2006	10d																			
19	Diseño de la Arquitectura de Soporte	13/01/2006	27/01/2006	11d																			
20	Diseño de Casos de Uso Reales	27/01/2006	10/02/2006	11d																			
21	Diseño de Clases	10/02/2006	24/02/2006	11d																			
22	Diseño Físico de Datos	24/02/2006	03/03/2006	6d																			
23	Verificación y Aceptación de la Arquitectura del Sistema	03/03/2006	17/03/2006	11d																			
24	Generación de Especificaciones de Construcción	17/03/2006	10/04/2006	17d																			
25	Especificación Técnica del Plan de Pruebas	10/04/2006	28/04/2006	15d																			
26	<b>Construcción del sistema de información</b>	<b>01/05/2006</b>	<b>31/08/2006</b>	<b>89d</b>																			
27	Preparación del Entorno de Generación y Construcción	01/05/2006	22/05/2006	16d																			
28	Generación del Código de los Componentes y Procedimientos	22/05/2006	01/08/2006	52d																			
29	Ejecución de las Pruebas Unitarias	01/08/2006	15/08/2006	11d																			
30	Ejecución de las Pruebas del Sistema	15/08/2006	31/08/2006	13d																			
31	<b>Implantación y aceptación del sistema</b>	<b>01/09/2006</b>	<b>13/10/2006</b>	<b>31d</b>																			
32	Incorporación del Sistema al Entorno de Operación	01/09/2006	07/09/2006	5d																			
33	Carga de Datos al Entorno de Operación	07/09/2006	20/09/2006	10d																			
34	Pruebas de Implantación del Sistema	20/09/2006	04/10/2006	11d																			
35	Pruebas de Aceptación del Sistema	04/10/2006	13/10/2006	8d																			
36	<b>Manuales de usuario</b>	<b>16/10/2006</b>	<b>14/11/2006</b>	<b>22d</b>																			

Diagrama EVS 2.1 – Gantt del proyecto

## 2.10. Gestión de Configuración

A continuación se detalla lo concerniente a la administración de configuración de software (ACS) que engloba todo lo necesario para gestionar la integridad de los productos software implementados durante el desarrollo del proyecto.

Las actividades relacionadas con el ACS para este proyecto son:

- ✓ Identificación de la Configuración.
- ✓ Control de la Configuración.

### 2.10.1. Identificación de la Configuración

Los elementos a considerar como integrantes de los elementos de configuración del software (ECS) y su tipo para este proyecto son los siguientes:

<b>PRODUCTOS SOFTWARE</b>	<b>TIPO</b>
Plan de sistemas de información	Documento
Estudio de Viabilidad del Sistema	Documento
Modelo de análisis	Documento
Diseño del sistema de información	Documento
Construcción del sistema de información	Documento
Implantación y aceptación del sistema	Documento
Manuales de usuario	Documento
Software	Software

Tabla EVS 2.34 - Tipos de Productos de Software

Los ECS se denominan anteponiéndole la sigla que identifica al proyecto, en este caso es “DM”.

El formato general para cada producto software es, DM\_CÓDIGO, siendo CÓDIGO una combinación alfanumérica que identifica unívocamente cada elemento en el desarrollo del proyecto.

En el código se indica a que línea base corresponde el elemento, dentro de esta a que fase pertenece por cada iteración. A continuación se incluirá la identificación propia del ECS y su número de versión.

La siguiente tabla EVS 2.35 indica los posibles valores que puede tomar el componente CÓDIGO:

<b>Código</b>	<b>Tipo de Producto</b>
CAP.XX_nombre_X.X	Texto de cada Capitulo de la Memoria del Proyecto.
ANX.XX_nombre_X.X	Texto de cada Anexo de la Memoria del Proyecto.

Tabla EVS 2.35 - Formato de Rótulos

La versión de cada documento consiste en la indicación al final del campo código de la combinación de dos dígitos separados por un punto. El primer dígito parte del valor uno (1), el segundo de cero (0).

El nombre indica la naturaleza del ECS que se esta tratando. En cada capítulo de la memoria se establece la información de configuración, conjuntamente con la fecha a que corresponde y el estado en que se encuentra.

El formato adoptado para la fecha es dd/mm/aaaa.

Se considera que un ECS se encuentra solamente en uno de estos tres estados posibles, a saber:

- ✓ En edición: indica en desarrollo o que fue elevado para su corrección.
- ✓ Finalizado: indica que ya se encuentra corregido y en etapa de aprobación por el tutor de tesis.

- ✓ Aprobado: indica aprobado por el tutor de tesis

### **2.10.2. Control de la Configuración**

El control de las versiones comienza con la versión 1.0 en cada ECS, por cada cambio menor significativo se aumentará el valor del dígito decimal en 1. Si fuese un cambio mayor se aumentará el primer dígito en una unidad y el valor del decimal vuelve a 0.

El detalle se encuentra en el anexo denominado “Control de Configuración” de ésta tesis. En mencionado anexo se listan únicamente las versiones de cada ECS.

## **2.11. Gestión de Aseguramiento de Calidad**

### **2.11.1. Introducción**

Las políticas de calidad están basadas en asegurar que, el producto final satisfaga los requisitos y que durante el desarrollo se adapte a los cambios que puedan producirse en los requerimientos.

Para llevar a cabo el control de la calidad se ejecutaran revisiones y pruebas de software.

La aplicación de conceptos para aseguramiento de calidad en el presente proyecto es desarrollado íntegramente por el maestrando, con la supervisión de la Directora de Tesis.

### **2.11.2. Identificación de las Propiedades de Calidad**

Las propiedades de calidad para el sistema “DM” pueden resumirse en los siguientes puntos:

- ✓ Corrección: El sistema debe satisfacer las especificaciones de requisitos establecidas. Se verificará el número de requisitos entregados con el producto contra los definidos.

- ✓ Facilidad de Uso: Operatoria simple e intuitiva por parte del usuario.
- ✓ Seguridad: Acceso a la funcionalidad del sistema únicamente a través de la combinación de nombre de usuario y clave.
- ✓ Performance: Tiempo de respuesta conforme a las necesidades del cliente o usuario.
- ✓ Facilidad de mantenimiento: Se establecerá bajo el análisis de la documentación, buscando la correspondencia entre lo escrito y el código ejecutable.

### **2.11.3. Revisiones**

Para todos los productos se debe realizar una revisión puntual antes de realizar la incorporación a la línea base (ya sea como consecuencia de cambios realizados en un documento existente en línea base, o durante su primera inserción en la misma) a efectos de que se cumplan las condiciones de calidad establecidas.

En el Anexo “Aseguramiento de Calidad” de esta tesis, se encuentran los resultados de las revisiones que se detallan en los apartados 2.11.3.1 al 2.11.3.3.

#### **2.11.3.1. Revisiones definidas para la etapa de Análisis**

##### *Catálogo de Requisitos*

Mediante una lista de preguntas de verificación se comprueba la precisión y completitud de los requisitos.

- ✓ ¿Los requisitos son entendibles?
- ✓ ¿Son factibles de implementar?
- ✓ ¿Existen requisitos superpuestos o que se complementan?
- ✓ ¿Son importantes para solucionar los problemas detectados?

- ✓ ¿Falta especificar algún requisito?

#### *Consistencia entre productos del Análisis*

En esta actividad se controla que se haya realizado la validación y verificación de los productos obtenidos durante la fase de análisis. La lista de verificación es:

- ✓ ¿Todos los requisitos tienen asociado un caso de uso?
- ✓ ¿Los casos de uso tienen su descripción completa?
- ✓ ¿Los casos de uso tienen su correspondiente clase de análisis?
- ✓ ¿Los casos de uso tienen definido su diagrama de colaboración correspondiente?

#### **2.11.3.2. Revisiones definidas para la etapa de Diseño**

Bajo esta actividad se controla que la arquitectura del sistema responda a los requisitos establecidos. La verificación se realiza mediante la lista de preguntas que siguen a continuación:

- ✓ ¿Los casos de uso tienen su correspondiente diagrama de clases de análisis y de diseño?
- ✓ ¿Las clases de diseño tienen su correspondiente descripción?
- ✓ ¿Para cada requisito funcional existe su correspondiente caso de uso, diagrama de clase de diseño y su descripción?

#### **2.11.3.3. Revisiones definidas para la etapa de las Pruebas**

Para esta actividad se definen las siguientes cuestiones:

- ✓ ¿Son probados todos los requisitos?
- ✓ ¿Se verifican los límites de rango?

- ✓ ¿Se realiza la prueba de interfaz entre cada modulo?
- ✓ ¿Han sido verificados los valores representativos de las clases?
- ✓ ¿Los casos de prueba verifican todos los procesos?

#### **2.11.4. Pruebas de Software**

Las pruebas que se necesitan desarrollar comprenden los siguientes tipos:

- ✓ Pruebas Unitarias: son las realizadas sobre cada modulo en forma independiente.
- ✓ Pruebas de Integración: son las realizadas en forma conjunta con los módulos que han pasado las Pruebas Unitarias.
- ✓ Pruebas del Sistema: Se prueba el sistema como un todo, verificando que se cumplan los requisitos de calidad estipulados.

#### **2.12. Gestión de Riesgos**

En todo proyecto desarrollo de software es necesario realizar una gestión de los riesgos que puede llegar a enfrentar el mismo. Esto incluye actividades tales como la identificación, priorización y cuantificación de riesgos, la realización de un plan de acción para minimizarlos y planes de contingencia para mitigar el impacto en caso de que ocurran. Los posibles riesgos pueden incluir aspectos de negocios, tecnológicos, recursos humanos, sociales y económicos según el caso.

##### **2.12.1. Metodología para la Gestión de los Riesgos**

- ✓ Identificación de los riesgos: Se realiza mediante una revisión de la documentación que dio origen al proyecto y a través de reuniones con los principales participantes pertenecientes a las diversas áreas del negocio y de tecnología. Es necesario establecer prioridades a cada uno de estos riesgos.

- ✓ Cuantificación de los riesgos: Después de realizar la identificación de riesgos con las causas asociadas, se debe determinar la probabilidad de ocurrencia y cuantificar el impacto del riesgo sobre el proyecto en caso de que este se presentara. Luego, conforme a los criterios especificados para cada riesgo detectado, se procede a su evaluación para considerarlo dentro del plan de mitigación.
- ✓ Desarrollo de respuesta a riesgos: Durante el proceso de la administración de riesgos es necesario desarrollar planes de respuesta a los riesgos. Uno de estos planes, el de acción, se concentra en la prevención de la ocurrencia de los riesgos. Algunas actividades permiten minimizar la posibilidad de aparición del riesgo. De igual forma, es necesario realizar un plan de contingencias para minimizar el impacto de los riesgos cuando estos se presenten.
- ✓ Control de respuesta a riesgos: Es necesario establecer controles regulares para revisar el status de los riesgos y en caso de algún cambio tomar acciones para minimizar su ocurrencia, o realizar las actividades establecidas en el plan de contingencias en el menor tiempo posible y evitar así mayores consecuencias.

### 2.12.2. Desarrollo de la Metodología para la Gestión de los Riesgos

A continuación, se presenta una tabla que denota el resultado de la actividad de identificación de riesgos para el presente proyecto, la prioridad se encuentra determinada por su orden de presentación:

Nro.	Riesgo
R01	La aplicación no responde a los requisitos definidos por el usuario.
R02	Existen dificultades técnicas para implementar ciertas funcionalidades.
R03	Falla de seguridad en la aplicación.
R04	Limitaciones de hardware que afectan al rendimiento del sistema.

<b>Nro.</b>	<b>Riesgo</b>
R05	Errores por reutilización de paquetes de software existentes.
R06	Retraso en la planificación del proyecto.

Tabla EVS 2.36 - Riesgos Considerados

Para determinar la probabilidad de ocurrencia de cada uno de los riesgos identificados, se emplean los calificadores usados para este tema por la Fuerza Área de los EE.UU [Collefo J., 2003], estos son: Muy Baja, Baja, Media, Alta y Muy Alta.

<b>Riesgo</b>	<b>Probabilidad</b>
R01	Muy Baja
R02	Baja
R03	Alta
R04	Baja
R05	Baja
R06	Baja

Tabla EVS 2.37 - Probabilidad por Riesgo

Para determinar el impacto que tendría la ocurrencia de cada uno de los riesgos identificados, se emplean los calificadores usados por la misma fuente citada para el caso de la probabilidad, estos son: Insignificante, Marginal, Crítico y Catastrófico.

<b>Riesgo</b>	<b>Impacto</b>
R01	Crítico
R02	Marginal
R03	Crítico
R04	Marginal
R05	Marginal
R06	Marginal

Tabla EVS 2.38 - Calificación por Riesgo

Los criterios de estimación para cada uno de los riesgos detectados son los siguientes:

- R01 La aplicación no responde a los requisitos definidos por el usuario.

Este riesgo se califica de muy baja probabilidad ya que, la visión del proyecto fue propuesta por el autor a una autoridad de la agencia, además se trabajará en contacto diario con cada actor interesado en el sistema y el desarrollo se hará en forma iterativa por lo que se dispone de prototipos para ir evaluando a medida que avance la implementación. Lógicamente, en caso de producirse, este riesgo se considera crítico.

- R02 Existen dificultades técnicas para implementar ciertas funcionalidades.

Para el caso del sistema “DM”, este riesgo significa la dificultad de implementar algunas de las funcionalidades exigidas a través del entorno de desarrollo elegido. Dado el énfasis expresado por la empresa que suministra el software de desarrollo respecto de su creación basada en el concepto de los paradigmas de la Internet, se considera de baja probabilidad la ocurrencia del mismo. En caso de presentarse, serían de cierta consideración pero de ningún modo críticas.

- R03 Falla de seguridad en la aplicación.

Por tratarse de un sistema al cual se puede tener acceso fuera de los límites controlados de la empresa y dada la posibilidad de cometer errores en el diseño o implementación del mismo que impliquen procesos de autorización de uso de recurso y validación de acceso, se considera este riesgo como de alta probabilidad.. En caso de verificarse su impacto se considera crítico.

- R04 Limitaciones de hardware afectan al rendimiento del sistema.

Este riesgo puede suceder si el hardware implicado para dar soporte al sistema no conlleva una performance adecuada para satisfacer los requisitos de rendimiento de la aplicación. Considerando las capacidades actuales con otras aplicaciones de acceso vía Internet y el margen operativo que cuentan los

canales de comunicación propios, se cree de baja probabilidad la ocurrencia de este riesgo. En caso de verificarse su impacto se considera marginal.

- R05 Errores por reutilización de paquetes de software existentes.

Este riesgo puede suceder cuando se utilizan paquetes de software ya desarrollados, sean propios o de terceros, y que ya han sido usados o referenciados por el mercado. Dado que para este proyecto se piensa utilizar paquetes desarrollados y probados por la misma empresa que soporta el entorno de desarrollo se considera de baja probabilidad la ocurrencia de este riesgo. En caso de verificarse su impacto se considera marginal.

- R06 Retraso en la implementación del proyecto.

Dado que la dedicación del personal de desarrollo asignado al proyecto “DM” es prioritaria, se califica la probabilidad de este riesgo como baja. Ya que este proyecto reemplazaría a un sistema existente y en funcionamiento no se altera la operatividad de la agencia por lo que el impacto en caso de producirse como marginal.

Bajo estos criterios, a continuación, a cada riesgo se le asigna un valor global surgido de la combinación probabilidad versus impacto conforme a la siguiente tabla EVS 2.39. Se descartará el riesgo que quede en categorías Bajo o Ninguno.

<b>Impacto / Probabilidad</b>	<b>Muy Alta</b>	<b>Alta</b>	<b>Media</b>	<b>Baja</b>	<b>Muy Baja</b>
Catastrófico	Muy Alto	Alto	Moderado	Moderado	Bajo
Critico	Alto	Alto	Moderado	Bajo	Ninguno
Marginal	Moderado	Moderado	Bajo	Ninguno	Ninguno
Insignificante	Moderado	Bajo	Bajo	Ninguno	Ninguno

Tabla EVS 2.39 - Riesgo Global

Los resultados obtenidos para los riesgos analizados son mostrados en la tabla EVS 2.40.

<b>Riesgo</b>	<b>Probabilidad</b>	<b>Impacto</b>	<b>Riesgo Global</b>	<b>Acción</b>
R01	Muy Baja	Crítico	Ninguno	Descartar
R02	Baja	Marginal	Ninguno	Descartar
R03	Alta	Crítico	Alto	Planificar
R04	Baja	Marginal	Ninguno	Descartar
R05	Baja	Marginal	Ninguno	Descartar
R06	Baja	Marginal	Ninguno	Descartar

Tabla EVS 2.40 - Resultados Evaluación Global del Riesgo

El riesgo a planificar es,

- ✓ R03 Falla de seguridad en la aplicación.

Para éste se desarrollará un Plan de Acción, que intenta minimizar el efecto del riesgo mediante acciones preventivas, y un Plan de Contingencia, que se activa una vez ocurrido el evento que provoca el riesgo y que provoca una atención rápida para mitigarlo. El mencionado plan se encuentra detallado en el anexo “Plan de Acción y Contingencia Frente al Riesgo” de esta tesis.

---

## CAPÍTULO 3 – MODELO DE ANÁLISIS

### 3.1. Definición del Sistema

#### 3.1.1. Determinación del Alcance del Sistema

##### 3.1.1.1. Modelo de Negocio

El modelo de negocio contempla los procesos principales del negocio bajo análisis y la forma en que los mismos se llevan a cabo. Dentro de este modelo, los procesos se representan mediante casos de uso de negocio. El detalle sobre las actividades llevadas a cabo y las entidades utilizadas para completar cada proceso, se documentan mediante diagramas de actividades.

El negocio cubierto por el sistema es la realización de minería de datos por parte de un analista del negocio, tal como se muestra en la figura ASI 3.1.



Figura ASI 3.1: Modelo de negocio

Actores:

**Analista del Negocio:** Este actor representa a los distintos usuarios del sistema, que tratan a partir de ciertos algoritmos, de descubrir patrones en los datos de entrada.

Casos de Uso:

**Realizar Minería de Datos:** se realiza esto mediante algoritmos específicos para esta actividad, en este caso utilizaremos dos, SOM y C5.0. Se obtiene datos de salida ya sean clusters o árboles de decisión, para que las persona a cargo del proceso traten de encontrar patrones de comportamiento.

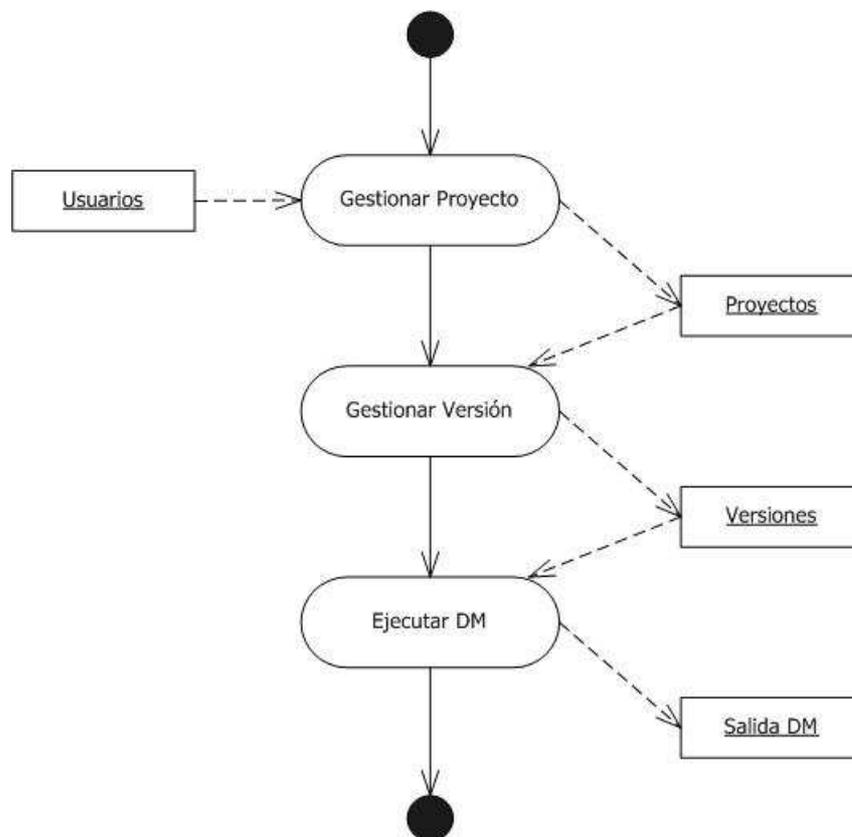


Figura ASI 3.2: Detalle de actividades llevadas a cabo durante un proyecto de Minería de Datos

A continuación se describen las actividades detalladas en el diagrama:

Crear Proyecto:

Esta actividad consiste en gestionar proyectos para realizar de Minería de Datos.

Crear Versión:

Esta actividad consiste en gestionar versiones de un proyecto de Minería de Datos.

Ejecutar DM:

Esta actividad consiste en realizar específicamente el proceso de Minería de Datos sobre un lote especificado.

### **3.1.2. Identificación de los Usuarios Participantes y Finales**

En esta tarea se identifican los usuarios participantes y finales, interlocutores tanto en la obtención de requisitos como en la validación de los distintos productos y la aceptación final del sistema. Para ello, se actualiza el catálogo de usuarios generado previamente en el Estudio de Viabilidad del Sistema (EVS).

Dada la importancia que la colaboración de los usuarios tiene en el proceso de obtención de los requisitos, es conveniente determinar quiénes van a participar en las sesiones de trabajo, especificando sus funciones y asignando responsabilidades. Así mismo, se informa del plan de trabajo a los usuarios identificados.

El alcance de este plan de trabajo se limita al proceso de análisis.

#### **3.1.2.1. Catalogo de Usuarios**

A continuación se detallan las características que deben cumplir los usuarios finales del sistema en función de su perfil de usuario:

##### *Usuario Administrador:*

Este perfil involucra a los usuarios con mas responsabilidades y poder dentro del sistema, ya que son los únicos facultados para poder gestionar cualquier proyecto existente. Cabe destacar que además es de suma importancia la administración adecuada de los usuarios, ya que en el caso que se trabaje con información de un servidor de bases de datos de la organización, si se abusa de la utilización del mismo, podría ir esto en detrimento de la performance general diaria con los consecuentes perjuicios.

##### *Usuario Analista:*

Este perfil involucra a los usuarios que estarán a cargo de la ejecución de la tarea en si misma, desarrollo de proyectos. El mismo tendrá atribuciones para dar de alta nuevos proyectos, consultar el estado de cada uno de los proyectos a su cargo.

## **3.2. Especificación de Requisitos**

### **3.2.1. Introducción**

Este documento es una Especificación de Requisitos Software (ERS) del Sistema de Explotación de Datos para cumplimentar con la Tesis Final del Magíster en Ingeniería del Software. Esta documentación es el resultado de analizar sistemas existentes y tratar de construir una herramienta que sea útil y versátil para los usuarios.

El objetivo de la especificación es definir de manera clara y precisa todas las funcionalidades y restricciones del sistema que se desea construir. Esta especificación está sujeta a revisiones por el grupo de usuarios que se recogerán por medio de sucesivas versiones del documento, hasta alcanzar su aprobación por parte de la directora de la tesis. Una vez aprobado, servirá de base al equipo de desarrollo para la construcción del nuevo sistema.

Esta especificación se ha realizado de acuerdo el estándar “IEEE Recommended Practice for Software Requirement Specifications (ANSI/IEEE 830-1993)”. Posteriormente en el punto 3.4.2 de este texto, se expondrán a modo de esquemas gráficos de los casos de uso y sus relaciones.

#### **3.2.1.1. Objetivos y Alcance del Sistema**

El sistema fundamentalmente permitirá a los usuarios aplicar las técnicas de explotación de datos SOM y C5.0, en forma independiente o una a partir de la anterior. Los datos podrán estar ubicados en un repositorio de datos local o remoto, y el resultado se podrá guardar en una tabla en el motor de bases de datos o bien en un archivo plano en la estación de trabajo. Además se proveerá de un esquema que facilitará la administración y versionado de proyectos. El sistema contará con ayudas en línea y un asistente para la creación de un proyecto desde cero.

### **3.2.1.2. Referencias**

IEEE Recommended Practice for Software Requirement Specifications (ANSI/IEEE 1993).

### **3.2.1.3. Visión General**

Este capítulo del documento consta de tres secciones además de ésta.

En la Sección 2 se da una descripción general del sistema, con el fin de conocer las principales funciones que debe soportar, los datos asociados y los factores que afectan al desarrollo, pero sin entrar en detalles.

En la Sección 3 se definen detalladamente los requisitos que debe satisfacer el sistema, indicando cuáles son las entradas, las salidas y el proceso de cada requisito.

## **3.2.2. Descripción General**

Esta sección nos presenta una descripción general del Sistema, con el fin de conocer las principales funciones que debe soportar, los datos asociados, las restricciones impuestas, así como, cualquier factor que pueda influir en la construcción del mismo.

### **3.2.2.1. Funciones del Sistema**

A grandes rasgos las principales funciones que debe realizar el sistema son las que se describen a continuación.

#### **3.2.2.1.1. Gestionar Proyectos**

El sistema debe permitir a los usuarios crear proyectos de explotación de datos, de modo que el mismo pueda administrar varios análisis a la vez, recuperando datos de proyectos efectuados. Este también proveerá de una facilidad de versionado de proyectos, pensado en el caso que el analista quiera sobre un proyecto específico, explorar distintas alternativas sobre un mismo problema.

### 3.2.2.1.2. Aplicar algoritmo SOM sobre los datos

El sistema proveerá de las facilidades necesarias para ejecutar el algoritmo SOM sobre un lote de datos, el usuario podrá establecer los valores de los parámetros de esta metodología o bien utilizar los que por defecto estarán cargados.

### 3.2.2.1.3. Aplicar algoritmo C5.0 sobre la salida de SOM

El sistema proveerá de las facilidades necesarias para ejecutar el algoritmo C5.0 sobre la salida de la ejecución del algoritmo SOM, el usuario podrá establecer los valores de los parámetros de esta metodología o bien utilizar los que por defecto estarán cargados. Se podrán ejecutar los dos algoritmos en una sola corrida, obteniendo directamente el árbol de decisión.

### 3.2.2.2. Diagrama de Contexto

Las relaciones que debe tener el Sistema con su entorno están reflejadas en el siguiente Diagrama de Contexto (ver figura ASI 3.3).

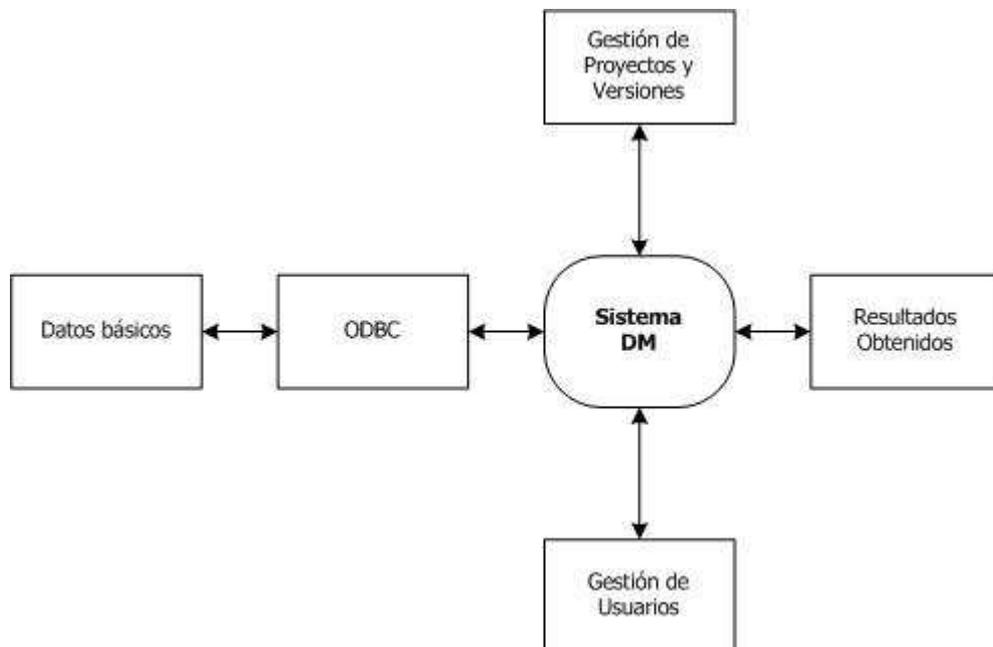


Figura ASI 3.3: diagrama de contexto.

### **3.2.2.3. Suposiciones y dependencias**

#### **3.2.2.3.1. Suposiciones**

Se asume que los requisitos descritos en este documento son estables una vez que sea aprobado. Cualquier petición de cambios en la especificación debe ser aprobada por todas las partes y gestionada por el grupo de Control de Configuración.

#### **3.2.2.3.2. Dependencias**

El sistema no establece una comunicación con sistemas externos.

### **3.2.3. Requisitos Específicos**

#### **3.2.3.1. Requisitos Funcionales**

De acuerdo a lo especificado en el ítem 2.6, se procede a describir los requisitos funcionales.

##### **3.2.3.1.1. Requisitos funcionales Generales**

###### **3.2.3.1.1.1. Ingreso al Sistema (RF01)**

###### **Introducción**

El sistema debe permitir a los usuarios ingresar al sistema solo luego de ingresar unos datos de lo identifiquen, para poder individualizarlo en los distintos proyectos.

###### **Entradas**

Las entradas son la identificación del usuario (DNI) y una clave de ingreso.

###### **Proceso**

Comprobar si se trata de un usuario dado de alta previamente y si la clave coincide con la estipulada.

### **Salida**

En caso de comprobar los datos de entrada, se habilita al usuario para operar en el sistema, de lo contrario el sistema no se ejecuta.

#### **3.2.3.1.1.2. Salir del Sistema (RF02)**

##### **Introducción**

El sistema debe permitir a los usuarios salir del sistema de manera correcta y eficiente, no dejando que las tablas abiertas, las variables de memoria sin liberar, las conexiones al SGBD abiertas, etc.

##### **Entradas**

La entrada es la orden de la opción de menú de salir del sistema.

##### **Proceso**

Comprobar que las tablas abiertas se cierren correctamente, las variables de memoria sean liberadas, las conexiones al SGBD sean cerradas, etc. y proceder a cerrar la aplicación.

##### **Salida**

Cerrar el sistema y volver al entorno de Microsoft Windows.

#### **3.2.3.1.2. Requisitos funcionales relacionados con Usuarios**

##### **3.2.3.1.2.1. Agregar Usuario (RF03)**

##### **Introducción**

El sistema debe permitir a los usuarios nuevos ingresar al sistema luego de ingresar los datos que lo identifiquen, para poder individualizarlo en los distintos proyectos.

**Entradas**

Las entradas son la identificación del usuario (DNI), nombre y apellido, una clave de ingreso, domicilio, empresa, etc.

**Proceso**

Comprobar que no se trate de un usuario dado de alta previamente y una vez cargados y comprobados los datos, podrá ingresar al sistema.

**Salida**

En caso de comprobar los datos de entrada, se habilita al usuario para operar en el sistema, de lo contrario el sistema no se ejecuta.

**3.2.3.1.2.2. Modificar Usuario (RF04)****Introducción**

El sistema debe permitir a los usuarios modificar los datos identificatorios ingresados, excepto el DNI.

**Entradas**

Las entradas son cualquiera de los datos identificatorios (nombre y apellido, clave de ingreso, domicilio, empresa, etc.) excepto el DNI.

**Proceso**

Comprobar que los datos a modificar son validos y realizar los cambios necesarios.

**Salida**

En caso de comprobar que los datos ingresados son correctos, se modifican en las tablas del sistema.

### **3.2.3.1.3. Requisitos funcionales relacionados con Proyectos**

#### **3.2.3.1.3.1. Crear Proyecto (RF05)**

##### **Introducción**

El sistema debe permitir a los usuarios crear proyectos de explotación de datos. Esta función es pensada principalmente para proveer de orden al analista, y que los distintos proyectos que lleven a cabo sean fácilmente individualizados.

##### **Entradas**

La entrada es el nombre del nuevo proyecto, internamente el sistema le estampa la fecha de creación y a que analista pertenece.

##### **Proceso**

El sistema comprueba que sea un nombre de proyecto válido, y lo carga en las tablas de proyectos del sistema.

##### **Salida**

En caso de comprobar que los datos ingresados son correctos, se crea un nuevo proyecto y el analista queda en condiciones de crear versiones del mismo.

#### **3.2.3.1.3.2. Eliminar Proyecto (RF06)**

##### **Introducción**

El sistema debe permitir a los usuarios eliminar los proyectos que tiene cargados a su nombre.

##### **Entradas**

La entrada es el nombre del proyecto (que lo elige de una lista

desplegable) y los datos identificatorios del usuario.

### **Proceso**

Comprobar que los datos del proyecto y del usuario son validos y eliminar el proyecto y los archivos asociados al mismo.

### **Salida**

Eliminación del proyecto.

### **3.2.3.1.3.3. Listar Proyectos (RF07)**

#### **Introducción**

El sistema debe permitir a los usuarios listar todos los proyectos que tiene cargados a su nombre.

#### **Entradas**

La entrada son los datos identificatorios del usuario.

#### **Proceso**

Comprobar que los datos del usuario son validos y listar los proyectos que encuentre a su nombre.

#### **Salida**

Lista de proyectos a su nombre.

### **3.2.3.1.4. Requisitos funcionales relacionados con Versiones**

#### **3.2.3.1.4.1. Crear Versiones de un Proyecto (RF08)**

##### **Introducción**

El sistema debe permitir a los usuarios crear versiones de un proyecto de explotación de datos. Esta funcionalidad es pensada para proveer al

analista la posibilidad que a un mismo proyecto lo ejecute variando algunos parámetros para poder comparar entre las diferentes salidas. Cabe destacar que cuando se crea un proyecto el sistema en forma automática da de alta la primera versión del proyecto.

### **Entradas**

La entrada es el nombre del proyecto, internamente el sistema crea una versión superior a la existente y le estampa la fecha de creación y a que analista pertenece.

### **Proceso**

El sistema comprueba que sea un nombre de proyecto y el número de versión sean válidos, y lo carga en las tablas de proyectos del sistema.

### **Salida**

En caso de comprobar que los datos ingresados son correctos, se crea una nueva versión del proyecto y el analista queda en condiciones trabajar con la versión creada.

#### **3.2.3.1.4.2. Abrir Versión (RF09)**

### **Introducción**

El sistema debe permitir a los usuarios abrir los distintos proyectos que tiene cargados a su nombre.

### **Entradas**

La entrada es el nombre del proyecto (que lo elige de una lista desplegable) y los datos identificatorios del usuario.

### **Proceso**

Comprobar que los datos del proyecto y del usuario son validos y abrir el proyecto.

### **Salida**

Apertura del proyecto elegido.

#### **3.2.3.1.4.3. Eliminar Versión (RF10)**

### **Introducción**

El sistema debe permitir a los usuarios eliminar versiones de los proyectos que tiene cargado a su nombre.

### **Entradas**

La entrada es el nombre del proyecto y la versión del mismo (lo elige de una lista desplegable) y los datos identificatorios del usuario.

### **Proceso**

Comprobar que los datos del proyecto, de la versión y del usuario son validos y eliminar la versión del proyecto y los archivos asociados al mismo.

### **Salida**

Eliminación de la versión del proyecto.

#### **3.2.3.1.4.4. Listar Versiones (RF11)**

### **Introducción**

El sistema debe permitir a los usuarios listar todas las versiones de los proyectos que tiene cargados a su nombre.

**Entradas**

La entrada son los datos identificatorios del usuario.

**Proceso**

Comprobar que los datos del usuario son validos y listar las versiones de proyectos que encuentre disponibles.

**Salida**

Lista de versiones de los proyectos que tiene el usuario a su nombre.

**3.2.3.1.5. Requisitos funcionales relacionados con la Explotación de Datos****3.2.3.1.5.1. Establecer Parámetros de SOM (RF12)****Introducción**

El sistema debe permitir a los usuarios modificar los parámetros del algoritmo de SOM, previamente a su ejecución, cabe destacar que se cargarán inicialmente valores por defecto.

**Entradas**

La entrada son los valores de los parámetros del algoritmo de SOM.

**Proceso**

Comprobar que los valores de los parámetros son validos y cargarlos en variables correspondientes para su posterior utilización.

**Salida**

Parámetros del algoritmo de SOM cargados con valores.

### **3.2.3.1.5.2. Establecer Parámetros de C5.0 (RF13)**

#### **Introducción**

El sistema debe permitir a los usuarios modificar los parámetros del algoritmo de C5.0, previamente a su ejecución, cabe destacar que se cargarán inicialmente valores por defecto.

#### **Entradas**

La entrada son los valores de los parámetros del algoritmo de C5.0.

#### **Proceso**

Comprobar que los valores de los parámetros son validos y cargarlos en variables correspondientes para su posterior utilización.

#### **Salida**

Parámetros del algoritmo de C5.0 cargados con valores.

### **3.2.3.1.5.3. Ejecutar Proceso de Explotación de Datos (RF14)**

#### **Introducción**

El sistema debe permitir a los usuarios ejecutar el proceso de Explotación de Datos, para lo cual el sistema fue creado. Se puede elegir correr los dos algoritmos en forma conjunta C5.0 a partir de la salida de SOM o bien en forma independiente.

#### **Entradas**

La entrada son los datos identificatorios del usuario, los datos del proyecto y su versión, los parámetros establecidos para los algoritmos y finalmente cuales métodos de Explotación de datos ejecutar.

**Proceso**

Comprobar que datos identificatorios del usuario, los datos del proyecto y su versión y los parámetros de los algoritmos de Explotación de Datos son validos, y proceder a ejecutar el/los algoritmo/s elegido/s sobre el conjunto de datos de entrada.

**Salida**

Resultado de ejecutar el/los algoritmo/s elegido/s sobre un conjunto de datos.

**3.2.3.2. Requisitos de Usuario**

Los usuarios serán analistas del negocio encargados de encarar proyectos de explotación de datos y con un suficiente background como para poder interpretar y analizar los resultados obtenidos. Se puede suponer que tienen conocimientos previos de informática. Tanto la aplicación como el manual de usuario de la misma deben ser intuitivos y fáciles de entender y aplicar por los analistas.

**3.2.3.3. Requisitos Tecnológicos**

En vista de que es necesario instalar la aplicación en equipos que realizaran trabajos de explotación de datos, y las fuentes de información pueden llegar a ser voluminosos, los mismo deberán ser acordes a las necesidades de procesamiento. La aplicación se ejecutará sobre PC, y estará conectada en red con la base de datos central. El sistema operativo sobre el que se debe ejecutar la aplicación es Microsoft Windows y será testeado para las versiones 2000 y XP.

La aplicación debe ser independiente del Sistema Gestor de Base de Datos (SGBD) del servidor central, aunque sí es requisito de la aplicación el que este SGBD sea relacional (SGBDR).

Para ello se utilizarán controladores de ODBC, que permiten el acceso a las bases de datos con una interfaz común, independientemente del SGBDR utilizado.

### **3.2.3.4. Requisitos de Interfaces Externos**

#### **3.2.3.4.1. Interfaces de Usuario**

La interfaz de usuario debe ser orientada a ventanas tipo WINDOWS.

#### **3.2.3.4.2. Interfaces Hardware**

No se han definido

#### **3.2.3.4.3. Interfaces Software**

La aplicación se debe comunicar con motores de base de datos a través de ODBC.

#### **3.2.3.4.4. Interfaces de Comunicación**

La conexión a la red puede establecerse por medio de una conexión directa con la red Ethernet del servidor. Esto será transparente para la aplicación la cuál, a todos los efectos, considerará que está en la misma red que el servidor.

### **3.2.3.5. Requisitos de Rendimiento**

No se establecen requisitos de rendimiento generales debido a la gran variabilidad del tamaño (en filas y columnas) de los lotes de datos a procesar, y de las diferentes alternativas de parámetros de cada algoritmo.

### **3.2.3.6. Requisitos de Desarrollo**

El ciclo de vida será de prototipado evolutivo, debiendo orientarse hacia el desarrollo de un sistema flexible que permita incorporar fácilmente cambios y nuevas funciones.

### **3.2.3.7. Restricciones de Diseño**

#### **3.2.3.7.1. Ajuste a estándares**

No se han tenido en cuenta estándares propios del dominio de la aplicación.

### **3.2.3.7.2. Limitaciones Hardware**

Se supondrá que cada estación de trabajo de la aplicación debe contar al menos con la configuración mínima que se describe:

- PC con un procesador clase Pentium II.
- Memoria RAM: 128 Mb mínimo, 256 Mb recomendado.
- Disco rígido: 10 Mb libres.

### **3.2.3.8. Atributos**

#### **3.2.3.8.1. Seguridad**

La seguridad de los datos será establecida con el Sistema Gestor de Base de Datos que se emplee y en forma local por los permisos asignados en cada carpeta de acuerdo a las características de resguardo de información provistas por el sistema de archivos NTFS.

El sistema provee de un esquema de protección de cada proyecto, a los cuales solo pueden acceder su usuario creador o un perfil denominado “Administrador”.

El procedimiento de copia de seguridad será de acuerdo a las normas de cada institución (diaria, semanal, etc.), tanto de los backups del SGBD como de los datos alojados en las estaciones de trabajo.

#### **3.2.3.8.2. Mantenimiento**

La aplicación debe ser construida con vistas a facilitar su mantenimiento, tanto debido a cambios como a ampliaciones.

### **3.2.3.9. Otros requisitos**

#### **3.2.3.9.1. Base de datos**

El Sistema Gestor de Base de Datos debe ser relacional y se accederá a él mediante los controladores de ODBC para Microsoft Windows en las versiones 2000 y XP.

#### **3.2.3.9.2. Operaciones**

No se han definido.

#### **3.2.3.9.3. Adaptación al entorno**

La aplicación debe adaptarse a la interfaz del sistema operativo Microsoft Windows en las versiones 2000 y XP.

#### **3.2.3.9.4. Políticas de Borrado**

Cada usuario será responsable por las versiones de proyectos que dé de alta, y tendrá a medida que concluye sus análisis, de realizar cierta limpieza de datos para no utilizar espacio innecesario en las estaciones de trabajo y en los SGBD.

### **3.3. Identificación de Subsistemas de Análisis**

A continuación en la figura ASI 3.4 se muestra el modelo de subsistemas de análisis, en el cual se expone que a los datos de entrada se le puede aplicar solamente SOM, que C5.0 se realiza en la salida de SOM, y que los datos de salida pueden provenir tanto de SOM como de la combinación de SOM y C5.0.

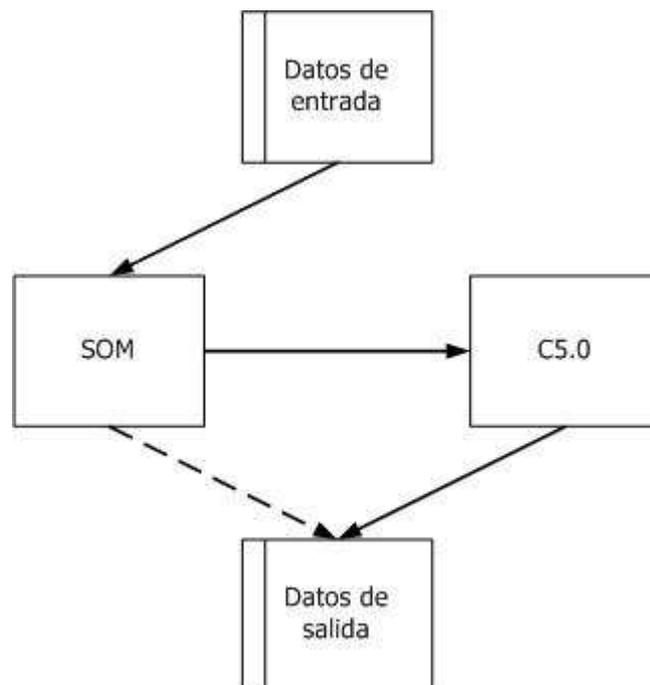


Figura ASI 3.4: Subsistemas de análisis

### 3.4. Establecimiento de Requisitos

#### 3.4.1. Especificación de Casos de Uso

Esta tarea es obligatoria en el caso de orientación a objetos, y opcional en el caso de análisis estructurado, como apoyo a la obtención de requisitos.

El objetivo de esta tarea es especificar cada caso de uso identificado en la tarea anterior, desarrollando el escenario.

Para completar los casos de uso, es preciso especificar información relativa a:

- Descripción del escenario, es decir, cómo un actor interactúa con el sistema, y cual es la respuesta obtenida.
- Precondiciones y poscondiciones.
- Identificación de interfaces de usuario.

- Condiciones de fallo que afectan al escenario, así como la respuesta del sistema (escenarios secundarios).

En escenarios complejos, es posible utilizar como técnica de especificación los diagramas de transición de estados, así como la división en casos de uso más simples, actualizando el modelo de casos de uso.

Para la obtención de esta información es imprescindible la participación activa de los usuarios.

#### **3.4.1.1. Casos de Uso**

A continuación, en la figura ASI 3.5, se muestra el diagrama de casos de uso del sistema:

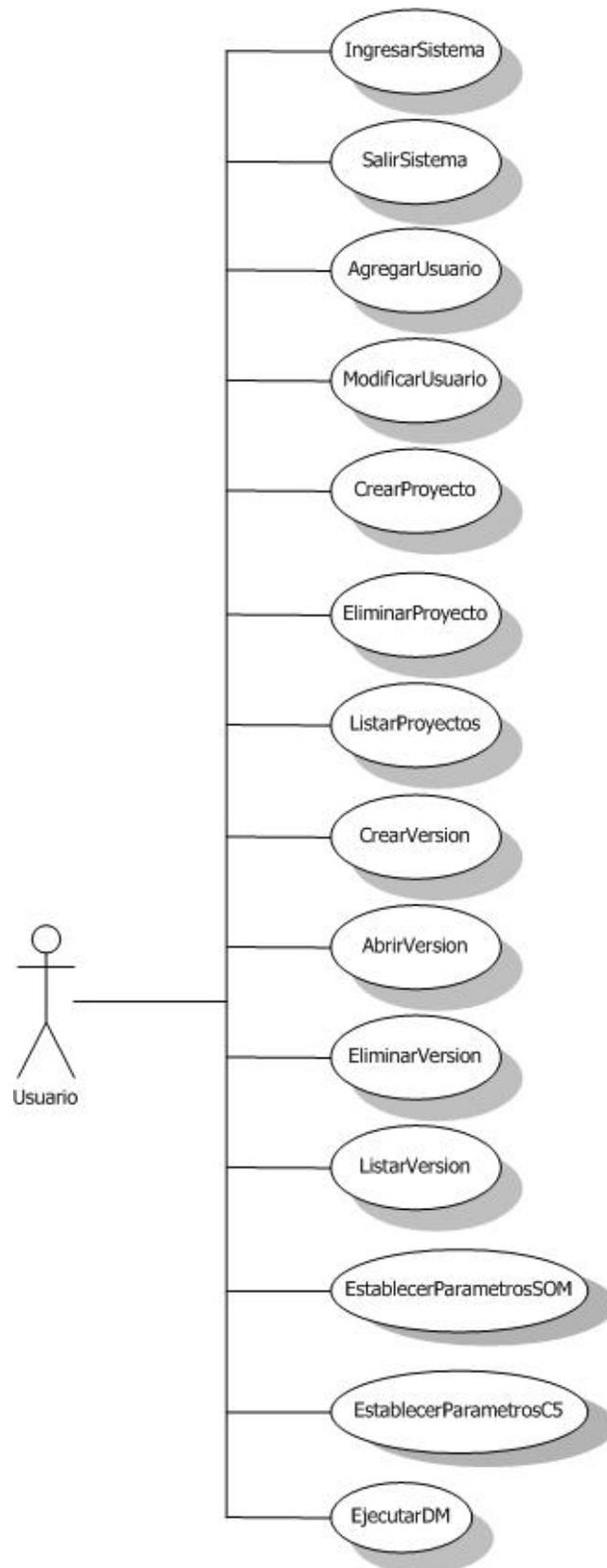


Figura ASI 3.5: Diagrama de Casos de Uso

A continuación se detalla la simbología aplicada a las flechas del diagrama:

- Las flechas de línea de punto sin extereotipo indican que entre ambos casos de uso existe una relación de Inclusión.
- Las flechas de línea llena que se identifican con el extereotipo “extends” indican que existe una relación de extensión entre ambos casos de uso.
- Las fechas de línea llena sin extereotipo indican que existen una relación de uso entre los participantes.

A continuación, en las tablas ASI 3.1 a ASI 3.14, se describen los casos de uso detallados en la figura ASI 3.5:

Caso de Uso	Ingreso al sistema (RF01)
Descripción del caso de uso	Para utilizar el sistema los usuarios debe ingresar un número identificador de usuario (por defecto usaremos el Documento Nacional de Identidad - DNI) y una clave válida.
Flujo de eventos	
Activación	Cuando el usuario activa la ejecución del sistema en cuestión se despliega una pantalla tipo “login” denominada “Datos de acceso...”.
Flujo principal	
<ol style="list-style-type: none"> <li>1. El usuario ejecuta el sistema.</li> <li>2. El sistema solicita el ingreso de los datos del usuario.</li> <li>3. El usuario carga los datos solicitados para ingresar.</li> <li>4. El sistema valida que los datos ingresados sean correctos.</li> <li>5. Si la validación es exitosa, el usuario ingresa al sistema, si no solicita la corrección de algún dato.</li> <li>6. Fin del caso de uso.</li> </ol>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: IngresarSistema     Sistema--&gt;&gt;Actor: PantallaIngreso     Actor-&gt;&gt;Sistema: IngresarUsuario     Sistema--&gt;&gt;Actor: IngresoSistema   </pre>
Flujos Alternativos	
Alternativa al paso 2	El usuario presiona el botón “Cancelar”.

Caso de Uso	Ingreso al sistema (RF01)
<p>7. El sistema debe concluir la ejecución del mismo.</p> <p>8. Fin del caso de uso.</p>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: IngresarSistema     Sistema--&gt;&gt;Actor: PantallaIngreso     Actor-&gt;&gt;Sistema: CancelarIngreso </pre>
<p>Alternativa al paso 4</p>	<p>Los datos ingresados no son correctos.</p>
<p>9. El sistema muestra un mensaje con el error encontrado. Esto sucede 3 veces, si no hay coincidencia a la tercera vez, el sistema termina su ejecución.</p> <p>10. El sistema vuelve al paso 2</p>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: IngresarSistema     Sistema--&gt;&gt;Actor: PantallaIngreso     Actor-&gt;&gt;Sistema: IngresarUsuario     Sistema--&gt;&gt;Actor: DatosErroneos     Actor-&gt;&gt;Sistema: IngresarUsuario     Sistema--&gt;&gt;Actor: IngresoSistema </pre>
<p>Requisitos especiales</p>	<p>Cuando se produzcan tres intentos de ingreso fallido consecutivos se debe abandonar el sistema, de forma similar a cuando el usuario oprime el botón de cancelar. Cabe destacar que el usuarios no podrá ingresar al sistema hasta tanto el administrador del mismo lo rehabilite.</p>
<p>Precondiciones</p>	<p>No posee</p>
<p>Poscondiciones</p>	<p>No posee</p>
<p>Puntos de extensión</p>	<p>No posee</p>

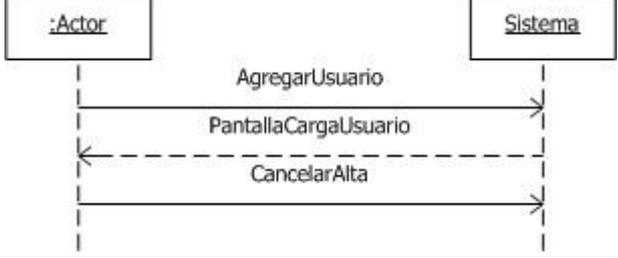
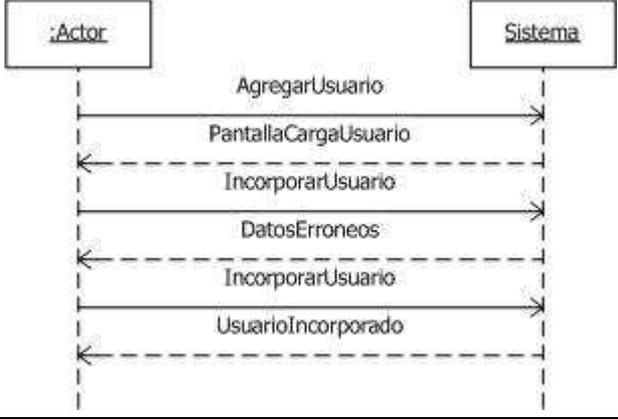
Tabla ASI 3.1: Descripción del caso de uso Ingreso al Sistema

Caso de Uso	Salir del Sistema (RF02)
<p>Descripción del caso de uso</p>	<p>Con esta opción, los usuarios cierran el sistema.</p>
<p>Flujo de eventos</p>	
<p>Activación</p>	<p>El usuario en el menú principal debe ingresar en el grupo “Proyectos” a la opción “Salir”.</p>
<p>Flujo principal</p>	

Caso de Uso	Salir del Sistema (RF02)
<p>1. El usuario selecciona la opción “Salir”.</p> <p>2. Se muestra una ventana “Si / No” con la leyenda “¿ Esta seguro que quiere abandonar la aplicación ?”.</p> <p>3. Si el usuario elije “Si”, el sistema cierra todas la tablas y variables y guarda los datos necesarios de la sesión.</p> <p>4. Fin del caso de uso.</p>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: SalirSistema     Sistema--&gt;&gt;Actor: PantallaSalirSistema     Actor-&gt;&gt;Sistema: SeleccionarSalir     Sistema--&gt;&gt;Actor: SaleSistema   </pre>
Flujos Alternativos	
Alternativa al paso 2	El usuario presiona el botón “No”.
<p>5. El sistema debe regresar a la pantalla donde se encontraba antes de que se seleccione esta opción de menú.</p> <p>6. Fin del caso de uso.</p>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: SalirSistema     Sistema--&gt;&gt;Actor: PantallaSalirSistema     Actor-&gt;&gt;Sistema: CancelaSalir   </pre>
Requisitos especiales	No posee
Precondiciones	No posee
Poscondiciones	No posee
Puntos de extensión	No posee

Tabla ASI 3.2 Descripción del caso de uso Salir del Sistema

Caso de Uso	Agregar Usuario (RF03)
Descripción del caso de uso	Para utilizar el sistema los usuarios deben estar previamente dados de alta en el mismo.
Flujo de eventos	
Activación	El usuario en la pantalla Ingreso activa la opción “Usuario nuevo”.

Caso de Uso	Agregar Usuario (RF03)
Flujo principal	
<p>1. El usuario selecciona la opción “Usuario nuevo”.</p> <p>2. El sistema solicita el ingreso de los datos del usuario a dar de alta.</p> <p>3. El usuario ingresa los datos solicitados para dar de alta.</p> <p>4. El sistema valida que los datos ingresados sean correctos y que no exista en el sistema otro usuario con igual identificador.</p> <p>5. Si la validación es exitosa, el usuario es ingresado a la base de datos del sistema, si no, solicita la corrección de algún dato.</p> <p>6. Fin del caso de uso.</p>	 <pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: AgregarUsuario     Sistema--&gt;&gt;Actor: PantallaCargaUsuario     Actor-&gt;&gt;Sistema: IncorporarUsuario     Sistema--&gt;&gt;Actor: UsuarioIncorporado   </pre>
Flujos Alternativos	
Alternativa al paso 2	El usuario presiona el botón “Cancelar”.
<p>7. El sistema debe regresar a la pantalla donde se encontraba antes de que se seleccione esta opción.</p> <p>8. Fin del caso de uso.</p>	 <pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: AgregarUsuario     Sistema--&gt;&gt;Actor: PantallaCargaUsuario     Actor-&gt;&gt;Sistema: CancelarAlta   </pre>
Alternativa al paso 4	Los datos ingresados no son correctos.
<p>9. El sistema muestra un mensaje con el error encontrado</p> <p>10. El sistema vuelve al paso 2</p>	 <pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: AgregarUsuario     Sistema--&gt;&gt;Actor: PantallaCargaUsuario     Actor-&gt;&gt;Sistema: IncorporarUsuario     Sistema--&gt;&gt;Actor: DatosErroneos     Actor-&gt;&gt;Sistema: IncorporarUsuario     Sistema--&gt;&gt;Actor: UsuarioIncorporado   </pre>
Requisitos especiales	No posee
Precondiciones	No posee
Poscondiciones	No posee

<b>Caso de Uso</b>	<b>Agregar Usuario (RF03)</b>
Puntos de extensión	No posee

Tabla ASI 3.3: Descripción del caso de uso Agregar Usuario

<b>Caso de Uso</b>	<b>Modificar Usuario (RF04)</b>
Descripción del caso de uso	Los usuarios ya dados de alta en el sistema pueden modificar algunos datos identificatorios de los mismo o su clave de acceso.
Flujo de eventos	
Activación	El usuario en el menú principal debe ingresar en el grupo “Usuarios” a la opción “Modificar datos de usuario”.
Flujo principal	
<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción “Modificar datos de usuario”.</li> <li>2. El usuario ingresa los datos a modificar.</li> <li>3. El sistema valida que los datos ingresados sean correctos.</li> <li>4. Si la validación es exitosa, los datos modificados del usuario son actualizados en la base de datos del sistema, si no solicita la corrección de algún dato.</li> <li>5. Fin del caso de uso.</li> </ol>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: ModificarUsuario     Sistema--&gt;&gt;Actor: PantallaModificarUsuario     Actor-&gt;&gt;Sistema: ModificacionesUsuario     Sistema--&gt;&gt;Actor: UsuarioModificado   </pre>
Flujos Alternativos	
Alternativa al paso 2	El usuario presiona el botón “Cancelar”.
<ol style="list-style-type: none"> <li>6. El sistema debe regresar a la pantalla donde se encontraba antes de que se seleccione esta opción de menú.</li> <li>7. Fin del caso de uso.</li> </ol>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: ModificarUsuario     Sistema--&gt;&gt;Actor: PantallaModificarUsuario     Actor-&gt;&gt;Sistema: CancelaModificacion   </pre>
Alternativa al paso 4	Los datos ingresados no son correctos.

Caso de Uso	Modificar Usuario (RF04)
<p>8. El sistema muestra un mensaje con el error encontrado</p> <p>9. El sistema vuelve al paso 2</p>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: ModificarUsuario     Sistema--&gt;&gt;Actor: PantallaModificarUsuario     Actor-&gt;&gt;Sistema: ModificacionesUsuario     Sistema--&gt;&gt;Actor: DatosErroneos     Actor-&gt;&gt;Sistema: ModificacionesUsuario     Sistema--&gt;&gt;Actor: UsuarioModificado     </pre>
Requisitos especiales	No posee
Precondiciones	No posee
Poscondiciones	No posee
Puntos de extensión	Cambiar Clave

Tabla ASI 3.4: Descripción del caso de uso Modificar Usuario

Caso de Uso	Crear Proyecto (RF05)
Descripción del caso de uso	Los usuarios del sistema pueden crear un proyecto, al cual solo ellos y el administrador accederán.
Flujo de eventos	
Activación	El usuario en el menú principal debe ingresar en el grupo "Proyectos" a la opción "Crear Proyecto".
Flujo principal	
<p>1. El usuario selecciona la opción "Crear Proyecto".</p> <p>2. El usuario ingresa la descripción del proyecto.</p> <p>3. El sistema valida que los datos ingresados sean correctos.</p> <p>4. Si la validación es exitosa, los datos del nuevo proyecto son almacenado en la base de datos del sistema, si no solicita la corrección de algún dato.</p>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: CrearProyecto     Sistema--&gt;&gt;Actor: PantallaCrearProyecto     Actor-&gt;&gt;Sistema: IngresarProyecto     Sistema--&gt;&gt;Actor: ProyectoCreado     </pre>

Caso de Uso	Crear Proyecto (RF05)
5. Fin del caso de uso.	
Flujos Alternativos	
Alternativa al paso 2	El usuario presiona el botón “Cancelar”.
6. El sistema debe regresar a la pantalla donde se encontraba antes de que se seleccione esta opción de menú. 7. Fin del caso de uso.	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: CrearProyecto     Sistema--&gt;&gt;Actor: PantallaCrearProyecto     Actor-&gt;&gt;Sistema: CancelarCreacion   </pre>
Alternativa al paso 4	Los datos ingresados no son correctos.
8. El sistema muestra un mensaje con el error encontrado 9. El sistema vuelve al paso 2	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: CrearProyecto     Sistema--&gt;&gt;Actor: PantallaCrearProyecto     Actor-&gt;&gt;Sistema: IngresarProyecto     Sistema--&gt;&gt;Actor: DatosErroneos     Actor-&gt;&gt;Sistema: IngresarProyecto     Sistema--&gt;&gt;Actor: ProyectoCreado   </pre>
Requisitos especiales	No posee
Precondiciones	No posee
Poscondiciones	No posee
Puntos de extensión	No posee

Tabla ASI 3.5: Descripción del caso de uso Crear Proyecto

Caso de Uso	Eliminar Proyecto (RF06)
Descripción del caso de uso	Los usuarios del sistema pueden eliminar un proyecto que crearon previamente.
Flujo de eventos	
Activación	El usuario en el menú principal debe ingresar en el grupo “Proyectos” a la opción “Eliminar Proyecto”.
Flujo principal	

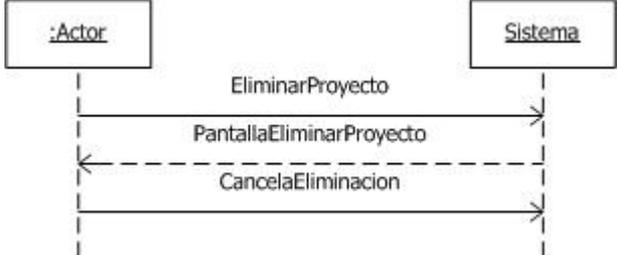
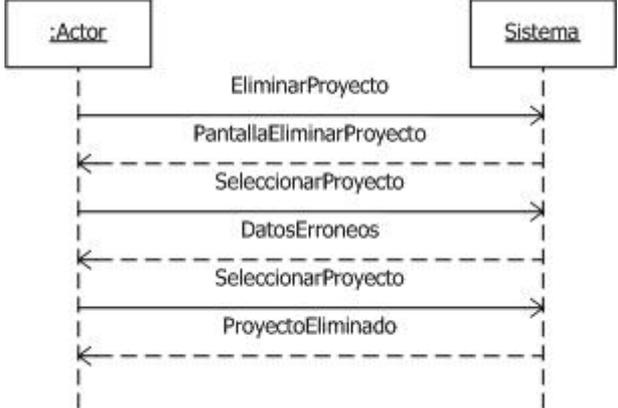
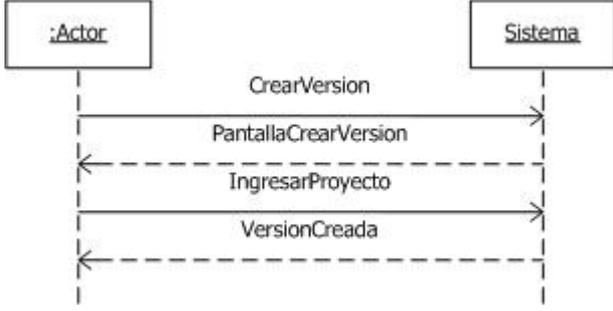
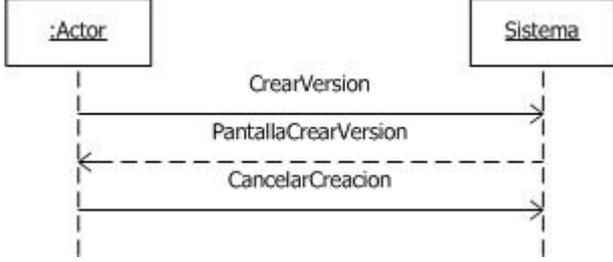
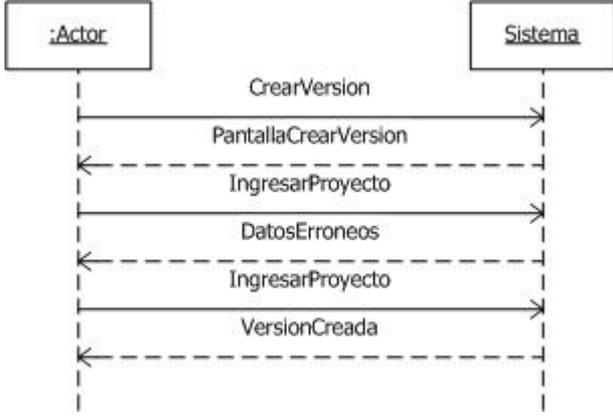
Caso de Uso	Eliminar Proyecto (RF06)
<p>1. El usuario selecciona la opción “Eliminar Proyecto”.</p> <p>2. El usuario elige de una lista el proyecto que desea eliminar.</p> <p>3. El sistema valida que los datos ingresados sean correctos.</p> <p>4. Si la validación es exitosa, el proyecto seleccionado es eliminado, si no solicita la corrección de algún dato.</p> <p>5. Fin del caso de uso.</p>	 <pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: EliminarProyecto     Sistema--&gt;&gt;Actor: PantallaEliminarProyecto     Actor-&gt;&gt;Sistema: SeleccionarProyecto     Sistema--&gt;&gt;Actor: ProyectoEliminado   </pre>
Flujos Alternativos	
Alternativa al paso 2	El usuario presiona el botón “Cancelar”.
<p>6. El sistema debe regresar a la pantalla donde se encontraba antes de que se seleccione esta opción de menú.</p> <p>7. Fin del caso de uso.</p>	 <pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: EliminarProyecto     Sistema--&gt;&gt;Actor: PantallaEliminarProyecto     Actor-&gt;&gt;Sistema: CancelaEliminacion   </pre>
Alternativa al paso 4	Los datos ingresados no son correctos.
<p>8. El sistema muestra un mensaje con el error encontrado</p> <p>9. El sistema vuelve al paso 2</p>	 <pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: EliminarProyecto     Sistema--&gt;&gt;Actor: PantallaEliminarProyecto     Actor-&gt;&gt;Sistema: SeleccionarProyecto     Sistema--&gt;&gt;Actor: DatosErroneos     Actor-&gt;&gt;Sistema: SeleccionarProyecto     Sistema--&gt;&gt;Actor: ProyectoEliminado   </pre>
Requisitos especiales	No posee
Precondiciones	No posee
Poscondiciones	No posee
Puntos de extensión	No posee

Tabla ASI 3.6: Descripción del caso de uso Eliminar Proyectos

<b>Caso de Uso</b>	<b>Listar Proyectos (RF07)</b>
Descripción del caso de uso	Los usuarios del sistema pueden listar los proyectos a los que tienen acceso.
Flujo de eventos	
Activación	El usuario en el menú principal debe ingresar en el grupo “Proyectos” a la opción “Listar Proyectos”.
Flujo principal	
1. El usuario selecciona la opción “Listar Proyectos”. 2. El sistema valida que los datos ingresados de usuario sean correctos. 3. Si la validación es exitosa, el sistema lista todos los proyectos a los que tiene acceso ese usuario, si no solicita la corrección de algún dato. 4. Fin del caso de uso.	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: ListarProyectos     Sistema--&gt;&gt;Actor: PantallaListadoProyectos           </pre>
Flujos Alternativos	
Alternativa al paso 2	Los datos ingresados no son correctos.
5. El sistema muestra un mensaje con el error encontrado 6. Fin del caso de uso	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: ListarProyectos     Sistema--&gt;&gt;Actor: DatosErroneos           </pre>
Requisitos especiales	No posee
Precondiciones	No posee
Poscondiciones	No posee
Puntos de extensión	No posee

Tabla ASI 3.7: Descripción del caso de uso Listar Proyectos

<b>Caso de Uso</b>	<b>Crear Versiones de un Proyecto (RF08)</b>
Descripción del caso de uso	Los usuarios del sistema pueden crear diferentes versiones para un mismo proyecto, a las cuales solo ellos y el administrador accederán.

Caso de Uso	Crear Versiones de un Proyecto (RF08)
Flujo de eventos	
Activación	El usuario en el menú principal debe ingresar en el grupo “Versiones” a la opción “Crear versiones de un proyecto”.
Flujo principal	
<p>1. El usuario selecciona la opción “Crear versiones de un proyecto”.</p> <p>2. El usuario elije un proyecto, de los que tiene a cargo, y el sistema le asigna una versión al mismo.</p> <p>3. El sistema valida que los datos ingresados sean correctos.</p> <p>4. Si la validación es exitosa, los datos de la nueva versión son almacenado en la base de datos del sistema, si no solicita la corrección de algún dato.</p> <p>5. Fin del caso de uso.</p>	 <pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: CrearVersion     Sistema--&gt;&gt;Actor: PantallaCrearVersion     Actor-&gt;&gt;Sistema: IngresarProyecto     Sistema--&gt;&gt;Actor: VersionCreada   </pre>
Flujos Alternativos	
Alternativa al paso 2	El usuario presiona el botón “Cancelar”.
<p>6. El sistema debe regresar a la pantalla donde se encontraba antes de que se seleccione esta opción de menú.</p> <p>7. Fin del caso de uso.</p>	 <pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: CrearVersion     Sistema--&gt;&gt;Actor: PantallaCrearVersion     Actor-&gt;&gt;Sistema: CancelarCreacion   </pre>
Alternativa al paso 4	Los datos ingresados no son correctos.
<p>8. El sistema muestra un mensaje con el error encontrado.</p> <p>9. El sistema vuelve al paso 2</p>	 <pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: CrearVersion     Sistema--&gt;&gt;Actor: PantallaCrearVersion     Actor-&gt;&gt;Sistema: IngresarProyecto     Sistema--&gt;&gt;Actor: DatosErroneos     Actor-&gt;&gt;Sistema: IngresarProyecto     Sistema--&gt;&gt;Actor: VersionCreada   </pre>
Requisitos especiales	No posee

Caso de Uso	Crear Versiones de un Proyecto (RF08)
Precondiciones	No posee
Poscondiciones	No posee
Puntos de extensión	No posee

Tabla ASI 3.8: Descripción del caso de uso Crear Versiones de un Proyecto

Caso de Uso	Abrir Versión (RF09)
Descripción del caso de uso	Los usuarios del sistema pueden abrir una versión de proyecto, en la cual trabajarán, a la cual solo ellos y el administrador accederán.
Flujo de eventos	
Activación	El usuario en el menú principal debe ingresar en el grupo “Versiones” a la opción “Abrir Versiones de un Proyecto”.
Flujo principal	
<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción “Abrir Versiones de un Proyecto”.</li> <li>2. El usuario elige de una lista el Proyecto y la Versión que desea abrir.</li> <li>3. El sistema valida que los datos ingresados sean correctos.</li> <li>4. Si la validación es exitosa, la versión del proyecto seleccionado se abre y queda activa, si no solicita la corrección de algún dato.</li> <li>5. Fin del caso de uso.</li> </ol>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: AbrirVersion     Sistema--&gt;&gt;Actor: PantallaAbrirVersion     Actor-&gt;&gt;Sistema: SeleccionarVersion     Sistema--&gt;&gt;Actor: VersionAbierta   </pre>
Flujos Alternativos	
Alternativa al paso 2	El usuario presiona el botón “Cancelar”.
<ol style="list-style-type: none"> <li>6. El sistema debe regresar a la pantalla donde se encontraba antes de que se seleccione esta opción de menú.</li> <li>7. Fin del caso de uso.</li> </ol>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: AbrirVersion     Sistema--&gt;&gt;Actor: PantallaAbrirVersion     Actor-&gt;&gt;Sistema: CancelarAbrirVersion   </pre>
Alternativa al paso 4	Los datos ingresados no son correctos.

Caso de Uso	Abrir Versión (RF09)
<p>9. El sistema muestra un mensaje con el error encontrado.</p> <p>10. El sistema vuelve al paso 2.</p>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: AbrirVersion     Sistema--&gt;&gt;Actor: PantallaAbrirVersion     Actor-&gt;&gt;Sistema: SeleccionarVersion     Sistema--&gt;&gt;Actor: DatosErroneos     Actor-&gt;&gt;Sistema: SeleccionarVersion     Sistema--&gt;&gt;Actor: VersionAbierta     </pre>
Requisitos especiales	No posee
Precondiciones	No posee
Poscondiciones	No posee
Puntos de extensión	No posee

Tabla ASI 3.9: Descripción del caso de uso Abrir Versión

Caso de Uso	Eliminar Versión (RF10)
Descripción del caso de uso	Los usuarios del sistema pueden eliminar una versión de un proyecto que creó previamente.
Flujo de eventos	
Activación	El usuario en el menú principal debe ingresar en el grupo “Versiones” a la opción “Eliminar Versiones de un Proyecto”.
Flujo principal	
<p>1. El usuario selecciona la opción “Eliminar Versiones de un Proyecto”.</p> <p>2. El usuario elige de una lista la versión que desea eliminar.</p> <p>3. El sistema valida que los datos ingresados sean correctos.</p> <p>4. Si la validación es exitosa, la versión seleccionada de un proyecto es eliminada, si no solicita la corrección de algún dato.</p> <p>5. Fin del caso de uso.</p>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: EliminarVersion     Sistema--&gt;&gt;Actor: PantallaEliminarVersion     Actor-&gt;&gt;Sistema: SeleccionarVersion     Sistema--&gt;&gt;Actor: VersionEliminada     </pre>

Caso de Uso	Eliminar Versión (RF10)
Flujos Alternativos	
Alternativa al paso 2	El usuario presiona el botón “Cancelar”.
6. El sistema debe regresar a la pantalla donde se encontraba antes de que se seleccione esta opción de menú. 7. Fin del caso de uso.	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: EliminarVersion     Sistema--&gt;&gt;Actor: PantallaEliminarVersion     Actor-&gt;&gt;Sistema: CancelaEliminacion </pre>
Alternativa al paso 4	Los datos ingresados no son correctos.
8. El sistema muestra un mensaje con el error encontrado 9. El sistema vuelve al paso 2	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: EliminarVersion     Sistema--&gt;&gt;Actor: PantallaEliminarVersion     Actor-&gt;&gt;Sistema: SeleccionarVersion     Sistema--&gt;&gt;Actor: DatosErroneos     Actor-&gt;&gt;Sistema: SeleccionarVersion     Sistema--&gt;&gt;Actor: VersionEliminada </pre>
Requisitos especiales	No posee
Precondiciones	No posee
Poscondiciones	No posee
Puntos de extensión	No posee

Tabla ASI 3.10: Descripción del caso de uso Eliminar Versión

Caso de Uso	Listar Versiones (RF11)
Descripción del caso de uso	Los usuarios del sistema pueden listar las versiones de los proyectos a los que tienen acceso.
Flujo de eventos	
Activación	El usuario en el menú principal debe ingresar en el grupo “Versiones” a la opción “Listar Versiones de Proyectos”.
Flujo principal	

Caso de Uso	Listar Versiones (RF11)
<p>1. El usuario selecciona la opción “Listar Versiones de Proyectos”.</p> <p>2. El sistema valida que los datos ingresados de usuario sean correctos.</p> <p>3. Si la validación es exitosa, el sistema lista todas las versiones de los proyectos a los que tiene acceso ese usuario.</p> <p>4. Fin del caso de uso.</p>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: ListarVersiones     Sistema--&gt;&gt;Actor: PantallaListadoVersiones   </pre>
Flujos Alternativos	
Alternativa al paso 2	Los datos ingresados no son correctos.
<p>5. El sistema muestra un mensaje con el error encontrado</p> <p>6. Fin del caso de uso</p>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: ListarVersiones     Sistema--&gt;&gt;Actor: DatosErroneos   </pre>
Requisitos especiales	No posee
Precondiciones	No posee
Poscondiciones	No posee
Puntos de extensión	No posee

Tabla ASI 3.11: Descripción del caso de uso Listar Versiones

Caso de Uso	Establecer Parámetros de SOM (RF12)
Descripción del caso de uso	Los usuarios del sistema pueden modificar los datos del algoritmo de SOM, previamente a su ejecución.
Flujo de eventos	
Activación	El usuario en el menú principal debe ingresar en el grupo “DataMining” a la opción “Parámetros de SOM”.
Flujo principal	

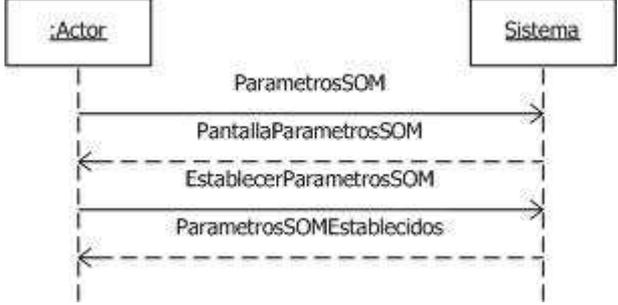
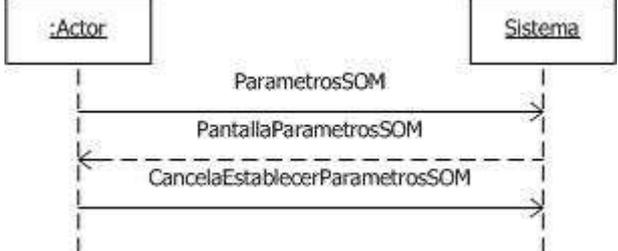
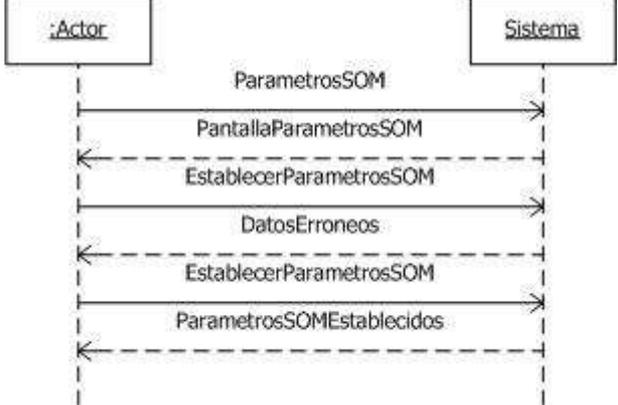
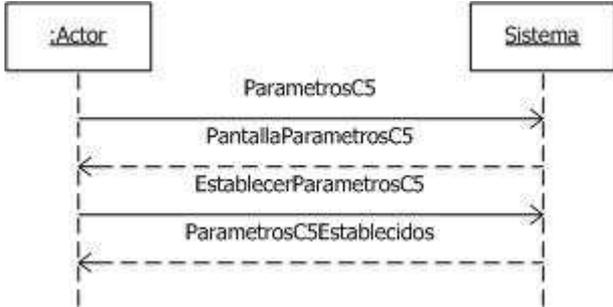
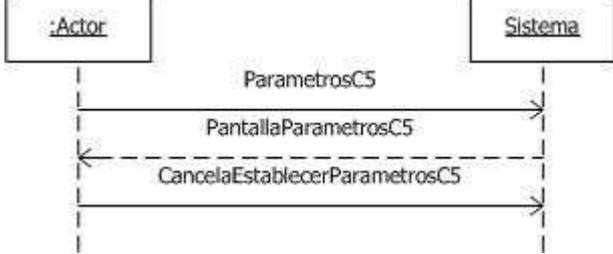
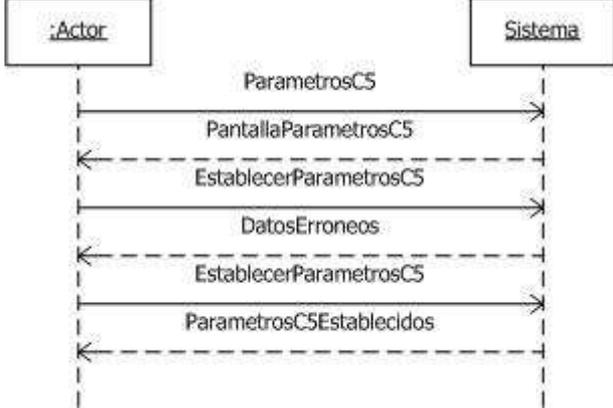
Caso de Uso	Establecer Parámetros de SOM (RF12)
<p>1. El usuario selecciona la opción “Parámetros de SOM”.</p> <p>2. El usuario elige de una lista la versión que desea eliminar.</p> <p>3. El sistema valida que los datos ingresados de usuario sean correctos.</p> <p>4. Si la validación es exitosa, el sistema establece los parámetros ingresados a considerar en la próxima corrida de SOM..</p> <p>5. Fin del caso de uso.</p>	 <pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: ParametrosSOM     Sistema--&gt;&gt;Actor: PantallaParametrosSOM     Actor-&gt;&gt;Sistema: EstablecerParametrosSOM     Sistema--&gt;&gt;Actor: ParametrosSOMEstablecidos   </pre>
Flujos Alternativos	
Alternativa al paso 2	Los datos ingresados no son correctos.
<p>6. El sistema debe regresar a la pantalla donde se encontraba antes de que se seleccione esta opción de menú.</p> <p>7. Fin del caso de uso.</p>	 <pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: ParametrosSOM     Sistema--&gt;&gt;Actor: PantallaParametrosSOM     Actor-&gt;&gt;Sistema: CancelaEstablecerParametrosSOM   </pre>
Alternativa al paso 4	Los datos ingresados no son correctos.
<p>8. El sistema muestra un mensaje con el error encontrado</p> <p>9. El sistema vuelve al paso 2</p>	 <pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: ParametrosSOM     Sistema--&gt;&gt;Actor: PantallaParametrosSOM     Actor-&gt;&gt;Sistema: EstablecerParametrosSOM     Sistema--&gt;&gt;Actor: DatosErroneos     Actor-&gt;&gt;Sistema: EstablecerParametrosSOM     Sistema--&gt;&gt;Actor: ParametrosSOMEstablecidos   </pre>
Requisitos especiales	No posee
Precondiciones	No posee
Poscondiciones	No posee
Puntos de extensión	No posee

Tabla ASI 3.12: Descripción del caso de uso Parámetros de SOM

Caso de Uso	Establecer Parámetros de C5.0 (RF13)
Descripción del caso de uso	Los usuarios del sistema pueden modificar los datos del algoritmo de C5.0, previamente a su ejecución.
Flujo de eventos	
Activación	El usuario en el menú principal debe ingresar en el grupo “DataMining” a la opción “Parámetros de C5.0”.
Flujo principal	
<p>1. El usuario selecciona la opción “Parámetros de C5.0”.</p> <p>2. El usuario elige de una lista la versión que desea eliminar.</p> <p>3. El sistema valida que los datos ingresados de usuario sean correctos.</p> <p>4. Si la validación es exitosa, el sistema establece los parámetros ingresados a considerar en la próxima corrida de C5.0.</p> <p>5. Fin del caso de uso.</p>	 <pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: ParametrosC5     Sistema--&gt;&gt;Actor: PantallaParametrosC5     Actor-&gt;&gt;Sistema: EstablecerParametrosC5     Sistema--&gt;&gt;Actor: ParametrosC5Establecidos   </pre>
Flujos Alternativos	
Alternativa al paso 2	Los datos ingresados no son correctos.
<p>6. El sistema debe regresar a la pantalla donde se encontraba antes de que se seleccione esta opción de menú.</p> <p>7. Fin del caso de uso.</p>	 <pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: ParametrosC5     Sistema--&gt;&gt;Actor: PantallaParametrosC5     Actor-&gt;&gt;Sistema: CancelaEstablecerParametrosC5   </pre>
Alternativa al paso 4	Los datos ingresados no son correctos.
<p>8. El sistema muestra un mensaje con el error encontrado</p> <p>9. El sistema vuelve al paso 2</p>	 <pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: ParametrosC5     Sistema--&gt;&gt;Actor: PantallaParametrosC5     Actor-&gt;&gt;Sistema: EstablecerParametrosC5     Sistema--&gt;&gt;Actor: DatosErroneos     Actor-&gt;&gt;Sistema: EstablecerParametrosC5     Sistema--&gt;&gt;Actor: ParametrosC5Establecidos   </pre>

Caso de Uso	Establecer Parámetros de C5.0 (RF13)
Requisitos especiales	No posee
Precondiciones	No posee
Poscondiciones	No posee
Puntos de extensión	No posee

Tabla ASI 3.13: Descripción del caso de uso Parámetros de C5.0

Caso de Uso	Ejecutar DataMining (RF14)
Descripción del caso de uso	Los usuarios del sistema pueden ejecutar el proceso de Explotación de Datos, de un conjunto determinado de datos.
Flujo de eventos	
Activación	El usuario en el menú principal debe ingresar en el grupo “DataMining” a la opción “Ejecutar DataMining”.
Flujo principal	
<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción “Ejecutar DataMining”.</li> <li>2. El usuario elige el o los algoritmos a utilizar, las fuentes de datos y lanza el proceso de Explotación de Datos.</li> <li>3. El sistema ejecuta el proceso de Explotación de Datos.</li> <li>4. Si la ejecución es exitosa, el sistema muestra los resultados, si no muestra que se produjo un error.</li> <li>5. Fin del caso de uso.</li> </ol>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: EjecutarDataMining     Sistema--&gt;&gt;Actor: PantallaEjecutarDataMining     Actor-&gt;&gt;Sistema: DataMining     Sistema--&gt;&gt;Actor: ResultadosDataMining   </pre>
Flujos Alternativos	
Alternativa al paso 2	El usuario presiona el botón “Cancelar”.
<ol style="list-style-type: none"> <li>6. El sistema debe regresar a la pantalla donde se encontraba antes de que se seleccione esta opción de menú.</li> <li>7. Fin del caso de uso.</li> </ol>	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: EjecutarDataMining     Sistema--&gt;&gt;Actor: PantallaEjecutarDataMining     Actor-&gt;&gt;Sistema: CancelaEjecutarDataMining   </pre>

Caso de Uso	Ejecutar DataMining (RF14)
Alternativa al paso 4	Los datos ingresados no son correctos.
8. El sistema muestra un mensaje con el error encontrado 9. El sistema vuelve al paso 2	<pre> sequenceDiagram     actor Actor as :Actor     participant Sistema     Actor-&gt;&gt;Sistema: EjecutarDataMining     Sistema--&gt;&gt;Actor: PantallaEjecutarDataMining     Actor-&gt;&gt;Sistema: DataMining     Sistema--&gt;&gt;Actor: ErrorEnDataMining     Actor-&gt;&gt;Sistema: DataMining     Sistema--&gt;&gt;Actor: ResultadosDataMining </pre>
Requisitos especiales	No posee
Precondiciones	No posee
Poscondiciones	No posee
Puntos de extensión	No posee

Tabla ASI 3.14: Descripción del caso de uso Ejecutar DataMining

### 3.4.2. Análisis de los Casos de Uso

#### 3.4.2.1. Identificación de Clases Asociadas a un Caso de Uso

##### 3.4.2.1.1. Identificación de clases

##### 3.4.2.2. Descripción de la Interacción de Objetos

El objetivo de esta tarea es describir la cooperación entre los objetos utilizados para la realización de un caso de uso, que ya fueron identificados en la tarea anterior.

Para representar esta información, se expondrán a modo de esquemas gráficos que contienen instancias de los actores participantes, objetos, y la secuencia de mensajes intercambiados entre ellos. Se pueden establecer criterios para determinar qué tipo de objetos y mensajes se va a incluir en este diagrama, como por ejemplo: si se incluyen objetos y llamadas a bases de datos, objetos de interfaz de usuario, de control, etc. Estos diagramas pueden ser tanto de secuencia como de colaboración, y su uso depende de si se quieren centrar en la secuencia cronológica o en cómo es la comunicación entre los

objetos.

En aquellos casos en los que se especifique más de un escenario para un caso de uso, puede ser conveniente representar cada uno de ellos en un diagrama de interacción. También es recomendable, sobre todo en el caso anterior, completar los diagramas con una descripción textual.

### 3.4.2.2.1. Identificación de la Relación entre Objetos

A continuación en las figuras ASI 3.6 a ASI 3.19, se muestran los esquemas de relaciones entre objetos del sistema.

Caso de Uso Ingreso al sistema (RF01):

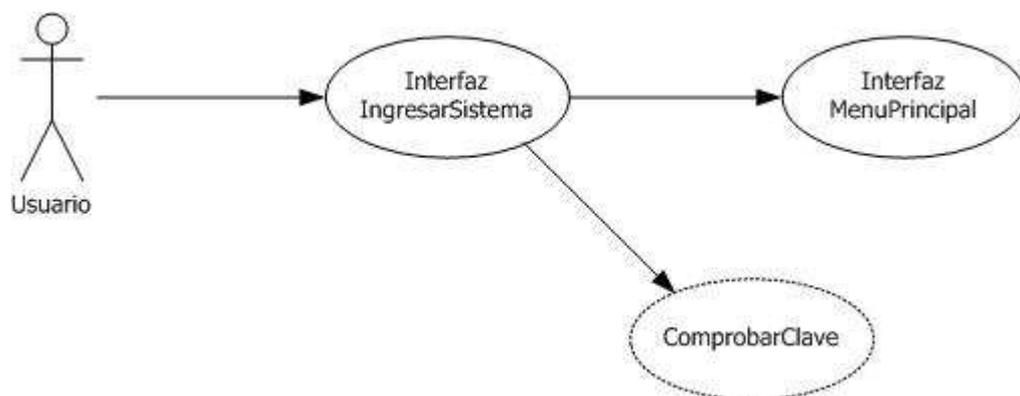


Figura ASI 3.6: Esquema de relación del caso de uso Ingreso al sistema.

Caso de Uso Agregar Usuario (RF03):

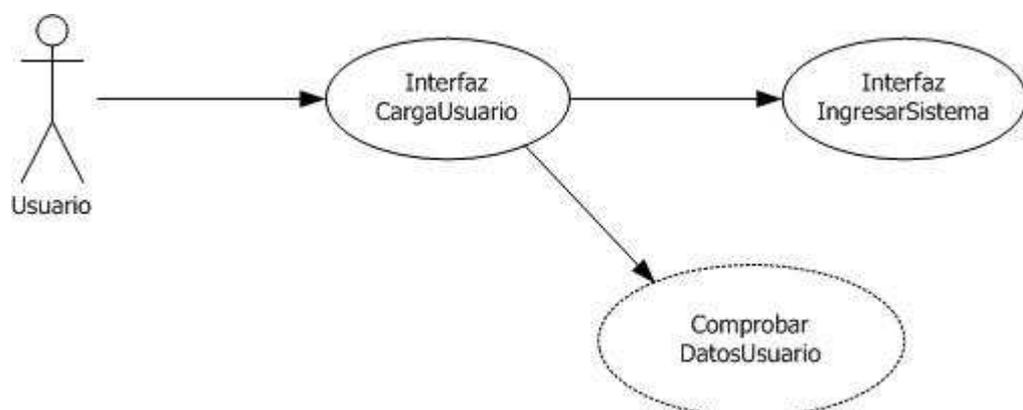


Figura ASI 3.7: Esquema de relación del caso de uso Agregar Usuario.

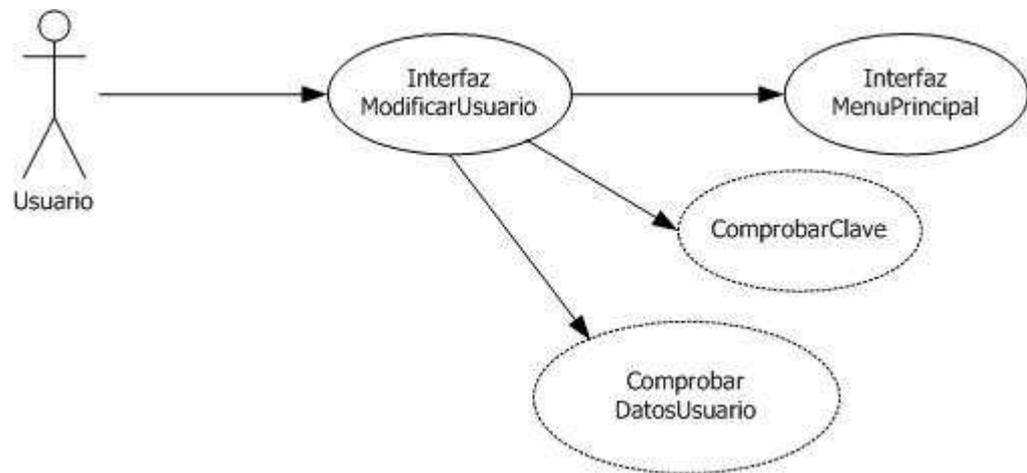
Caso de Uso Modificar Usuario (RF04):

Figura ASI 3.8: Esquema de relación del caso de uso Modificar Usuario.

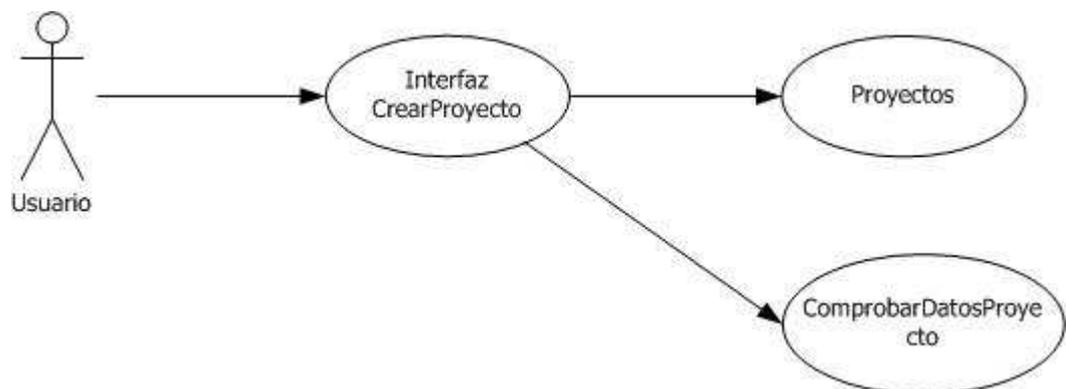
Caso de Uso Crear Proyecto (RF05):

Figura ASI 3.9: Esquema de relación del caso de uso Crear Proyecto.

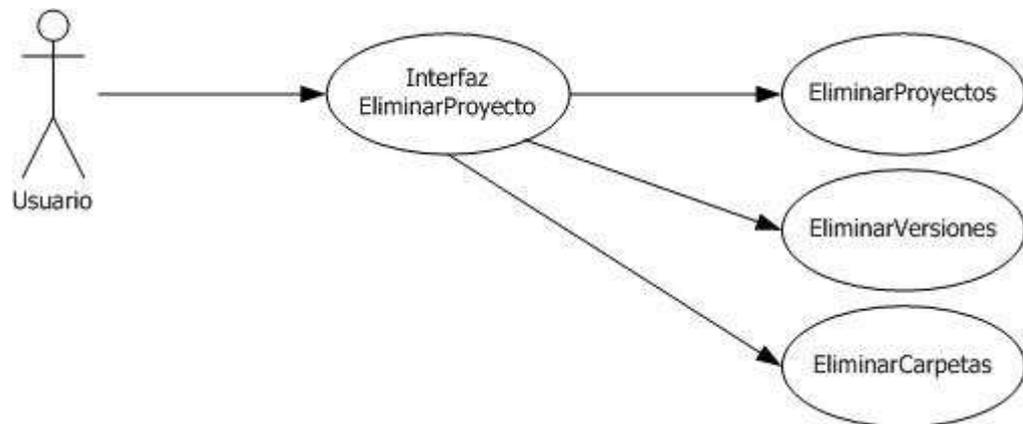
Caso de Uso Eliminar Proyecto (RF06):

Figura ASI 3.10: Esquema de relación del caso de uso Eliminar Proyecto.

Caso de Uso Listar Proyectos (RF07):

Figura ASI 3.11: Esquema de relación del caso de uso Listar Proyectos.

Caso de Uso Salir del Sistema (RF02):

Figura ASI 3.12: Esquema de relación del caso de uso Salir del Sistema.

Caso de Uso Crear Versiones de un Proyecto (RF08):

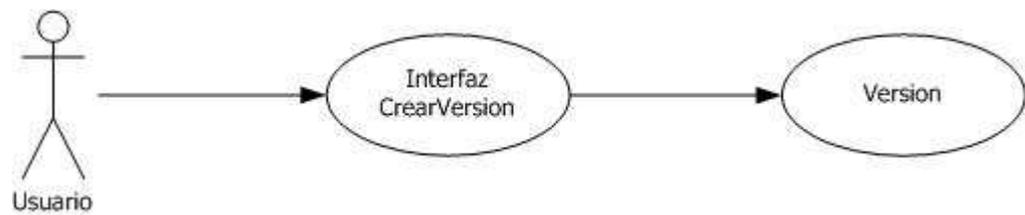


Figura ASI 3.13: Esquema de relación del caso de uso Crear Versiones de un Proyecto.

Caso de Uso Abrir Versión (RF09):

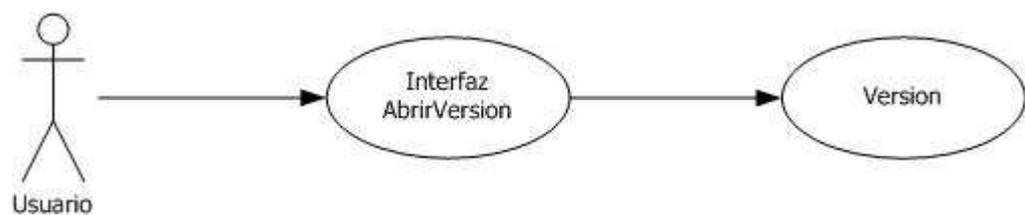


Figura ASI 3.14: Esquema de relación del caso de uso Abrir Versión.

Caso de Uso Eliminar Versión (RF10):

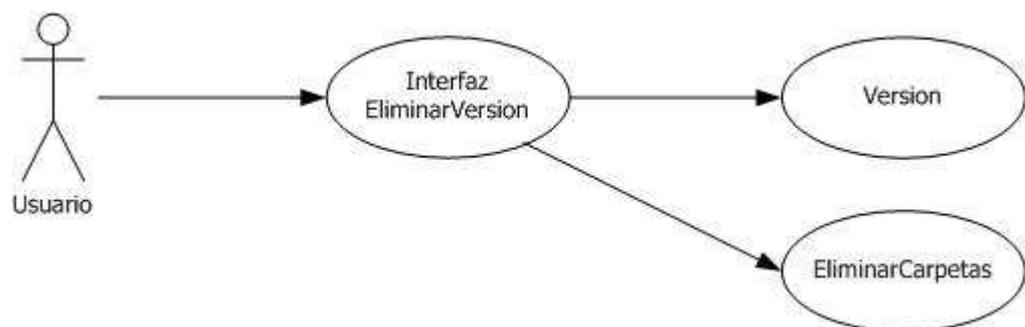


Figura ASI 3.15: Esquema de relación del caso de uso Eliminar Versión.

Caso de Uso Listar Versiones (RF11):



Figura ASI 3.16: Esquema de relación del caso de uso Listar Versiones.

Caso de Uso Establecer Parámetros de SOM (RF12):

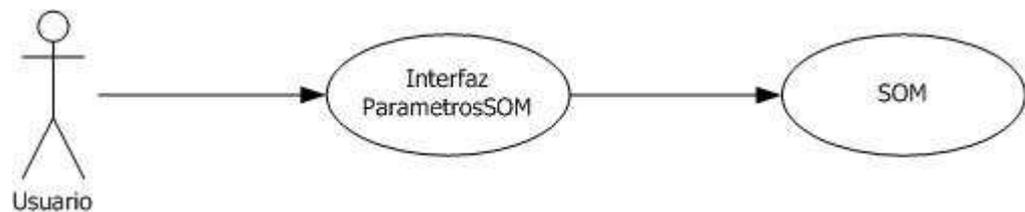


Figura ASI 3.17: Esquema de relación del caso de uso Establecer Parámetros de SOM.

Caso de Uso Establecer Parámetros de C5.0 (RF13):

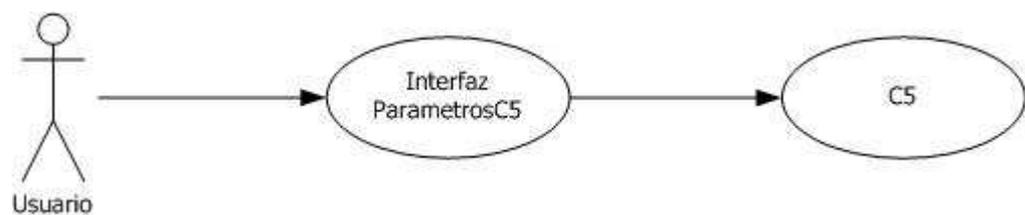


Figura ASI 3.18: Esquema de relación del caso de uso Establecer Parámetros de C5.0.

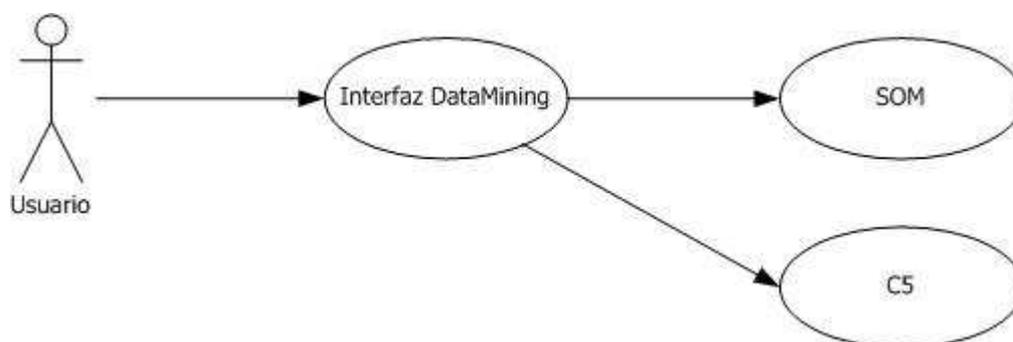
Caso de Uso Ejecutar Proceso de Explotación de Datos (RF14):

Figura ASI 3.19: Esquema de relación del caso de uso Ejecutar Proceso de Explotación de Datos.

### 3.4.3. Análisis de Clases

El objetivo de esta actividad que sólo se realiza en el caso de Análisis Orientado a Objetos es describir cada una de las clases que ha surgido, identificando las responsabilidades que tienen asociadas, sus atributos, y las relaciones entre ellas. Para esto, se debe tener en cuenta la normativa establecida en la tarea Especificación de Estándares y Normas, de forma que el modelo de clases cumpla estos criterios, con el fin de evitar posibles inconsistencias en el diseño.

Teniendo en cuenta las clases identificadas en la actividad Análisis de los Casos de Uso, se elabora el modelo de clases para cada subsistema. A medida que avanza el análisis, dicho modelo se va completando con las clases que vayan apareciendo, tanto del estudio de los casos de uso, como de la interfaz de usuario necesaria para el sistema de información.

#### 3.4.3.1. Identificación de Responsabilidades y Atributos

El objetivo de esta tarea es identificar las responsabilidades y atributos relevantes de una clase. Las responsabilidades de una clase definen la funcionalidad de esa clase, y están basadas en el estudio de los papeles que desempeñan sus objetos dentro de los distintos casos de uso. A partir de estas responsabilidades, se puede comenzar a encontrar las operaciones que van a pertenecer a la clase. Estas deben ser

relevantes, simples, y participar en la descripción de la responsabilidad. Los atributos de una clase especifican propiedades de la clase, y se identifican por estar implicados en sus responsabilidades. Los tipos de estos atributos deberían ser conceptuales y conocidos en el dominio.

De manera opcional, se elabora una especificación para cada clase, que incluye: la lista de sus operaciones y las clases que colaboran para cubrir esas operaciones y una descripción de las responsabilidades, atributos y operaciones de esa clase. Para aquellas clases cuyo comportamiento dependa del estado en el que se encuentren se realiza, también de manera opcional, un diagrama de transición de estados.

#### 3.4.3.1.1. Definición de Responsabilidades y Atributos

A continuación, en la tabla ASI 3.15, se muestran las distintas clases del sistema junto con sus responsabilidades y atributos:

Clase	Responsabilidades	Atributos
C5	Representa al algoritmo C5.0, sus parámetros y acciones.	<ul style="list-style-type: none"> <li>➤ DNI</li> <li>➤ idProyecto</li> <li>➤ idVersion</li> <li>➤ LimpiezaAtributos</li> <li>➤ Reglas</li> <li>➤ OrdUtilidad</li> <li>➤ OrdUtilidadBandas</li> <li>➤ Boost</li> <li>➤ BoostEj</li> <li>➤ SubcjoValores</li> <li>➤ UsarEj</li> <li>➤ UsarEjPorc</li> <li>➤ LockEj</li> <li>➤ ValidacionCruzada</li> <li>➤ ValidacionCruzadaHojas</li> <li>➤ IgnorarArchivoCostos</li> <li>➤ FuzzyThresholds</li> <li>➤ PodaGlobal</li> </ul>

Clase	Responsabilidades	Atributos
		<ul style="list-style-type: none"> <li>➤ PodaCF</li> <li>➤ MinimoCasos</li> </ul>
Proyecto	Representa a los proyectos que se llevan a cabo.	<ul style="list-style-type: none"> <li>➤ idProyecto: número único identificatorio de un proyecto.</li> <li>➤ DNI: número único identificatorio del usuario propietario del proyecto.</li> <li>➤ NombreProyecto: descripción textual del proyecto en cuestión.</li> <li>➤ FechaCreacion: fecha de creación del proyecto.</li> </ul>
SOM	Representa al algoritmo SOM, sus parámetros y acciones.	<ul style="list-style-type: none"> <li>➤ DNI</li> <li>➤ idProyecto</li> <li>➤ idVersion</li> <li>➤ CantidadCiclosEntrenamiento</li> <li>➤ CantidadObservaciones</li> <li>➤ CantidadVariable</li> <li>➤ CantidadNeuronasMapa</li> </ul>
Usuario	Representa a los usuarios del sistema, a sus datos y diferentes validaciones.	<ul style="list-style-type: none"> <li>➤ DNI: número único identificatorio del usuario.</li> <li>➤ Nombre: nombre y apellido del usuario.</li> <li>➤ Clave: clave asociada al usuario para ingresar al sistema.</li> <li>➤ Domicilio: domicilio del usuario.</li> <li>➤ CP: código postal del domicilio del usuario.</li> <li>➤ Localidad: localidad donde se encuentra el domicilio del usuario.</li> <li>➤ Provincia: provincia en que se encuentra ubicado el domicilio del usuario.</li> <li>➤ Pais: país en que se encuentra ubicado el domicilio del usuario.</li> </ul>

Clase	Responsabilidades	Atributos
		<ul style="list-style-type: none"> <li>➤ Telefono: número telefónico del usuario.</li> <li>➤ Email: dirección de correo electrónico del usuario.</li> <li>➤ Empresa: empresa a la que pertenece el usuario.</li> </ul>
Version	Representa a las versiones de cada proyecto proyectos que se lleva a cabo.	<ul style="list-style-type: none"> <li>➤ idVersion: número identificador de una versión de un proyecto.</li> <li>➤ idProyecto: número único identificador de un proyecto.</li> <li>➤ FechaCreacion: fecha de creación de la versión.</li> <li>➤ PCid: identificación de la PC en la cual se creo la versión del proyecto.</li> <li>➤ Path: camino dentro de las unidades de disco donde se encuentra la versión del proyecto.</li> <li>➤ DSN: (Data Source Name) Nombre de origen de datos</li> </ul>

Tabla ASI 3.15: Descripción de clases

### 3.4.3.2. Identificación de Asociaciones y Agregaciones

#### 3.4.3.2.1. Diagrama de Clases donde se identifican Asociaciones y Agregaciones

En la figura ASI 3.20 se muestra el diagrama de clases del sistema en cuestión. Se especifica para cada clase sus campos y mensajes.

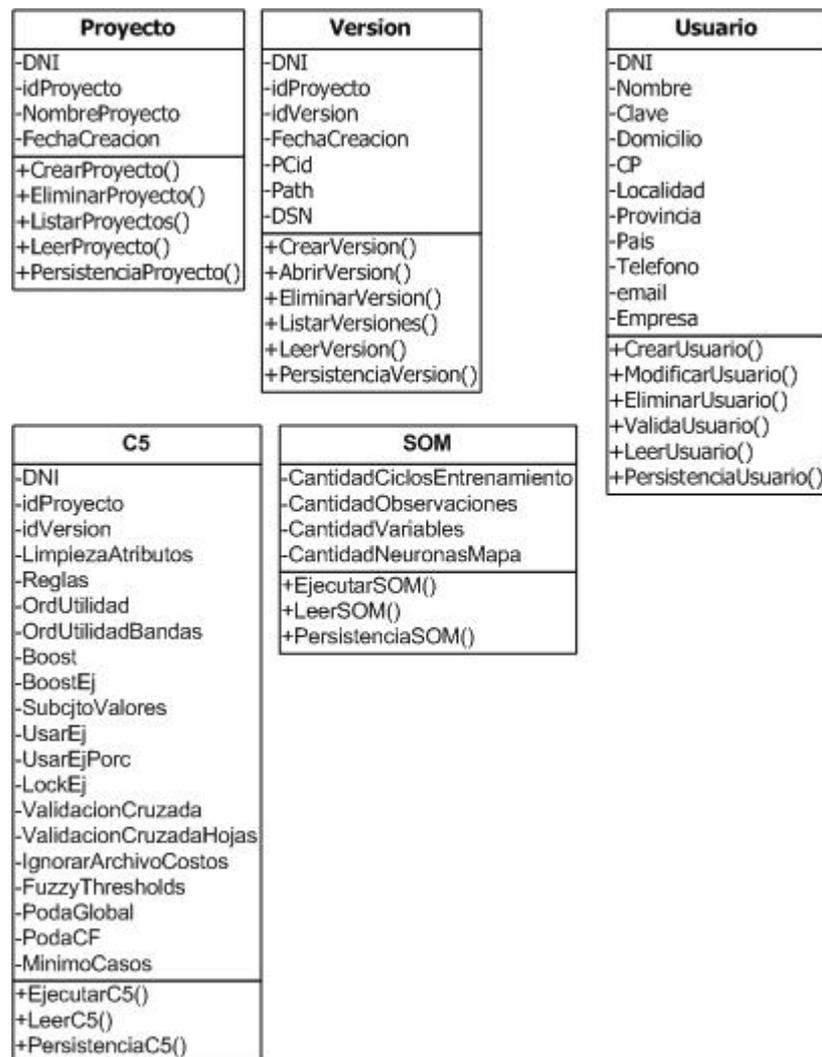


Figura ASI 3.20: diagrama de clases

### 3.5. Especificación de Interfaz de Usuario

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

Al comienzo de este análisis es necesario seleccionar el entorno en el que es operativa la interfaz, considerando estándares internacionales y de la instalación, y establecer las directrices aplicables en los procesos de diseño y construcción. El propósito es construir una interfaz de usuario acorde a sus necesidades, flexible, coherente, eficiente y sencilla de utilizar, teniendo en cuenta la facilidad de cambio a otras plataformas, si fuera necesario.

Se identifican los distintos grupos de usuarios de acuerdo con las funciones que realizan, conocimientos y habilidades que poseen, y características del entorno en el que trabajan. La identificación de los diferentes perfiles permite conocer mejor las necesidades y particularidades de cada uno de ellos.

Asimismo, se determina la naturaleza de los procesos que se llevan a cabo (en lotes o en línea). Para cada proceso en línea se especifica qué tipo de información requiere el usuario para completar su ejecución realizando, para ello, una descomposición en diálogos que refleje la secuencia de la interfaz de pantalla tipo carácter o pantalla gráfica.

Finalmente, se define el formato y contenido de cada una de las interfaces de pantalla especificando su comportamiento dinámico.

Se propone un flujo de trabajo muy similar para desarrollos estructurados y orientados a objetos, coincidiendo en la mayoría de las tareas, si bien es cierto que en orientación a objetos, al identificar y describir cada escenario en la especificación de los casos de uso, se hace un avance muy significativo en la toma de datos para la posterior definición de la interfaz de usuario.

Como resultado de esta actividad se genera la especificación de interfaz de usuario, como producto que engloba los siguientes elementos:

- Principios generales de la interfaz.
- Catálogo de perfiles de usuario.
- Descomposición funcional en diálogos.

- Catálogo de controles y elementos de diseño de interfaz de pantalla.
- Formatos individuales de interfaz de pantalla.
- Modelo de navegación de interfaz de pantalla.
- Formatos de impresión.
- Prototipo de interfaz interactiva.
- Prototipo de interfaz de impresión.

### **3.5.1. Especificación de Principios Generales de la Interfaz**

El objetivo de esta tarea es especificar los estándares, directrices y elementos generales a tener en cuenta en la definición de la interfaz de usuario, tanto para la interfaz interactiva (gráfica o carácter), como para los informes y formularios impresos.

En primer lugar, se selecciona el entorno de la interfaz interactiva (gráfico, carácter, etc.), siguiendo estándares internacionales y de la instalación, y se determinan los principios de diseño de la interfaz de usuario, contemplando:

- Directrices generales en cuanto a la interfaz y aspectos generales de interacción.
- Principios de composición de pantallas y criterios de ubicación de los distintos elementos dentro de cada formato.
- Normas para los mensajes de error y aviso, codificación, presentación y comportamientos.
- Normas para la presentación de ayudas.

Hay que establecer criterios similares para la interfaz impresa:

- Directrices generales.
- Principios de composición de informes y formularios.

- Normas de elaboración, distribución y salvaguarda de la información.

#### **3.5.1.1. Principios Generales de la Interfaz**

La interfaz de usuario será gráfica e interactiva, del tipo estándar utilizado en todos los aplicativos basados en ventanas.

Los lineamientos principales para la construcción de la interfaz de usuarios son los siguientes:

- La activación de las distintas operaciones del sistema se producen mediante una barra de menús y botones opcionales.
- El acceso a los proyectos se realiza mediante opciones que mostrarán las algunas características del mismo y el detalle de las versiones existentes, para que el usuario pueda decidir a que versión desea acceder.
- Las pantallas tendrán, en general, un botón para aceptar los datos provistos y otro para cancelarlos y, dependiendo de la funcionalidad provista, botones auxiliares para realizar otro tipo de operaciones.
- Los mensajes de error se mostrarán mediante pantallas emergentes.
- Cualquier operación de cancelación, cierre, establecimiento de versiones, etc. de una pantalla exigirá la confirmación por parte del usuario.

#### **3.5.2. Especificación de Formatos Individuales de la Interfaz de Pantalla**

El objetivo de esta tarea es especificar cada formato individual de la interfaz de pantalla, desde el punto de vista estático. Para cada proceso en línea identificado en la tarea anterior o en la especificación de los casos de uso, y teniendo en cuenta los formatos estándar definidos en la tarea Especificación de Principios Generales de la Interfaz, se definen los formatos individuales de la interfaz de pantalla requerida para completar la especificación de cada diálogo.

En el caso de un análisis orientado a objetos, estos formatos individuales van completando las especificaciones de los casos de uso.

En un análisis estructurado se tiene en cuenta, para la realización de esta tarea, el modelo de datos y el modelo de procesos generados en paralelo en las actividades Elaboración del Modelo de Datos y Elaboración del Modelo de Procesos. También se considera el catálogo de requisitos, para especificar las interfaces relacionadas con las consultas.

En la definición de cada interfaz de pantalla se deben definir aquellos aspectos considerados de interés para su posterior diseño y construcción:

- Posibilidad de cambio de tamaño, ubicación, modalidad (modal del sistema, modal de aplicación), etc.
- Dispositivos de entrada necesarios para su ejecución.
- Conjunto y formato de datos asociados, identificando qué datos se usan y cuáles se generan como consecuencia de su ejecución.
- Controles y elementos de diseño asociados, indicando cuáles aparecen inicialmente activos e inactivos al visualizar la interfaz de pantalla.

#### **3.5.2.1. Especificación del Comportamiento Dinámico de la Interfaz**

El objetivo de esta tarea es definir los flujos entre los distintos formatos de interfaz de pantalla, y también dentro del propio formato. Este comportamiento se describe mediante un modelo de navegación de interfaz de pantalla.

En este modelo se completan las interfaces de usuario que existen en el sistema y la forma en que las mismas pueden navegarse.

A continuación en la figura ASI 3.21 se muestra las distintas interfaces del sistema y la forma en que se vinculan entre ellas.

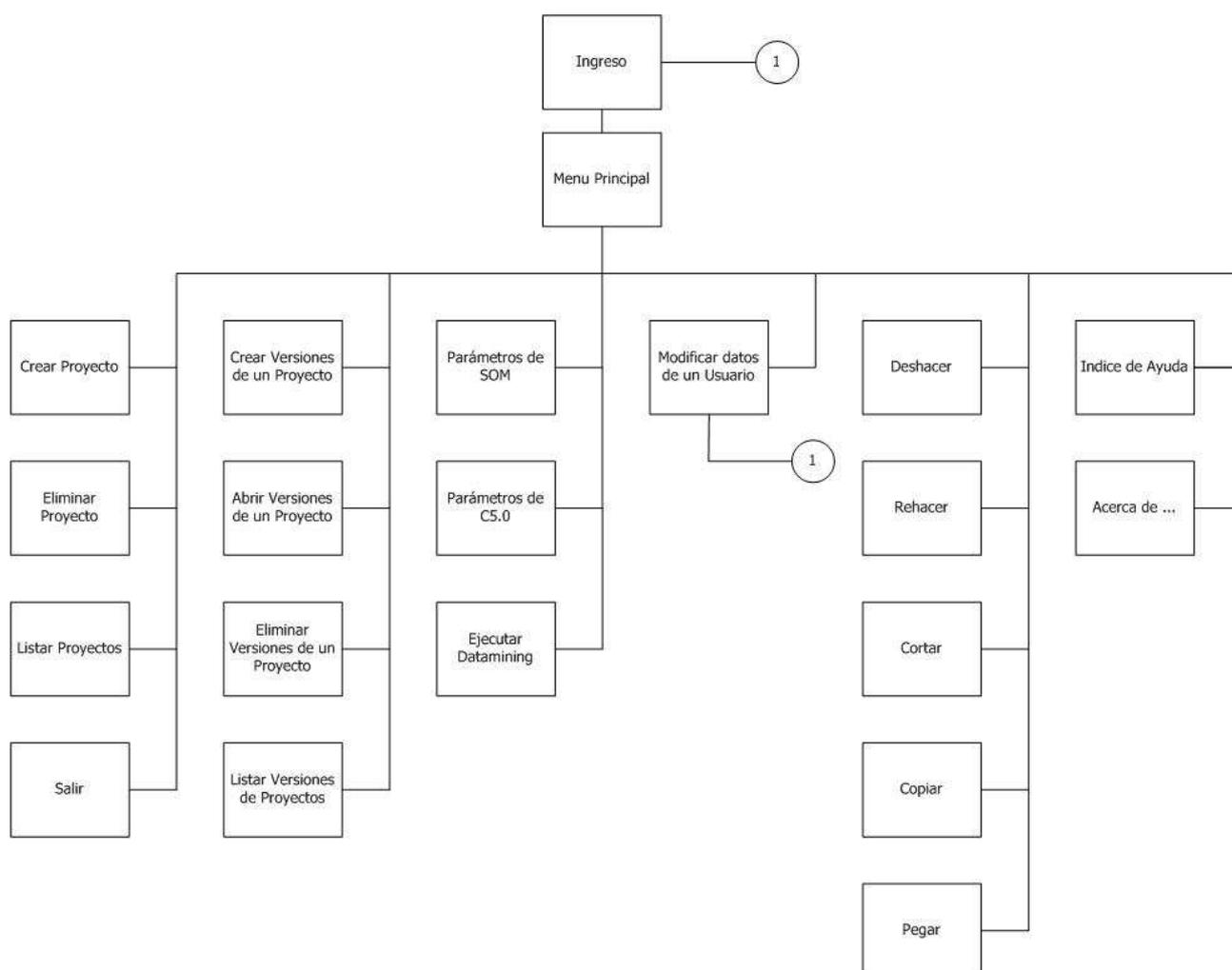


Figura ASI 3.21: Diagrama de Jerarquía de Pantallas

### 3.5.2.1.1. Descripción de las características generales de cada pantalla

#### *Ingreso:*

Es la ventana de ingreso al sistema, en la cual el usuario ingresa su código de usuario (DNI) y clave, para que el sistema habilite su ingreso.

#### *Menú Principal:*

Es la ventana principal del sistema, fondo de las demás interfaces. Contiene una barra de menú para que el usuario seleccione la función a realizar en el sistema.

### *Crear Proyecto:*

En esta ventana, el usuario ingresa los datos del nuevo proyecto, cargando el nombre del proyecto y la ubicación del mismo como demás datos mas significativos.

### *Eliminar Proyectos*

En esta pantalla, el usuario puede seleccionar de una lista de proyectos que tiene bajo su responsabilidad. Una vez elegido el proyecto, el sistema pregunta si esta seguro de realizar la operación de borrado del mismo. Se despliega el nombre del proyecto y la fecha de su creación. Cabe destacar que se eliminan todas las versiones existentes del proyecto.

### *Listar Proyectos*

En esta pantalla, el usuario puede realizar un listado (por impresora o por pantalla) de los proyectos que tiene bajo su responsabilidad. Se lista el nombre del proyecto y la fecha de su creación.

### *Crear versiones de un proyecto*

Para un proyecto previamente elegido, se crea una nueva versión del mismo. Se carga un número de identificación de la versión y la fecha de establecimiento.

### *Abrir versiones de un proyecto*

En esta pantalla, el usuario puede seleccionar de una lista de versiones, de un proyecto previamente elegido o no. Se despliega el número de versión y la fecha de su creación.

### *Eliminar versiones de un proyecto*

En esta pantalla, el usuario puede seleccionar de una lista de versiones de proyectos. Una vez elegida la versión, el sistema pregunta si esta seguro de realizar la operación de borrado de la misma. Se despliegan el nombre del

proyecto, el número de versión y la fecha de su creación.

#### *Listar versiones proyectos*

En esta pantalla, el usuario puede realizar un listado (por impresora o por pantalla) de las versiones de un proyecto que tiene bajo su responsabilidad. Se lista el nombre del proyecto, número de versión y ambas fechas de su creación.

#### *Parámetros de SOM*

En esta pantalla, el usuario puede modificar los valores establecidos de los parámetros del algoritmo SOM. Una vez establecidos estos parámetros, cuando se ejecute la operación de Explotación de Datos, los mismos serán los utilizados por el algoritmo. El ámbito en el que se conservarán estos parámetros, será la versión del proyecto.

#### *Parámetros de C5.0*

En esta pantalla, el usuario puede modificar los valores establecidos de los parámetros del algoritmo C5.0. Una vez establecidos estos parámetros, cuando se ejecute la operación de Explotación de Datos, los mismos serán los utilizados por el algoritmo. El ámbito en el que se conservarán estos parámetros, será la versión del proyecto.

#### *Ejecutar DataMining*

En esta opción se establece que algoritmo se empleará el en proceso e inicia el mismo.

#### *Modificar / Ingresar datos de usuario*

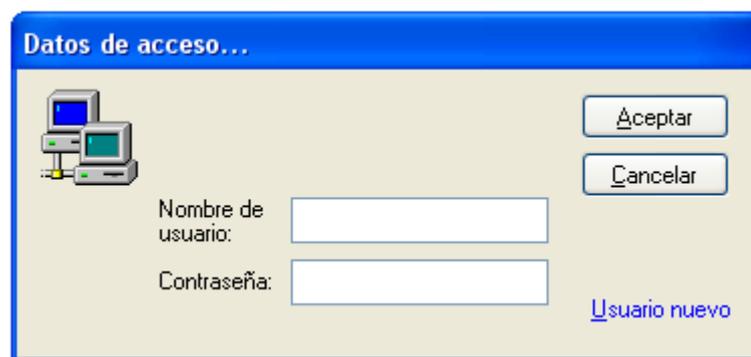
En esta opción se realiza la carga de los datos del usuarios del sistema. Existen campos obligatorios y optativos. Esta opción sirve tanto para dar de alta un usuario como para modificar el perfil de cada uno.

### 3.5.3. Prototipos de Interfaz

A continuación se detallan los prototipos de las pantallas del sistema:

#### *Pantalla de Ingreso al sistema*

A continuación, en la Figura ASI 3.22, se muestra el diseño de la pantalla de ingreso al sistema. Esta pantalla presenta dos campos de texto, en los cuales se debe ingresar el identificador del usuario y la clave para validar el ingreso al sistema. En caso que sea la primera vez que el usuario ingresa al sistema, tiene a través del botón “Usuario nuevo” que completar una serie de datos que se solicitan obligatoriamente y opcionalmente otro. Una vez realizado esto vuelve el sistema a esta pantalla con los campos Nombre de usuario y Contraseña cargados.



El prototipo de la pantalla de ingreso al sistema, titulado "Datos de acceso...", muestra un icono de un computador a la izquierda. A la derecha del icono hay dos botones: "Aceptar" y "Cancelar". Debajo del icono hay dos campos de texto: "Nombre de usuario:" y "Contraseña:". En la parte inferior derecha del formulario hay un botón de texto "Uuario nuevo" (con un error de ortografía).

Figura ASI 3.22: pantalla de ingreso al sistema

---

*Pantalla Menú principal sin menús activados*

A continuación, en la Figura ASI 3.23, se muestra el diseño de la pantalla de Menú Principal del sistema. Esta pantalla una barra de menú en la parte superior y un panel vacío en la parte inferior. Cada una de las opciones del menú despliega una lista de ítems a seleccionar que se detallan mas a delante.

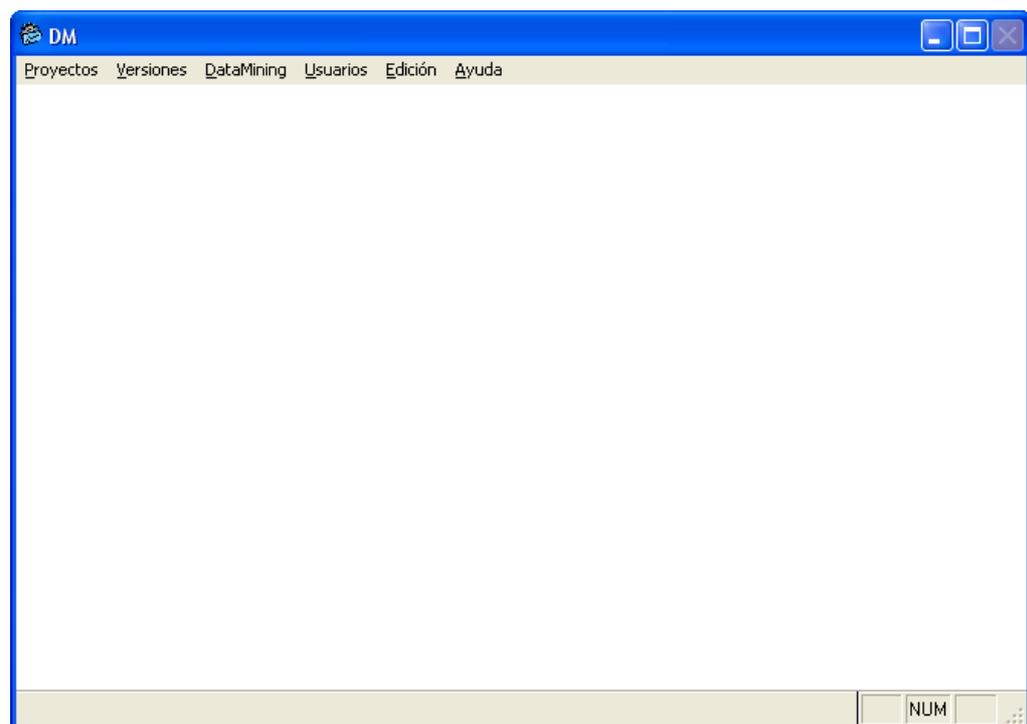


Figura ASI 3.23: pantalla de menú principal del sistema

*Pantalla Menú principal, con el menú de “Proyectos” activado*

A continuación, en la Figura ASI 3.24, se muestra el diseño de la pantalla de Menú Principal del sistema cuando se activa el menú de Proyectos. El cual posee cuatro opciones principales: Crear, Abrir, Eliminar, Listar Proyectos y Salir. Por último la quinta opción permite salir del sistema, volviendo al sistema operativo.

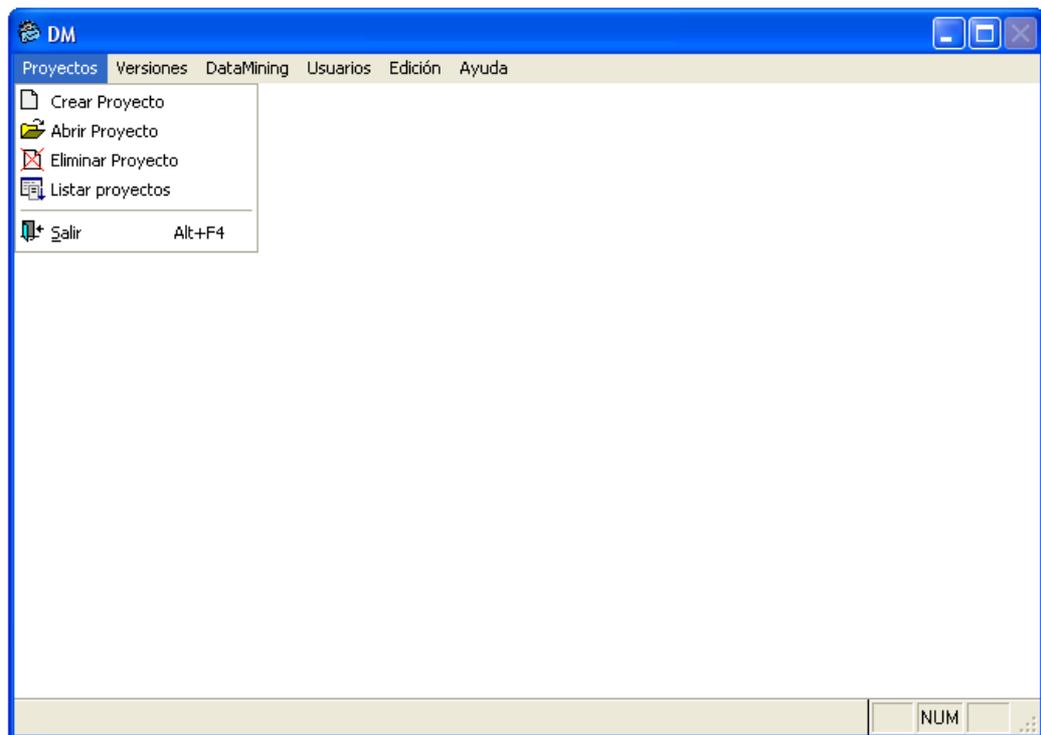


Figura ASI 3.24: pantalla de la opción de menú de Proyectos

*Pantalla Menú principal, con el menú de “Versiones” activado*

A continuación, en la Figura ASI 3.25, se muestra el diseño de la pantalla de Menú Principal del sistema cuando se activa el menú de Versiones. El cual posee cuatro opciones principales: Crear, Abrir, Eliminar y Listar Versiones de los proyectos.

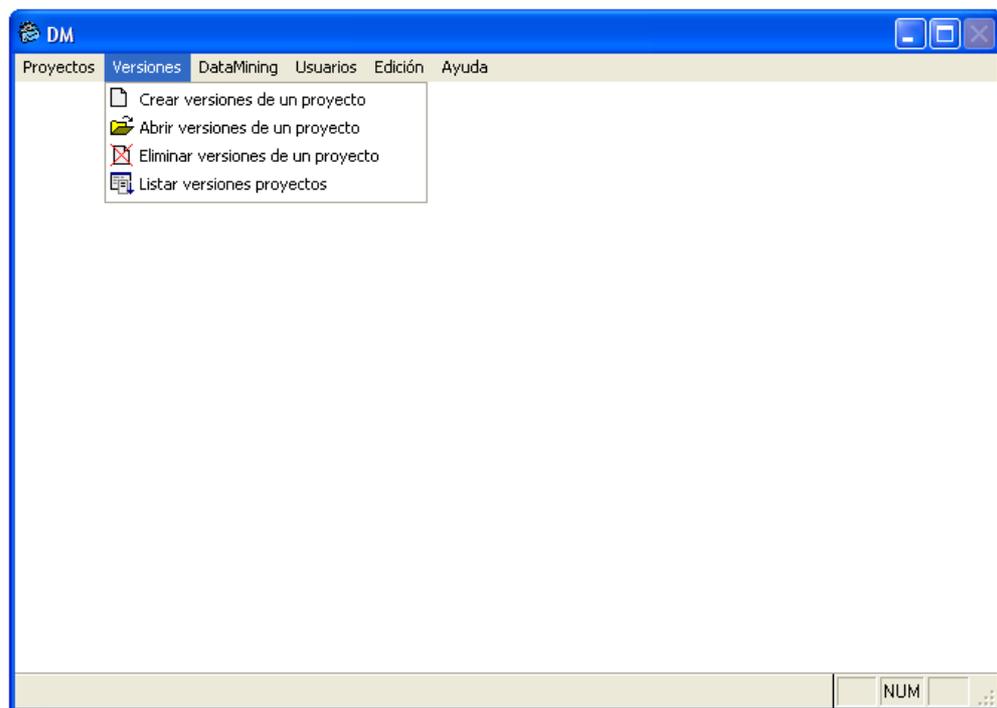


Figura ASI 3.25: pantalla de la opción de menú de Versiones

*Pantalla Menú principal, con el menú de “DataMining” activado*

A continuación, en la Figura ASI 3.26, se muestra el diseño de la pantalla de Menú Principal del sistema cuando se activa el menú de DataMining. El cual posee tres opciones principales: Parámetros de SOM, Parámetros de C5.0 y ejecutar DataMining. Cuando se quiere ejecutar la tercera opción los parámetros de los algoritmos están establecidos con valores por defecto, salvo que se hayan cambiado para la versión corriente.

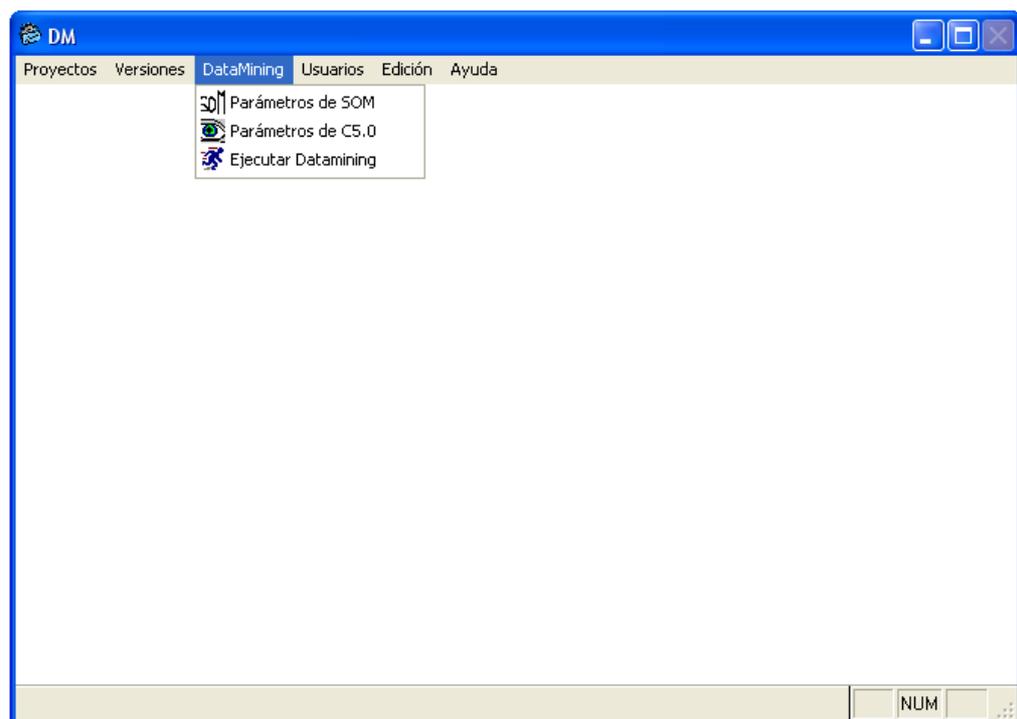


Figura ASI 3.26: pantalla de la opción de menú de DataMining

*Pantalla Menú principal, con el menú de “Usuarios” activado*

A continuación, en la Figura ASI 3.27, se muestra el diseño de la pantalla de Menú Principal del sistema cuando se activa el menú de Usuarios. El cual posee dos opciones principales: Agregar Usuario y Modificar Usuario. La primera opción solo se activa desde la pantalla de ingreso al sistema.

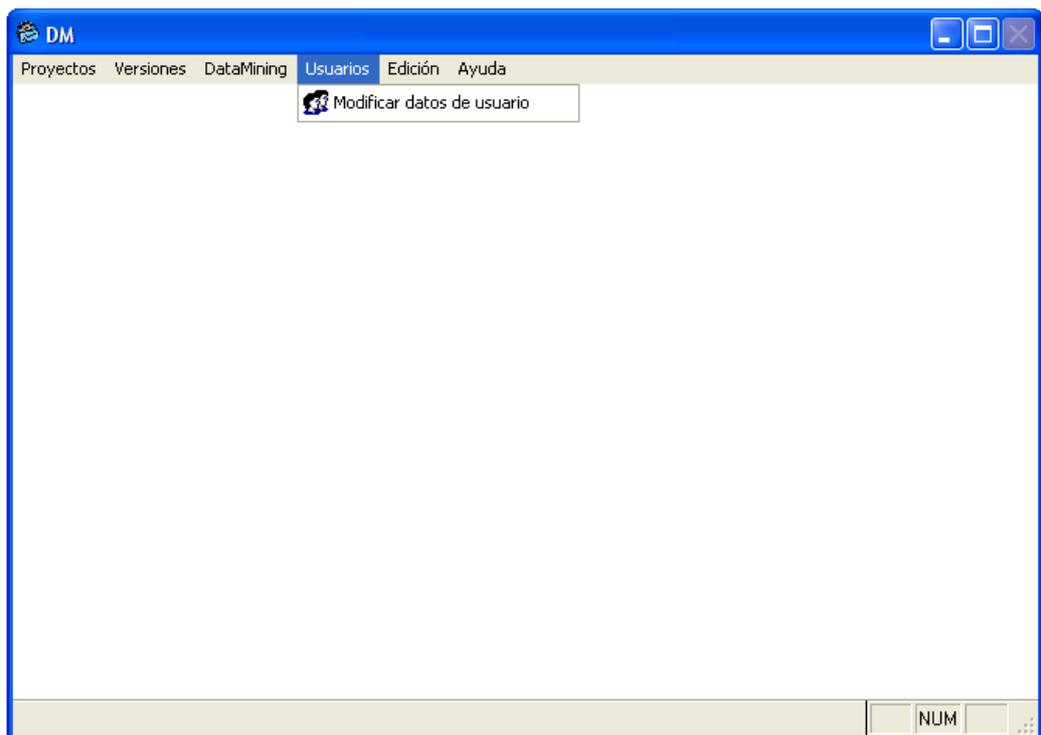


Figura ASI 3.27: pantalla de la opción de menú de Usuarios

*Pantalla Menú principal, con el menú de “Edición” activado*

A continuación, en la Figura ASI 3.28, se muestra el diseño de la pantalla de Menú Principal del sistema cuando se activa el menú de Edición. Estas opciones son estándares de Microsoft Windows y pueden ser de ayuda en la operación a los que operen con este sistema. Posee cinco opciones: Deshacer, Rehacer, Cortar, Copiar y Pegar.

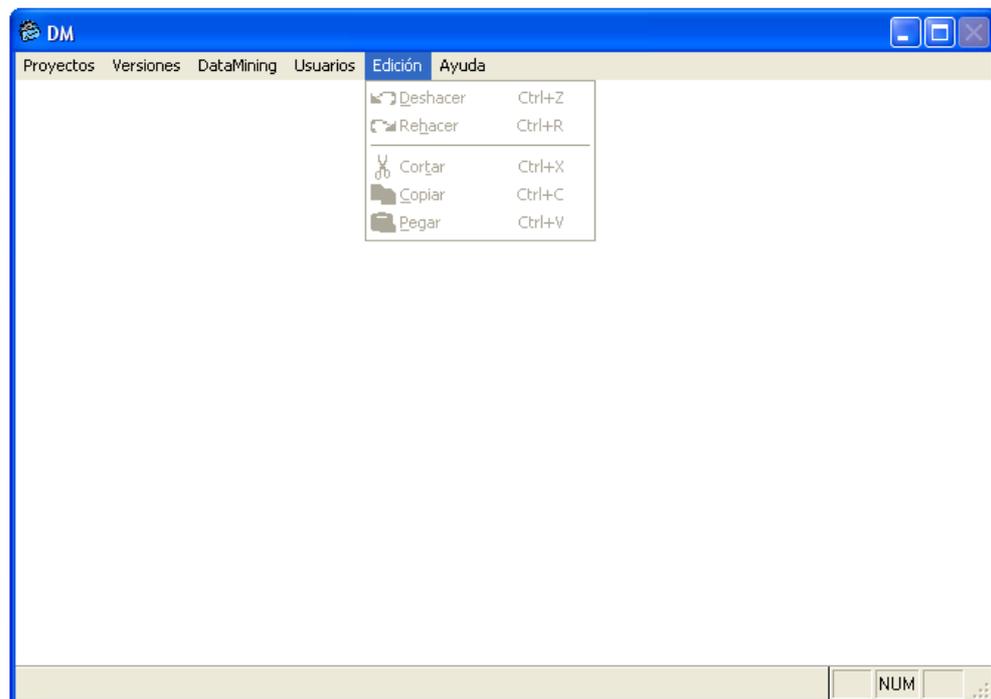


Figura ASI 3.28: pantalla de la opción de menú de Edición

*Pantalla Menú principal, con el menú de “Ayuda” activado*

A continuación, en la Figura ASI 3.29, se muestra el diseño de la pantalla de Menú Principal del sistema cuando se activa el menú de Ayuda. El cual posee dos opciones: Índice de Ayuda y Acerca de...

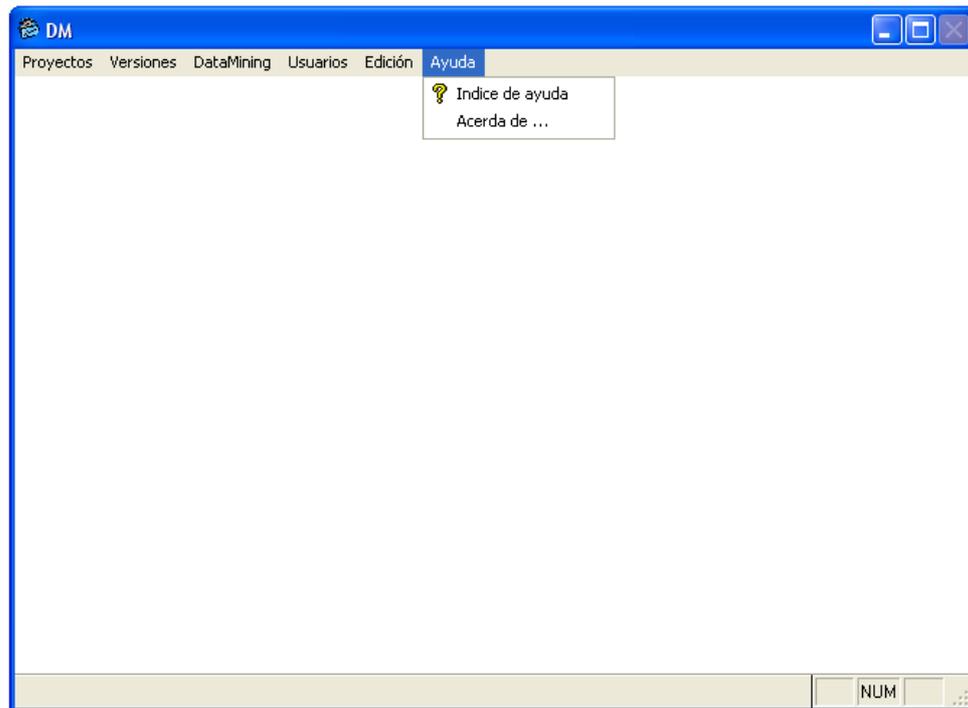


Figura ASI 3.29: pantalla de la opción de menú de Versiones

### *Pantalla Crear Proyecto Nuevo*

A continuación, en la Figura ASI 3.30, se muestra el diseño de la pantalla de Crear Proyecto Nuevo, en la cual se ingresa el nombre del proyecto a crear.

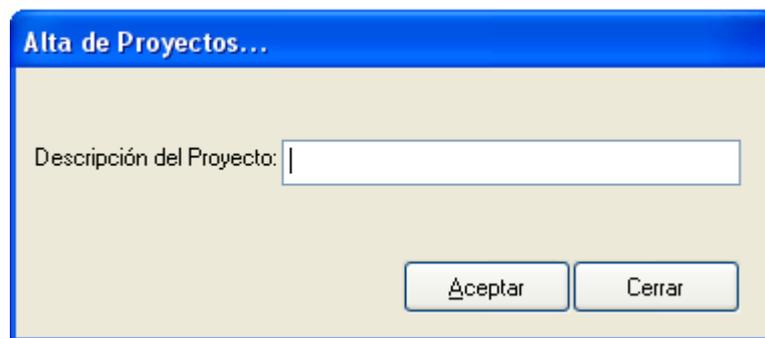


Figura ASI 3.30: pantalla de Crear Proyecto Nuevo

### *Pantalla Eliminar Proyectos*

A continuación, en la Figura ASI 3.31, se muestra el diseño de la pantalla de Eliminar Proyectos, en la cual se elige de una lista desplegable el proyecto que se desea borrar.

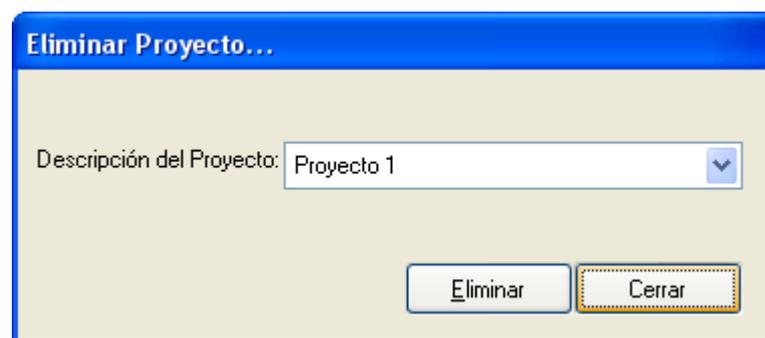


Figura ASI 3.31: pantalla de Eliminar Proyectos

### *Pantalla Listar Proyectos*

A continuación, en la Figura ASI 3.32, se muestra el diseño de la pantalla Listar Proyectos, en la cual se despliegan todos los proyectos pertenecientes al analista.

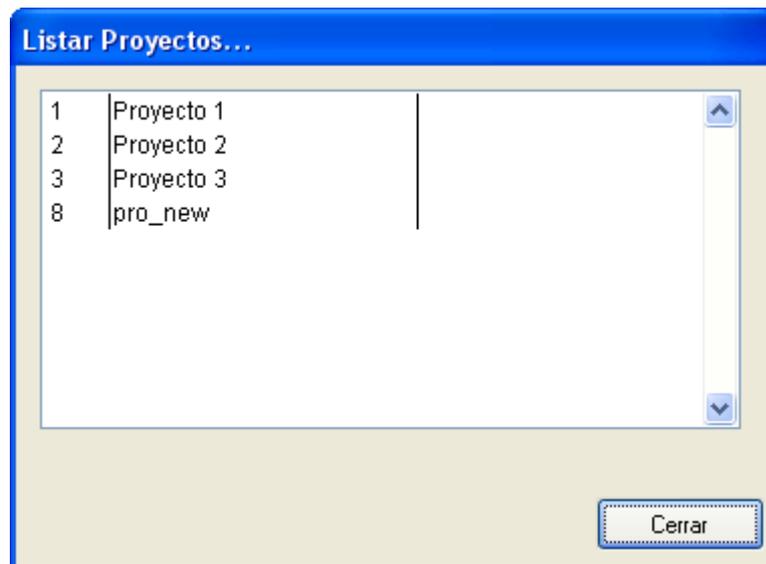


Figura ASI 3.32: pantalla Listar Proyectos

### *Pantalla Crear versiones de un proyecto*

A continuación, en la Figura ASI 3.33, se muestra el diseño de la pantalla de Crear Versión de un Proyecto, en la cual se elige de un desplegable el Proyecto al cual se le crea una versión.

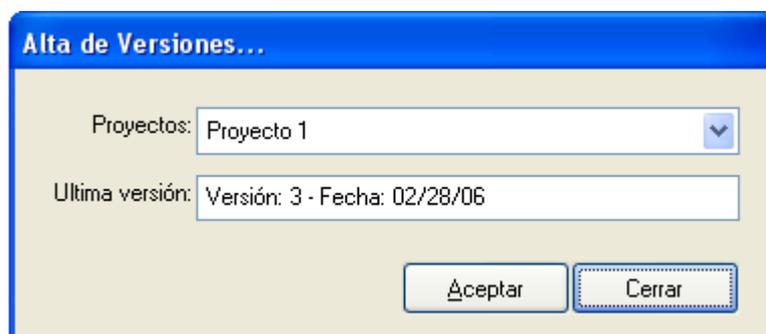


Figura ASI 3.33: pantalla de Crear Versión de un Proyecto

*Pantalla Abrir versiones de un proyecto*

A continuación, en la Figura ASI 3.34 se muestra el diseño de la pantalla de Abrir Versiones, en la cual se elige de una lista desplegable la Versión de un Proyecto con la que se quiere interactuar.

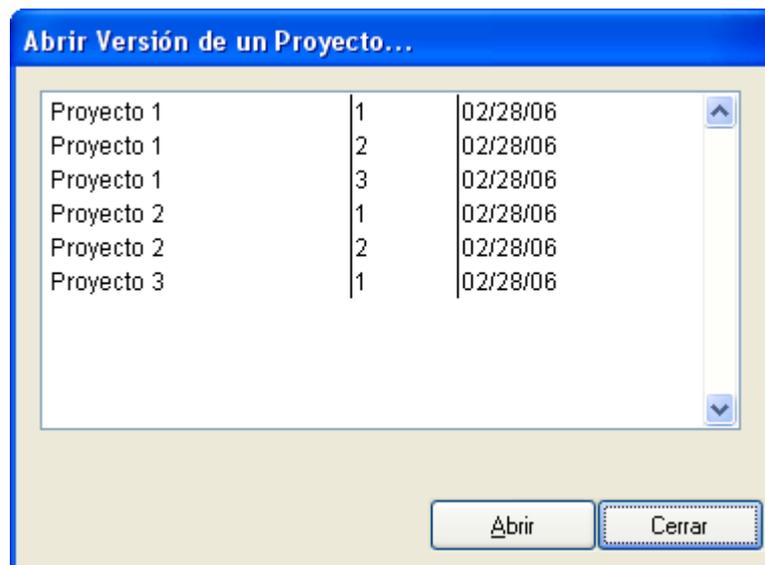


Figura ASI 3.34: pantalla de Abrir Versión de un Proyectos

*Pantalla Eliminar versión de un proyecto*

A continuación, en la Figura ASI 3.35, se muestra el diseño de la pantalla de Eliminar Versión de un Proyecto, en la cual se elige de una lista desplegable el proyecto y la versión que se desea borrar.

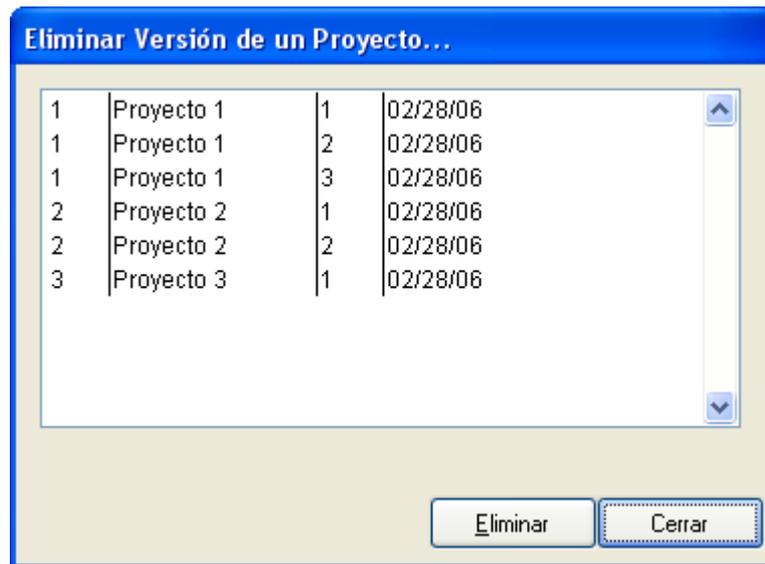


Figura ASI 3.35: pantalla de Eliminar Versión de un Proyecto

*Pantalla Listar versiones de proyectos*

A continuación, en la Figura ASI 3.36, se muestra el diseño de la pantalla Listar Versiones de Proyectos, en la cual se despliegan todas las versiones de los distintos proyectos que tiene a cargo el analista.

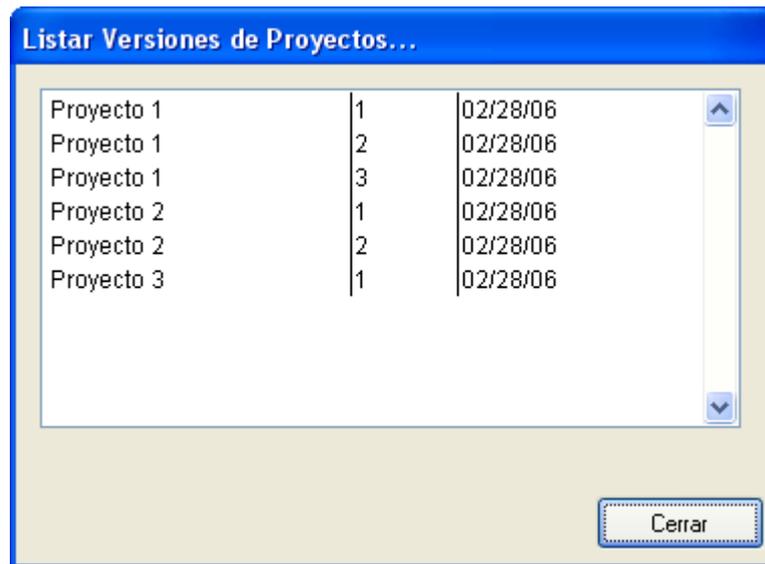


Figura ASI 3.36: pantalla Listar Versiones de Proyectos

*Pantalla Parámetros de SOM*

A continuación, en la Figura ASI 3.37, se muestra el diseño de la pantalla Parámetros SOM, en la cual se establecen las pautas para la ejecución del algoritmo SOM.

The image shows a software dialog box titled "Algoritmo SOM". It contains several input fields and two sub-sections for learning parameters. The main parameters are:

- Número de observaciones: 2996953
- Número de Variables: 5
- Neuronas en el mapa: 10
- Número de ciclos de entrenamiento: 4

There are two sub-sections for learning parameters:

- Parámetro de aprendizaje:**
  - Valor inicial: 0.9
  - Valor final: 0.1
  - Decay: Exponencial
- Sigma para un vecindario Gaussiano como % de ancho del mapa:**
  - Valor inicial: 0.5
  - Valor final: 0.01
  - Decay: Exponencial

An "Aceptar" button is located at the bottom right of the dialog box.

Figura ASI 3.37: pantalla Parámetros SOM

*Pantalla Parámetros de C5.0*

A continuación, en la Figura ASI 3.38, se muestra el diseño de la pantalla Parámetros C5.0, en la cual se establecen las pautas para la ejecución del algoritmo C5.0.

**Algoritmo C5.0**

Limpieza de atributos

Reglas

Ordenar por utilidad  bandas

Boost  ejemplos

Subsets of values

Usar ejemplos  %

Lock sample

Validación cruzada  hojas

Ignorar archivo de costos

Opciones avanzadas

Fuzzy thresholds

Poda global

Poda CF  %

Minimo  casos

Aceptar

Figura ASI 3.38: pantalla Parámetros C5.0

### *Pantalla Ejecutar Datamining*

A continuación, en la Figura ASI 3.39, se muestra el diseño de la pantalla Ejecutar DataMining, en la cual se seleccionan los algoritmos a utilizar en el proceso de exploración de datos.

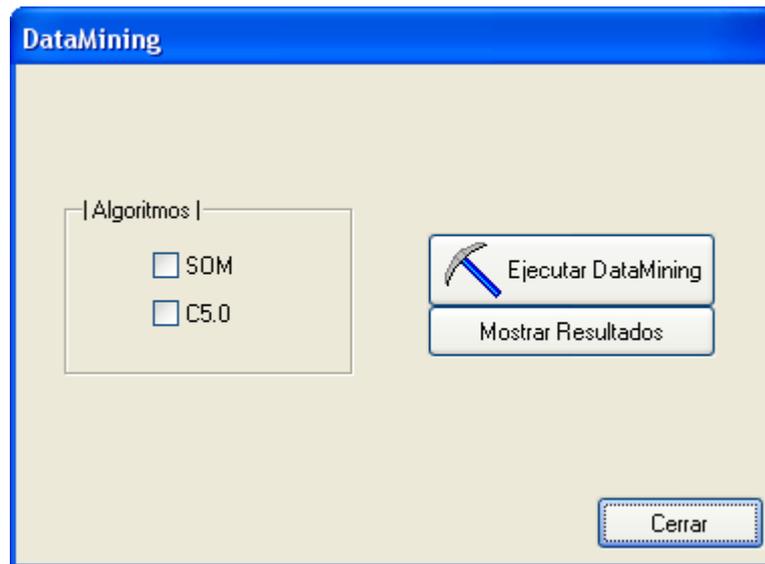


Figura ASI 3.39: pantalla Ejecutar DataMining

### *Pantalla Modificar / ingresar datos de usuario*

A continuación, en la Figura ASI 3.40, se muestra el diseño de la pantalla de Detalles de Usuario. En ella se cargan los datos de un nuevo usuario en el sistema o bien se modifican los datos de los previamente dados de alta (excepto el número de DNI ya que es la clave principal). Están señalados los campos que son obligatorios que estén cargados.

The screenshot shows a window titled "Detalles de usuarios...". It contains several input fields with red asterisks indicating they are mandatory. The fields are: DNI (with a vertical bar), Apellido y Nombre, Clave, Repetir Clave, Domicilio, Localidad, CP, Provincia (a dropdown menu showing "Ciudad de Buenos Aires - C"), Teléfono, País (a dropdown menu), e-mail, and Empresa. At the bottom right, there are two buttons: "Modificar" and "Cerrar". At the bottom left, there is a red text label: "(\*) Campos obligatorios".

Figura ASI 3.40: pantalla de Modificación / Ingreso de datos de usuario

### **3.5.4. Especificación de Formatos de Impresión**

El objetivo de esta tarea es especificar los formatos y características de las salidas o entradas impresas del sistema.

De acuerdo a los estándares establecidos en la tarea Especificación de Principios Generales de la Interfaz, se definen los formatos individuales de informes y formularios, estos últimos si son necesarios, así como sus características principales, entre las que se especifican la periodicidad, confidencialidad, procedimientos de entrega o difusión, y salvaguarda de copia.

Opcionalmente, se recomienda la utilización de prototipos.

#### **3.5.4.1. Formatos de Impresión**

A continuación se detallan los diseños de impresión asociados a cada una de las pantallas de consulta indicadas en la sección Definición de las Pantallas del Sistema.

##### *Reporte de Usuarios:*

A continuación, en la Figura ASI 3.41, se muestra el diseño del formato del reporte asociado a la pantalla de Consulta de Usuarios.

*Sistema: DM*

**REPORTE DE USUARIOS**

08/05/2006

<b>DNI</b>	<b>Nombre y Apellido</b>	
<b>Domicilio</b>		<b>Cod. Postal</b>
<b>Localidad</b>	<b>Provincia</b>	<b>País</b>
<b>Telefono</b>	<b>e-mail</b>	
<b>Empresa</b>		

Página: 1

Figura ASI 3.41: reporte de usuario

## Reporte de Proyectos de un usuario

A continuación, en la Figura ASI 3.42, se muestra el diseño del formato del reporte asociado a la pantalla Listar Versiones de Proyectos.

*Sistema: DM*

**REPORTE DE VERSIONES DE PROYECTO POR USUARIO** 08/05/2006

DNI	Nombre y Apellido		
Id. Proyecto	Nombre de Proyecto	Fecha Creación	
Id. Versión	Fecha Creación		

Página: 1

Figura ASI 3.42: reporte de Proyectos de un usuario.

---

## Reporte de Versiones de un Proyecto de un usuario

A continuación, en la Figura ASI 3.43, se muestra el diseño del formato del reporte asociado a la pantalla Listar Versiones de Proyectos.

*Sistema: DM*

**REPORTE DE PROYECTOS POR USUARIO**

08/06/2006

DNI	Nombre y Apellido			
Id. Proyecto	Nombre de Proyecto		Fecha Creación	

Página: 1

Figura ASI 3.43: reporte de Versiones de un Proyecto de un usuario.

## Reporte de Detalles de una Versión de un Proyecto de un usuario

A continuación, en la Figura ASI 3.44, se muestra el diseño del formato del reporte asociado a la pantalla Listar Versiones de Proyectos.

<i>Sistema: DM</i>		
<b>REPORTE DE DETALLES DE VERSION DE PROYECTO</b>		
		08/06/2006
<b>DNI</b>	<b>Nombre y Apellido</b>	
<b>Id. Proyecto</b>	<b>Nombre de Proyecto</b>	<b>Fecha Creación</b>
<b>Id. Versión</b>	<b>Fecha Creación</b>	
<b>PC Id.</b>	<b>DSN</b>	
<b>Path</b>		
<b>Parámetro 001</b>	<b>Parámetro 004</b>	
<b>Parámetro 002</b>	<b>Parámetro 005</b>	
<b>Parámetro 003</b>		

Figura ASI 3.44: reporte de Detalles de una Versión de un Proyecto de un usuario

### **3.6. Análisis de Consistencia y Especificación de Requisitos**

El objetivo de esta actividad es garantizar la calidad de los distintos modelos generados en el proceso de Análisis del Sistema de Información, y asegurar que los usuarios y los Analistas tienen el mismo concepto del sistema. Para cumplir dicho objetivo, se llevan a cabo las siguientes acciones:

- Verificación de la calidad técnica de cada modelo.
- Aseguramiento de la coherencia entre los distintos modelos.
- Validación del cumplimiento de los requisitos.

Esta actividad requiere una herramienta de apoyo para realizar el análisis de consistencia. También se elabora en esta actividad la Especificación de Requisitos Software (ERS), como producto para la aprobación formal, por parte del usuario, de las especificaciones del sistema.

La Especificación de Requisitos Software se convierte en la línea base para los procesos posteriores del desarrollo del software, de modo que cualquier petición de cambio en los requisitos que pueda surgir posteriormente, debe ser evaluada y aprobada.

#### **3.6.1. Verificación de los modelos**

El objetivo de esta tarea es asegurar la calidad formal de los distintos modelos, conforme a la técnica seguida para la elaboración de cada producto y a las normas determinadas en el Catálogo de Normas.

#### **3.6.2. Verificación de Modelos**

Se verificó la calidad de los distintos modelos de forma separada con el propósito de garantizar su adecuación al problema a resolver y su seguimiento respecto de las técnicas de análisis seleccionadas.

### 3.6.3. Análisis de Consistencia entre métodos

El objetivo de esta tarea es asegurar que los modelos son coherentes entre sí, comprobando la falta de ambigüedades o duplicación de información.

Las diferentes comprobaciones varían en función del tipo de desarrollo, aunque, en general, son matrices entre los elementos comunes de los distintos modelos. Estas comprobaciones forman parte del producto Resultado de Análisis de Consistencia.

Los análisis de consistencia propuestos en Desarrollo Estructurado son, Comentario Paola Britos: Existe un modelado orientado a objetos, con solo un diagrama de ER de estructurado, sugiero unificar el modelado por ende la validación utilizando un solo estilo de modelado:

➤ Modelo Lógico de Datos Normalizado / Modelo de Procesos:

Se verifica que:

- Cada uno de los almacenes definidos en el modelo de procesos se corresponde con una parte del modelo lógico de datos normalizado. Es decir, un almacén se puede corresponder con una entidad, atributos de una entidad o con varias entidades relacionadas.
- Los atributos del modelo lógico de datos normalizado y del modelo de procesos se ajustan a una misma especificación.
- El modelo lógico de datos normalizado satisface las principales consultas de información. Para comprobar que el modelo lógico de datos normalizado puede soportar dichas consultas, se proponen, como técnicas opcionales, la determinación de caminos de acceso lógico en consultas y el cálculo de accesos lógicos.
- Todas y cada una de las entidades del modelo lógico normalizado son accedidas por algún proceso primitivo. Para dicha comprobación, se propone una matriz de entidades/procesos, donde se especifique que tipo

de acceso se realiza (alta, baja, modificación o consulta).

➤ Modelo Lógico de Datos Normalizado / Interfaz de Usuario:

- En este análisis se comprueba que los atributos relevantes que aparecen en cada diálogo de la interfaz de usuario forman parte del modelo lógico de datos normalizado o, en su caso, atributos derivados de los mismos.

➤ Modelo de Procesos / Interfaz de Usuario:

- Se comprueba que todo proceso en línea tiene asociado al menos un diálogo.

El resultado del análisis de consistencia en un análisis estructurado es un producto que engloba los siguientes elementos:

- Matriz de almacenes de datos / entidades del modelo lógico de datos normalizado.
- Matriz de atributos de interfaz / atributos de entidades del modelo lógico de datos normalizado.
- Caminos de acceso lógico en consultas.
- Cálculo de accesos lógicos.
- Matriz de entidades / procesos.
- Matriz de diálogos / procesos.

Los análisis de consistencia propuestos en Desarrollo Orientado a Objetos son los siguientes:

Considerando que la interfaz de usuario incluye diagramas dinámicos y forma parte del modelo de clases, los análisis de consistencia con la interfaz pueden solaparse con los del resto de los modelos. Los análisis de consistencia propuestos son:

➤ Modelo de Clases / Diagramas Dinámicos:

Se comprueba que:

- Cada mensaje entre objetos se corresponde con una operación de una clase y que todos los mensajes se envían a las clases correctas.

- La clase que recibe un mensaje con petición de datos tiene capacidad para proporcionar esos datos.

- Cada objeto del diagrama de interacción de objetos tiene una correspondencia en el modelo de clases.

➤ En el caso de haber elaborado diagramas de transición de estados para clases significativas:

- Se verifica que, para cada uno de ellos, todo evento se corresponde con una operación de la clase. También se tiene que establecer si las acciones y actividades de los diagramas de transición de estado se corresponden con operaciones de la clase.

➤ Modelo de clases / Interfaz de usuario

- Cada clase que requiera una clase de interfaz de usuario, debe tener asociación con ella en el modelo de clases.

- Todas las clases, atributos y operaciones identificados en la interfaz de usuario, deben tener su correspondencia con algún atributo, operación o clase en el modelo de clases.

➤ Análisis de la Realización de los Casos de Uso / Interfaz de Usuario

- Cada elemento que active la navegación entre pantallas, debe estar asociado con un mensaje del diagrama de interacción de objetos.

Además, se revisa que los subsistemas satisfagan la realización de todos los casos de uso, e incluyan las clases identificadas hasta el momento.

El resultado del análisis de consistencia en un análisis orientado a objetos es un producto que engloba los siguientes elementos:

- Matriz de mensajes del diagrama de interacción de objetos / operaciones del modelo de clases.
- Matriz de mensajes del diagrama de interacción de objetos / operaciones y atributos del modelo de clases.
- Matriz de objetos del diagrama de interacción de objetos / clases, atributos del modelo de clases.
- Matriz (evento, acción, actividad de clase) / operaciones de clase.
- Correspondencia elementos de negocio de interfaz de usuario / modelo de clases.
- Correspondencia entre elementos de navegación de interfaz de usuario / mensajes del diagrama de interacción de objetos.

#### **3.6.3.1. Análisis de Consistencia**

Se comprueba la coherencia entre los distintos modelos de acuerdo a las trazabilidades que se presentan en la Figura ASI 3.45.

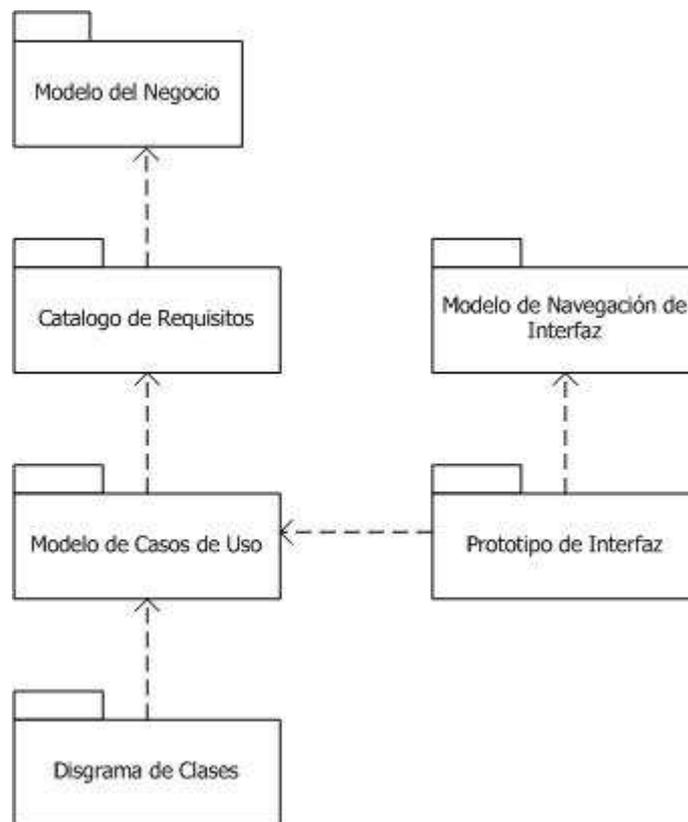


Figura ASI 3.45: Trazabilidades entre los distintos modelos

La comprobación se lleva a cabo mediante un conjunto de matrices de trazabilidad.

#### Catalogo de Requisitos – Modelo de Negocio

La Matriz de la tabla ASI 3.16 muestra en las filas los requisitos del catálogo de requisitos y en la columnas las actividades del modelo de negocio. Como puede verse en la misma, cada una de las actividades tiene su correspondencia con algún requisito. Existen además algunos requisitos que no tienen correspondencia con actividades del negocio. Estos requisitos fueron agregados como consecuencia de la interacción con el usuario y cubren aspectos de funcionamiento operativo.

	Gestionar Proyecto	Gestionar Versión	Ejecutar DM
RF 1 - Ingreso al sistema			
RF 2 - Agregar Usuario			
RF 3 - Modificar Usuario			
RF 4 - Crear Proyecto	<b>X</b>		
RF 5 - Eliminar Proyecto	<b>X</b>		
RF 6 - Listar Proyectos	<b>X</b>		
RF 7 - Salir del Sistema			
RF 8 - Crear Versiones de un Proyecto		<b>X</b>	
RF 9 - Abrir Versión		<b>X</b>	
RF 10 - Eliminar Versión		<b>X</b>	
RF 11 - Listar Versiones		<b>X</b>	
RF 12 - Establecer Parámetros de SOM			<b>X</b>
RF 13 - Establecer Parámetros de C5.0			<b>X</b>
RF 14 - Ejecutar Proceso de Explotación de Datos			<b>X</b>

Tabla ASI 3.16: Trazabilidades entre el Catalogo de Requisitos y Modelo de Negocio

### Modelo de Casos de Uso – Catalogo de Requisitos

La Matriz de la tabla ASI 3.17 muestra en las filas los requisitos del catálogo de requisitos y en las columnas los casos de uso del Modelo de Casos de Uso. Como puede verse en la misma cada requisito tiene correspondencia con algún caso de uso y viceversa.

	Ingreso al sistema	Agregar Usuario	Modificar Usuario	Crear Proyecto	Eliminar Proyecto	Listar Proyectos	Salir del Sistema	Crear Versiones de un Proyecto	Abrir Versión	Eliminar Versión	Listar Versiones	Establecer Parámetros de SOM	Establecer Parámetros de C5.0	Ejecutar Proceso de Explotación de Datos
RF 1 - Ingreso al sistema	<b>X</b>													
RF 2 - Agregar Usuario		<b>X</b>												
RF 3 - Modificar Usuario			<b>X</b>											
RF 4 - Crear Proyecto				<b>X</b>										
RF 5 - Eliminar Proyecto					<b>X</b>									
RF 6 - Listar Proyectos						<b>X</b>								
RF 7 - Salir del Sistema							<b>X</b>							
RF 8 - Crear Versiones de un Proyecto								<b>X</b>						

	Ingreso al sistema	Agregar Usuario	Modificar Usuario	Crear Proyecto	Eliminar Proyecto	Listar Proyectos	Salir del Sistema	Crear Versiones de un Proyecto	Abrir Versión	Eliminar Versión	Listar Versiones	Establecer Parámetros de SOM	Establecer Parámetros de C5.0	Ejecutar Proceso de Explotación de Datos
RF 9 - Abrir Versión									<b>X</b>					
RF 10 - Eliminar Versión										<b>X</b>				
RF 11 - Listar Versiones											<b>X</b>			
RF 12 - Establecer Parámetros de SOM												<b>X</b>		
RF 13 - Establecer Parámetros de C5.0													<b>X</b>	
RF 14 - Ejecutar Proceso de Explotación de Datos														<b>X</b>

Tabla ASI 3.17: Trazabilidades entre el Modelo de Casos de Uso y el Catalogo de Requisitos

### Modelo de Casos de Uso – Prototipo de Interfaz

La Matriz de la tabla ASI 3.18 muestra en las filas las pantallas del modelo del Prototipo de Interfaz y en las columnas los casos de uso del Modelo de Casos de Uso. Como puede verse en la misma cada pantalla tiene correspondencia con algún caso de uso y viceversa.

	Ingresar al sistema	Crear Proyecto	Eliminar Proyectos	Listar Proyectos	Crear versiones	Abrir versiones	Eliminar versión	Listar versiones	Parámetros de SOM	Parámetros de C5.0	Ejecutar Datamini	Ingresar datos de	Modificar datos de
Ingreso al sistema	<b>X</b>												
Menú Principal													
Proyectos													
Crear Proyecto		<b>X</b>											
Eliminar Proyecto			<b>X</b>										
Listar Proyectos				<b>X</b>									
Versiones													
Crear Versiones de un Proyecto					<b>X</b>								
Abrir Versión						<b>X</b>							
Eliminar Versión							<b>X</b>						
Listar Versiones								<b>X</b>					
DataMining													
Establecer Parámetros de SOM									<b>X</b>				
Establecer										<b>X</b>			

	Ingresar al sistema	Crear Proyecto	Eliminar Proyectos	Listar Proyectos	Crear versiones	Abrir versiones	Eliminar versión	Listar versiones	Parámetros de SOM	Parámetros de C5.0	Ejecutar Datamini	Ingresar datos.de	Modificar datos.de
Parámetros de C5.0													
Ejecutar Proceso de Explotación de Datos											X		
Usuarios													
Agregar Usuario												X	
Modificar Usuario													X
Salir del Sistema													
Ayuda													

Tabla ASI 3.18: Trazabilidades entre el Modelo de Casos de Uso y el Prototipo de Interfaz

### Modelo de Modelo de Navegación de Interfaz – Prototipo de Interfaz

La Matriz de la tabla ASI 3.19 muestra en las filas las pantallas del modelo del Prototipo de Interfaz y en las columnas las pantallas del Modelo de Navegación. Como puede verse en la misma cada pantalla de la interfaz de usuario tiene correspondencia con alguna de las pantallas del modelo de navegación.

	Ingreso al sistema	Menú Principal	Proyectos	Crear Proyecto
Ingreso al sistema	X			
Menú Principal		X		
Proyectos			X	
Crear Proyecto				X
Eliminar Proyecto				
Listar Proyectos				
Versiones				
Crear Versiones de un Proyecto				
Abrir Versión				
Eliminar Versión				
Listar Versiones				
DataMining				
Establecer Parámetros de SOM				
Establecer Parámetros de C5.0				
Ejecutar Proceso de Explotación de Datos				
Usuarios				
Agregar Usuario				
Modificar Usuario				
Salir del Sistema				
Ayuda				

Abrir Versión	Crear Versiones de un Proyecto	Versiones	Listar Proyectos	Eliminar Proyecto	
					Ingreso al sistema
					Menú Principal
					Proyectos
					Crear Proyecto
				X	Eliminar Proyecto
			X		Listar Proyectos
		X			Versiones
	X				Crear Versiones de un Proyecto
X					Abrir Versión
					Eliminar Versión
					Listar Versiones
					DataMining
					Establecer Parámetros de SOM
					Establecer Parámetros de C5.0
					Ejecutar Proceso de Explotación de Datos
					Usuarios
					Agregar Usuario
					Modificar Usuario
					Salir del Sistema
					Ayuda

Establecer Parámetros de SOM	DataMining	Listar Versiones	Eliminar Versión	
				Ingreso al sistema
				Menú Principal
				Proyectos
				Crear Proyecto
				Eliminar Proyecto
				Listar Proyectos
				Versiones
				Crear Versiones de un Proyecto
				Abrir Versión
			X	Eliminar Versión
		X		Listar Versiones
	X			DataMining
X				Establecer Parámetros de SOM
				Establecer Parámetros de C5.0
				Ejecutar Proceso de Explotación de Datos
				Usuarios
				Agregar Usuario
				Modificar Usuario
				Salir del Sistema
				Ayuda

Usuarios	Ejecutar Proceso de Explotación de Datos	Establecer Parámetros de C5.0	
			Ingreso al sistema
			Menú Principal
			Proyectos
			Crear Proyecto
			Eliminar Proyecto
			Listar Proyectos
			Versiones
			Crear Versiones de un Proyecto
			Abrir Versión
			Eliminar Versión
			Listar Versiones
			DataMining
			Establecer Parámetros de SOM
		X	Establecer Parámetros de C5.0
	X		Ejecutar Proceso de Explotación de Datos
X			Usuarios
			Agregar Usuario
			Modificar Usuario
			Salir del Sistema
			Ayuda

Ayuda	Salir del Sistema	Modificar Usuario	Agregar Usuario	
				Ingreso al sistema
				Menú Principal
				Proyectos
				Crear Proyecto
				Eliminar Proyecto
				Listar Proyectos
				Versiones
				Crear Versiones de un Proyecto
				Abrir Versión
				Eliminar Versión
				Listar Versiones
				DataMining
				Establecer Parámetros de SOM
				Establecer Parámetros de C5.0
				Ejecutar Proceso de Explotación de Datos
				Usuarios
			X	Agregar Usuario
		X		Modificar Usuario
	X			Salir del Sistema
X				Ayuda

Tabla ASI 3.19: Trazabilidades entre el Modelo de Navegación de Interfaz y el Prototipo de Interfaz

### Modelo de Casos de Uso – Clases

La Matriz de la tabla ASI 3.20 muestra en las filas los casos de uso del Modelo de Casos de Uso y en las columnas las Clases. Como puede verse en la misma, cada clase tiene correspondencia con algún caso de uso y viceversa.

	Proyecto	Versión	Usuario	C5	SOM
Ingreso al sistema			X		
Agregar Usuario			X		
Modificar Usuario			X		
Crear Proyecto	X				
Eliminar Proyecto	X				
Listar Proyectos	X				
Salir del Sistema					
Crear Versiones de un Proyecto		X			
Abrir Versión		X			
Eliminar Versión		X			
Listar Versiones		X			
Establecer Parámetros de SOM				X	
Establecer Parámetros de C5.0					X
Ejecutar Proceso de Explotación de Datos				X	X

Tabla ASI 3.20: Trazabilidades entre el Diagrama de Clases y el Modelo de Casos de Uso

## **3.7. Aprobación del Análisis del Sistema de Información**

### **3.7.1. Presentación y Aprobación del Análisis del Sistema de Información**

En esta tarea se realiza la presentación del análisis del sistema de información al Comité de Dirección, para la aprobación final del mismo.

En una reunión mantenida entre el tesista y la Directora del proyecto se dio por aprobada la fase de Análisis del Sistema de Información.

## CAPÍTULO 4 – DISEÑO DEL SISTEMA DE INFORMACIÓN

El objetivo del proceso de Diseño del Sistema de Información (DSI) es la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información.

A partir de dicha información, se generan todas las especificaciones de construcción relativas al propio sistema, así como la descripción técnica del plan de pruebas, la definición de los requisitos de implantación y el diseño de los procedimientos de migración y carga inicial, éstos últimos cuando proceda.

Al ser MÉTRICA Versión 3 una metodología que cubre tanto desarrollos estructurados como orientados a objetos, las actividades de ambas aproximaciones están integradas en una estructura común.

Las actividades de este proceso se agrupan en dos grandes bloques.

- En un primer bloque de actividades, que se llevan a cabo en paralelo, se obtiene el diseño de detalle del sistema de información. La realización de estas actividades exige una continua realimentación. En general, el orden real de ejecución de las mismas depende de las particularidades del sistema de información y, por lo tanto, de generación de sus productos.

En la actividad Definición de la Arquitectura del Sistema (DSI 1), se establece el particionamiento físico del sistema de información, así como su organización en subsistemas de diseño, la especificación del entorno tecnológico, y sus requisitos de operación, administración, seguridad y control de acceso. Se completan los catálogos de requisitos y normas, en función de la definición del entorno tecnológico, con aquellos aspectos relativos al diseño y construcción que sea necesario contemplar. Asimismo, se crea un catálogo de excepciones del sistema, en el que se registran las situaciones de funcionamiento secundario o anómalo que se estime oportuno considerar y, por lo tanto, diseñar y probar. Este catálogo de excepciones se

utiliza como referencia en la especificación técnica de las pruebas del sistema.

El particionamiento físico del sistema de información permite organizar un diseño que contemple un sistema de información distribuido, como por ejemplo la arquitectura cliente/servidor, siendo aplicable a arquitecturas multinivel en general. Independientemente de la infraestructura tecnológica, dicho particionamiento representa los distintos niveles funcionales o físicos del sistema de información. La relación entre los elementos del diseño y particionamiento físico, y a su vez, entre el particionamiento físico y el entorno tecnológico, permite una especificación de la distribución de los elementos del sistema de información y, al mismo tiempo, un diseño orientado a la movilidad a otras plataformas o la reubicación de subsistemas.

El sistema de información se estructura en subsistemas de diseño. Éstos a su vez se clasifican como de soporte o específicos, al responder a propósitos diferentes.

- Los subsistemas de soporte contienen los elementos o servicios comunes al sistema y a la instalación, y generalmente están originados por la interacción con la infraestructura técnica o la reutilización de otros sistemas, con un nivel de complejidad técnica mayor.
- Los subsistemas específicos contienen los elementos propios del sistema de información, generalmente con una continuidad de los subsistemas definidos en el proceso de Análisis del Sistema de Información (ASI).

También se especifica en detalle el entorno tecnológico del sistema de información, junto con su planificación de capacidades (capacity planning), y sus requisitos de operación, administración, seguridad y control de acceso.

El diseño detallado del sistema de información, siguiendo un enfoque

estructurado, comprende un conjunto de actividades que se llevan a cabo en paralelo a la Definición de la Arquitectura del Sistema (DSI 1). El alcance de cada una de estas actividades se resume a continuación:

- Diseño de la Arquitectura de Soporte (DSI 2), que incluye el diseño detallado de los subsistemas de soporte, el establecimiento de las normas y requisitos propios del diseño y construcción, así como la identificación y definición de los mecanismos genéricos de diseño y construcción.
- Diseño de la Arquitectura de Módulos del Sistema (DSI 5), dónde se realiza el diseño de detalle de los subsistemas específicos del sistema de información y la revisión de la interfaz de usuario.
- Diseño Físico de Datos (DSI 6), que incluye el diseño y optimización de las estructuras de datos del sistema, así como su localización en los nodos de la arquitectura propuesta.

En el caso de Diseño Orientado a Objetos, conviene señalar que el diseño de la persistencia de los objetos se lleva a cabo sobre bases de datos relacionales, y que el diseño detallado del sistema de información se realiza en paralelo con la actividad de Diseño de la Arquitectura de Soporte (DSI 2), y se corresponde con las siguientes actividades:

- Diseño de Casos de Uso Reales (DSI 3), con el diseño detallado del comportamiento del sistema de información para los casos de uso, el diseño de la interfaz de usuario y la validación de la división en subsistemas.
- Diseño de Clases (DSI 4), con el diseño detallado de cada una de las clases que forman parte del sistema, sus atributos, operaciones, relaciones y métodos, y la estructura jerárquica del mismo. En el caso de que sea necesario, se realiza la definición de un plan de migración y carga inicial de datos.

Una vez que se tiene el modelo de clases, se comienza el diseño físico en la actividad Diseño Físico de Datos (DSI 6), común con el enfoque estructurado.

Una vez finalizado el diseño de detalle, se realiza su revisión y validación en la actividad Verificación y Aceptación de la Arquitectura del Sistema (DSI 7), con el objeto de analizar la consistencia entre los distintos modelos y conseguir la aceptación del diseño por parte de los responsables de las áreas de Explotación y Sistemas.

- El segundo bloque de actividades complementa el diseño del sistema de información. En él se generan todas las especificaciones necesarias para la construcción del sistema de información:
  - Generación de Especificaciones de Construcción (DSI 8), fijando las directrices para la construcción de los componentes del sistema, así como de las estructuras de datos.
  - Diseño de la Migración y Carga Inicial de Datos (DSI 9), en el que se definen los procedimientos de migración y sus componentes asociados, con las especificaciones de construcción oportunas.
  - Especificación Técnica del Plan de Pruebas (DSI 10), que incluye la definición y revisión del plan de pruebas, y el diseño de las verificaciones de los niveles de prueba establecidos. El catálogo de excepciones permite, de una forma muy ágil, establecer un conjunto de verificaciones relacionadas con el propio diseño o con la arquitectura del sistema.
  - Establecimiento de Requisitos de Implantación (DSI 11), que hace posible concretar las exigencias relacionados con la propia implantación del sistema, tales como formación de usuarios finales, infraestructura, etc.

Finalmente, en la actividad de Presentación y Aprobación del Diseño del Sistema de Información (DSI 12), se realiza una presentación formal y aprobación de los distintos productos del diseño.

#### **4.1. Definición de la Arquitectura del Sistema**

En esta actividad se define la arquitectura general del sistema de información, especificando las distintas particiones físicas del mismo, la descomposición lógica en subsistemas de diseño y la ubicación de cada subsistema en cada partición, así como la especificación detallada de la infraestructura tecnológica necesaria para dar soporte al sistema de información.

El particionamiento físico del sistema de información se especifica identificando los nodos y las comunicaciones entre los mismos, con cierta independencia de la infraestructura tecnológica que da soporte a cada nodo.

Con el fin de organizar y facilitar el diseño, se realiza una división del sistema de información en subsistemas de diseño, como partes lógicas coherentes y con interfaces claramente definidas. Para esto, se establece una distinción entre subsistemas específicos del sistema de información (en adelante, subsistemas específicos) y subsistemas de soporte, con la finalidad de independizar, en la medida de lo posible, las funcionalidades a cubrir por el sistema de información de la infraestructura que le da soporte. En la mayoría de los casos, los subsistemas específicos provienen directamente de las especificaciones de análisis y de los subsistemas de análisis, mientras que los subsistemas de soporte provienen de la necesidad de interacción del sistema de información con la infraestructura y con el resto de los sistemas, así como de la reutilización de módulos o subsistemas ya existentes en la instalación.

Debido a que la definición de los subsistemas de soporte puede exigir la participación de distintos perfiles técnicos, se propone el diseño de ambos tipos de subsistemas en actividades distintas, aunque en paralelo. Una vez identificados y definidos los distintos subsistemas de diseño, se determina su ubicación óptima de acuerdo a la arquitectura propuesta. La asignación de dichos subsistemas a cada nodo permite disponer, en función de la carga de proceso y comunicación existente entre los

nodos, de la información necesaria para realizar una estimación de las necesidades de infraestructura tecnológica que da soporte al sistema de información. Este factor es especialmente crítico en arquitecturas multinivel o cliente/servidor, donde las comunicaciones son determinantes en el rendimiento final del sistema.

Se propone crear un catálogo de excepciones en el que se especifiquen las situaciones anómalas o secundarias en el funcionamiento y ejecución del sistema de información, y que se irá completando a medida que se avance en el diseño detallado de los subsistemas. Para esto, en esta actividad también se establecen los requisitos, normas y estándares originados como consecuencia de la adopción de una determinada solución de arquitectura o infraestructura, que serán aplicables tanto en este proceso como en la Construcción del Sistema de Información (CSI). Se detallan a su vez, de acuerdo a las particularidades de la arquitectura del sistema propuesta, los requisitos de operación, seguridad y control, especificando los procedimientos necesarios para su cumplimiento.

Como resultado final de esta actividad, se actualizan los catálogos de requisitos y normas, y se generan los siguientes productos:

- Diseño de la Arquitectura del Sistema, como producto que engloba el particionamiento físico del sistema de información y la descripción de subsistemas de diseño.
- Entorno Tecnológico del Sistema, que a su vez comprende la especificación del entorno tecnológico, las restricciones técnicas y la planificación de capacidades.
- Catálogo de Excepciones.
- Procedimientos de Operación y Administración del Sistema.
- Procedimientos de Seguridad y Control de Acceso.

### **4.1.1. Definición de Niveles de Arquitectura**

En esta tarea se definen los niveles de arquitectura software, mediante la definición de los principales participaciones físicas del sistema de información, representadas como nodos y comunicaciones entre nodos.

Se entenderá por nodo cada participación física o parte significativa del sistema de información, con características propias de ejecución o función, e incluso, de diseño y construcción.

Para facilitar la comprensión del sistema, el mismo se documentará mediante un Modelo de Despliegue de Componentes de UML. A continuación se describen los elementos que contempla este tipo de diagrama:

- Nodos de procesamiento
- Dispositivos hardware
- Comunicación entre nodos y con dispositivos
- Componentes de software empaquetados en unidades instalables

#### **4.1.1.1. Descripción de los Niveles de Arquitectura del Sistema**

A continuación, en la figura DSI 4.1, se describen los componentes del presente sistema:

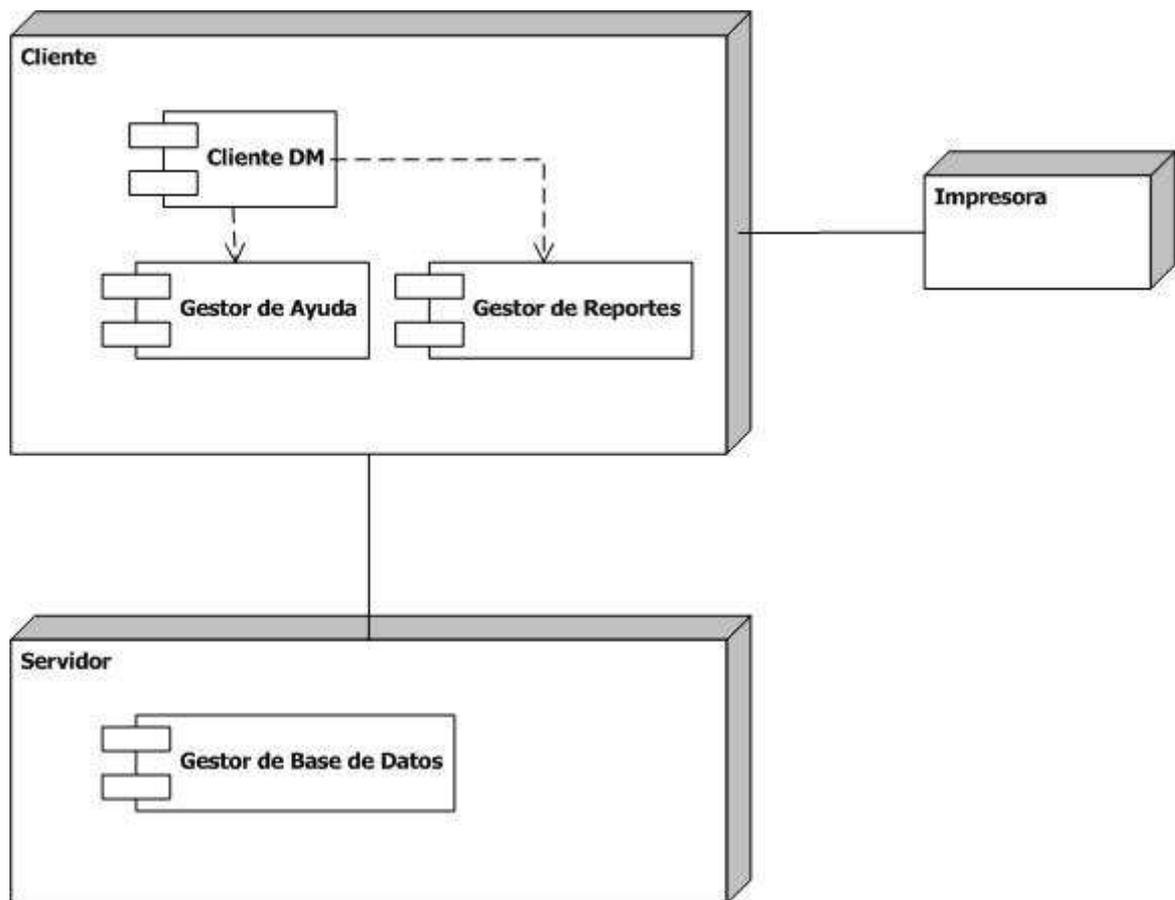


Figura DSI 4.1: Niveles de arquitectura del sistema

A continuación se describen los elementos del sistema identificados en la figura DSI 4.1:

Descripción de los nodos identificados:

a) *Equipo cliente:* Representa al equipo en el cual se desplegará la interfaz de usuario, si bien esta función puede ejecutarse también en el equipo donde se ubica la función del servidor, a continuación se describen las características del mismo para desempeñar solo la función de iteración con el usuario:

- Requerimientos de hardware
  - Un procesador Pentium II,
  - 128 Mb de Memoria RAM,

- 10 Mb libre en el disco rígido.

- Sistema operativo Microsoft Windows 2000 / XP

b) *Impresora*: Permite imprimir los reportes generados a través de la aplicación.

c) *Equipo servidor*: Representa al equipo en el cual se llevarán a cabo los procesos de manejo de la lógica de negocio y administración de la base de datos. Si bien sobre este mismo equipo se puede ejecutar la función de iteración con el usuario, los requisitos que se definen a continuación tienen que ver principalmente con la ejecución de los procesos de servicio que se proveerán al equipo cliente:

- Requerimientos de hardware

- un procesador Pentium II,
- 256 Mb de Memoria RAM,
- 10 Mb libre en el disco rígido.

- Sistema operativo Microsoft Windows 2000 / XP

- Base de datos Microsoft SQL Server

Descripción de los componentes identificados:

a) Cliente DM: este componente representa a la función del cliente del sistema, desde aquí el usuario podrá realizar la administración de los proyectos y versiones.

b) Gestor de ayuda: representa al manejador de ayuda CHM que brinda Microsoft FoxPro.

c) Gestor de Reportes: representa al manejador de reportes que brinda Microsoft FoxPro, el cual se utilizara para la gestión de impresión de reportes y listados.

d) Gestor de Base de datos: representa a la base de datos relacional donde se guarda la información referente a los usuarios, proyectos y versiones, como así también de parte

de la lógica del negocio vía Procedimientos Almacenados. Esta función será implementada en una base de datos Microsoft SQL Server.

La distribución de componentes mostrada en la figura DSI 4.1 ha tenido en cuenta los siguientes aspectos:

- Los usuarios se encuentran distribuidos dentro de la empresa u organización donde se implemente el sistema, lo cual implica que los mismos estarán ubicados en lugares físicos diferentes.
- Los datos deben estar centralizado. Esto permitirá a los distintos participantes del proyecto acceder a información unificada y consistente del mismo. Además, el hecho de que la información se encuentre unificada permite que solo sea necesario realizar un único backup para el resguardo de los datos así como la administración de seguridad de los mismos.
- Los procesos se encontrarán distribuidos entre los componentes clientes y servidor de la aplicación. De esta manera los componentes clientes se encargarán de las cuestiones referentes a un usuario en particular (carga de datos, consultas, etc.) y el componente servidor que tendrá que ser normativamente mas robusto dado que deberá soportar la concurrencia de múltiples usuarios y la gestión de los datos. Por otro lado, es indispensable asegurar el correcto funcionamiento de los mismos y su alta disponibilidad dado que ningún nodo cliente del sistema funcionará correctamente si los componentes del servidor no se encuentran disponibles.

A partir de este análisis, podemos identificar un segundo diagrama lógico, el cual muestra la relación entre los componentes identificados en la figura DSI 4.1 y es complementario al mismo:

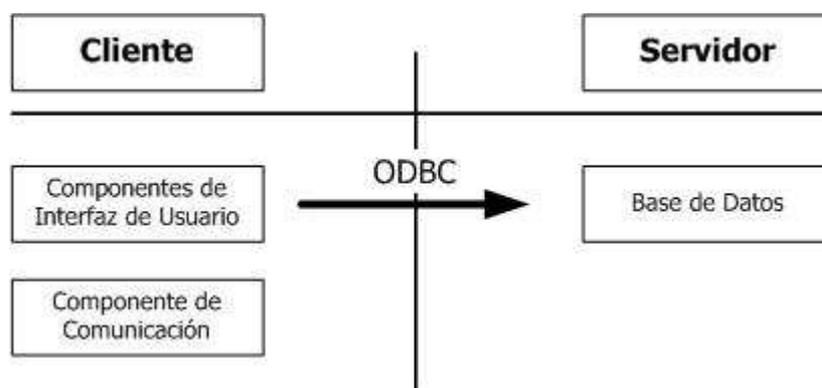


Figura DSI 4.2: Relación entre componentes

#### Descripción de la comunicación entre componentes:

a) Comunicación Cliente-Servidor: Esta comunicación se establece físicamente a través de la red local que debe estar implementada en la organización, sobre la cual se utilizarán los servicios de ODBC (Open DataBase Connectivity) provistos para el servidor de base de datos, en este caso en particular Microsoft SQL Server.

#### **4.1.2. Identificación de Requisitos de Diseño y Construcción**

En esta tarea se realiza la especificación de los requisitos que están directamente relacionados con la adopción o diseño de una arquitectura o infraestructura concreta, y que pueden condicionar el diseño o la construcción del sistema de información.

Entre estos requisitos pueden estar los relacionados con lenguajes, rendimiento de los distintos elementos de la arquitectura, así como criterios de ubicación de módulos y datos en los distintos nodos.

Por tanto, como resultado de esta tarea se actualiza el catálogo de requisitos elaborado en el proceso Análisis de Sistemas de Información.

##### **4.1.2.1. Descripción de los requisitos adicionales**

A continuación, en la Tabla DSI 4.1, se detallan los requisitos no funcionales identificados a lo largo de esta fase:

<b>Código de requisito</b>	<b>Identificación del requisito</b>	<b>Descripción del Requisito</b>
RNF-12	Impresión remota	El sistema debe poder imprimir en cualquier impresora accesible desde el equipo cliente, ya sea esta local o de red
RNF-13	Carga de trabajo del cliente	Los componentes del cliente serán desarrollados teniendo en cuenta soportar la carga de un único usuario por sesión
RNF-15	Backup centralizado	Recupero de datos centralizado.

Tabla DSI 4.1: Requisitos no funcionales de diseño

### 4.1.3. Especificaciones de Excepción

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

Se propone su catalogación como ayuda para el diseño del sistema de información y como guía en la especificación técnica de las pruebas, al permitir la generación de algunos casos de prueba de forma inmediata. Dicho catálogo se va completando a partir de las actividades correspondientes al diseño detallado de los subsistemas.

Las excepciones se describen incluyendo, al menos, los siguientes conceptos:

- Tipo y descripción de la excepción.
- Condiciones previas del sistema de información.
- Elemento afectado (nodo, módulo, caso de uso).
- Respuesta del sistema de información.
- Elemento asociado a la respuesta esperada del sistema (módulo, clase,

procedimiento, etc.).

Las excepciones que se proponen como obligatorias son las relacionadas con el funcionamiento general del sistema de información, habitualmente asociadas a:

- Nodos y comunicaciones del particionamiento físico del sistema de información. Este tipo de excepciones tiene lugar cuando no están disponibles los gestores de bases de datos o los recursos compartidos del sistema (representados como nodos), cuando se producen fallos en las comunicaciones entre nodos, etc.
- Rangos o valores no válidos en la entrada de datos, como pueden ser atributos obligatorios, con formatos específicos, etc.

Se recomienda, según el nivel de especificación que se establezca en cada caso, catalogar también las excepciones particulares que se identifiquen en las actividades del diseño de detalle.

#### **4.1.3.1. Catalogo de Excepciones**

Para el presente desarrollo se han determinado tres tipos de excepciones: comunicación, validación y permisos:

- *Las excepciones de comunicación* contemplan los problemas que pueden suscitarse cuando no existe conexión entre los componentes principales del sistema, es decir, cuando el cliente no puede comunicarse con el servidor, o cuando este último no puede comunicarse con la base de datos.
- *Las excepciones de validación* tienen que ver con aspecto que hacen a la conformidad de los datos a ingresar en los distintos campos de pantalla.
- *Las excepciones de permisos* están relacionadas con el control que hace el sistema para verificar que el usuario que está accediendo a un determinado documento posea los permisos necesarios para hacerlo.

A continuación en las tablas DSI 4.2 a DSI 4.6 se describen las excepciones que contempla el presente desarrollo:

<b>Excepción</b>	<b>EX-C001</b>
Tipo	Comunicación
Descripción	El componente servidor intenta comunicarse el sistema gestor de base de datos pero el mismo no responde.
Condiciones previas	<ul style="list-style-type: none"> <li>➤ El sistema no se encuentra conectado al gestor de base de datos.</li> <li>➤ El componente servidor ha recibido una petición del componente cliente para ejecutar una transacción</li> </ul>
Elemento afectado	Componente Servidor
Respuesta del sistema	El componente servidor debe comunicar al componente cliente la imposibilidad de ejecutar la transacción. Este mensaje debe ser informado al usuario: “El servidor informa que es imposible ejecutar la transacción indicada”

Tabla DSI 4.2: Excepción de comunicación

<b>Excepción</b>	<b>EX-C002</b>
Tipo	Comunicación
Descripción	El componente cliente logra comunicarse con el componente servidor, pero ocurre un error de comunicación en medio de la transacción
Condiciones previas	El componente cliente ejecuta la transacción en el componente servidor
Elemento afectado	Componente Cliente
Respuesta del sistema	Mensaje de error al usuario indicando la imposibilidad de ejecutar correctamente la transacción: “Ha ocurrido un error de comunicación”

Tabla DSI 4.3: Excepción de comunicación

<b>Excepción</b>	<b>EX-C003</b>
Tipo	Comunicación
Descripción	El sistema intenta enviar datos a la impresora local pero esta no responde
Condiciones previas	El componente cliente no está debidamente conectado a la impresora
Elemento afectado	Componente Cliente
Respuesta del sistema	Mensaje de error al usuario indicando la imposibilidad de imprimir el reporte: “Ha ocurrido un error de impresión”

Tabla DSI 4.4: Excepción de comunicación

<b>Excepción</b>	<b>EX-V001</b>
Tipo	Validación
Descripción	Se pretende cargar un dato cuyo valor se encuentra fuera de los rangos permitidos para el mismo.
Condiciones previas	Se selecciona una opción que requiere el ingreso de un valor.
Elemento afectado	Componente Cliente
Respuesta del sistema	Mensaje de error al usuario indicando que dato ingresado es inválido: “Ha ocurrido un error de comunicación”

Tabla DSI 4.5: Excepción de validación

<b>Excepción</b>	<b>EX-V002</b>
Tipo	Permiso
Descripción	Se pretende acceder a la base de datos, pero no se cuenta con los permisos de acceso necesarios.
Condiciones previas	Se seleccionó una versión e proyecto para la cual el usuario tenía permisos.
Elemento afectado	Componente Cliente
Respuesta del sistema	Mensaje de error al usuario indicando que no posee permisos para acceder a la base de datos:

Excepción	EX-V002
	“Ha ocurrido un error, no cuenta con permisos de acceso a la base de datos”

Tabla DSI 4.6: Excepción de validación

#### 4.1.4. Identificación de Subsistemas de Diseño

En esta tarea se divide de forma lógica el sistema de información en subsistemas de diseño, con el fin de reducir la complejidad y facilitar el mantenimiento. Hay que tomar como referencia inicial los subsistemas de análisis especificados en el proceso de Análisis del Sistema de Información (ASI).

La división en subsistemas de diseño se puede realizar con una continuidad directa de los modelos del análisis, o aplicando nuevos criterios de diseño, entre los que es posible citar los siguientes:

- Facilidad de mantenimiento.
- Reutilización de elementos del propio sistema o de la instalación.
- Optimización de recursos (por ejemplo, líneas de comunicaciones).
- Características de ejecución (en línea o por lotes).
- Funcionalidad común.
- Aplicación de mecanismos genéricos de diseño al nivel de arquitectura.

Los subsistemas resultantes se califican como específicos o de soporte, asignando cada subsistema al nodo correspondiente.

Los subsistemas específicos contemplan las funcionalidades propias del sistema de información, mientras que los de soporte cubren servicios comunes, proporcionando un acceso transparente a los distintos recursos. Estos últimos están relacionados con:

- Comunicaciones entre subsistemas.

- Gestión de datos (acceso a bases de datos, ficheros, áreas temporales, importación y exportación de datos, sincronización de bases de datos, etc.).
- Gestión de transacciones.
- Control y gestión de errores.
- Seguridad y control de acceso.
- Gestión de interfaz.
- Interacción con los recursos propios del sistema.

La interacción del sistema de información con la infraestructura que le da soporte, así como con el resto de los sistemas y servicios de la instalación, puede originar la necesidad de nuevos subsistemas, módulos, clases o servicios no especificados en el análisis.

La definición del comportamiento externo de cada subsistema se completa durante el diseño de detalle con la especificación de su interfaz, así como con la dependencia entre subsistemas.

El diseño de detalle de los subsistemas identificados por criterios de optimización y reutilización, puede aconsejar la reorganización y reubicación de los elementos que forman parte de cada subsistema y, a su vez, puede dar lugar a la identificación de nuevos subsistemas de soporte.

En diseño estructurado, la descripción de los subsistemas de diseño que conforman el sistema de información se especifica mediante un diagrama de estructura de alto nivel, que muestra los distintos subsistemas de que consta el sistema, incluidos los subsistemas de soporte, junto con la definición de la interfaz de cada subsistema.

La ubicación de subsistemas en nodos y la dependencia entre subsistemas se especifica por medio de técnicas matriciales, o bien en lenguaje natural o pseudocódigo.

#### **4.1.4.1. Subsistemas de Diseño**

En esta sección se divide de forma lógica el sistema en subsistemas para el diseño, para reducir la complejidad y facilitar el mantenimiento del mismo. Para facilitar esta tarea se va a aplicar la arquitectura cliente/servidor, la cual consiste de distribuir la carga de trabajo entre el cliente y el servidor de base de datos.

La arquitectura de software cliente/servidor es una infraestructura versátil, modular, basada en mensajes, cuya intención es mejorar usabilidad, flexibilidad, interoperabilidad y escalabilidad, con respecto a los sistemas centralizados, mainframe y de tiempo compartido. Definen a un cliente como a un consumidor de servicios y a un servidor como el proveedor de servicios. Una sola máquina puede ser un cliente y un servidor a la vez, dependiendo de la configuración del software. Usando un sistema de base de datos relacional, las consultas de un usuario se podrían contestar directamente. En arquitecturas cliente/servidor para comunicarse entre el cliente y el servidor se utilizan típicamente las llamadas remotas a procedimientos (Remote Procedure Calls - RPCs) o el lenguaje estándar de consultas (Standard Query Language - SQL). En este sistema se utilizará el lenguaje estándar de consultas (SQL) para la comunicación.

#### **4.1.5. Especificación del Entorno Tecnológico**

En esta tarea se definen en detalle los distintos elementos de la infraestructura técnica que dan soporte al sistema de información, determinando la implementación concreta de los nodos y comunicaciones especificados en la tarea Definición de Niveles de Arquitectura (DSI 1.1). Para esto, se propone agrupar los elementos de la infraestructura en los siguientes conceptos:

- Hardware: procesadores, unidades de almacenamiento, estaciones de trabajo, etc.
- Software: sistemas operativos, subsistemas, middleware, gestores de bases de datos, sistemas de ficheros, software de base, herramientas y utilidades de gestión propias del sistema, etc.

- Comunicaciones: diseño de la topología de la red, protocolos, nodos de red, etc.

La definición de los distintos elementos puede generar restricciones técnicas que afecten al diseño o construcción del sistema de información.

Asimismo, se realiza una estimación de la planificación de capacidades (capacity planning) o se especifican los parámetros que Explotación y Sistemas precisen para realizar dicha planificación. Se indican, al menos, las necesidades previstas de:

- Almacenamiento: espacio en disco, espacio en memoria, pautas de crecimiento y evolución estimada del sistema de información, etc.
- Procesamiento: número y tipo de procesadores, memoria, etc.
- Comunicaciones: líneas, caudal, capacidades de elementos de red, etc.

Para poder determinar la planificación de capacidades, es necesario conocer los diseños detallados de los módulos / clases y escenarios, incluida la información de control en las comunicaciones, así como el diseño físico de datos optimizado, productos que se están generando en paralelo a esta actividad. También se tienen en cuenta, cuando proceda, las estimaciones de volúmenes de datos propios de la migración y carga inicial de datos.

#### **4.1.5.1. Especificación de Hardware**

El sistema podrá ser ejecutado en equipos de distintas tecnologías. Se prevén las siguientes configuraciones mínimas:

- Plataforma Intel: Procesador Pentium II o superior. 128 Mb. De RAM. 100 Mb de espacio libre en disco. Placa de Red.
- Plataforma AMD: Procesador Athlon o superior. 128 Mb. De RAM. 100 Mb de espacio libre en disco. Placa de Red.

#### **4.1.5.2. Especificación de Software**

Para ambas plataformas:

- Sistema operativo Microsoft Windows.
- Base de datos compatible ANSI SQL.

#### **4.1.5.3. Especificación de Comunicación**

Como ya se mencionó el sistema esta preparado para ser ejecutado sobre una red de área local, preferentemente del tipo Ethernet.

#### **4.1.6. Especificación de Requisitos de Operación y Seguridad**

El objetivo de esta tarea es definir los procedimientos de seguridad y operación necesarios para no comprometer el correcto funcionamiento del sistema y garantizar el cumplimiento de los niveles de servicios que exigirá el sistema en cuanto a la gestión de operaciones (procesos por lotes, seguridad, comunicaciones, etc.). Los niveles de servicio se especifican formalmente en el proceso Implantación y Aceptación del Sistema (IAS).

Tomando como referencia los requisitos establecidos para el sistema, y teniendo en cuenta la arquitectura propuesta y las características del entorno tecnológico definido en esta actividad, se lleva a cabo la definición de los requisitos de seguridad y control de acceso necesarios para garantizar la protección del sistema y minimizar el riesgo de pérdida, alteración o consulta indebida de la información. Para ello, se diseñan los procedimientos relacionados con:

- Acceso al sistema y a sus recursos (datos, transacciones, librerías, etc.).
- Mantenimiento de la integridad y confidencialidad de los datos.
- Control y registro de accesos al sistema (logs, certificación, etc.).
- Copias de seguridad y recuperación de datos y su periodicidad.

- Recuperación ante catástrofes.

Asimismo, se definen los requisitos de operación para los distintos elementos del sistema (módulos, clases, estructuras físicas de datos, sistemas de ficheros), que se están elaborando en paralelo a esta actividad, y se diseñan los procedimientos asociados relacionados con:

- Tratamiento en línea (franja horaria/periodos críticos, número máximo de usuarios, etc.).
- Tratamiento por lotes (periodicidad y secuencia de ejecución, interdependencias, petición de ejecución, etc.).
- Control y planificación de trabajos.
- Recuperación y reanudación de trabajos.
- Distribución de información generada por el sistema, tanto trabajos planificados o bajo petición.
- Control y seguimiento del correcto funcionamiento de los procedimientos de backup y recuperación utilizados habitualmente.

#### **4.1.6.1. Acceso al sistema y a sus recursos**

El sistema cuenta con una base de datos relacional para almacenar sus datos y ejecutar determinadas transacciones. El acceso a dichos datos, estructura de datos y transacciones se encuentra protegido por el mecanismo de autenticación básica provisto por el vendedor de la base de datos en cuestión. En este caso la base de datos provee un sistema de seguridad basado en usuario y contraseña y un mecanismo que permite legislar los equipos desde los cuales es posible conectarse, o bien, delegar la autenticación al sistema operativo.

Los pares usuario/contraseña que utilizaran los usuarios del sistema son de nivel aplicativo, es decir, son administrados por la aplicación y no tiene sentido o uso a nivel de sistema operativo o base de datos. Cuando el usuario quiera operar con el gestor de

base de datos deberá contar con los permisos correspondientes, obtenidos vía el administrador de la misma.

#### **4.1.6.2. Mantenimiento de la integridad y confidencialidad de los datos**

La confidencialidad de los datos se obtiene debido a que la aplicación no permite ver los datos que no estén relacionados con el usuario autenticado que está usando el sistema; y por otro lado, la base de datos se encuentra protegida dentro del servidor de aplicaciones en una carpeta con acceso restringido.

#### **4.1.6.3. Control y registro de acceso al sistema**

Se llevará a cabo un registro de auditoría, el cual mantendrá información sobre las operaciones no triviales realizadas por el usuario, por ejemplo:

- Ingresar al sistema,
- Ejecutar una transacción

Dicho registro estará en una tabla .dbf local. El nombre del archivo de los será “dm\_log.dbf”. En esta tabla se registrará cada operación no trivial y el momento de su ejecución (fecha y hora), permitiendo una fácil ubicación de las mismas.

#### **4.1.6.4. Copia de seguridad y recuperación de datos y su periodicidad**

Las bases de datos relacionales prevén mecanismos específicos para los resguardos de seguridad y la recuperación ante una eventual necesidad. Estos mecanismos varían desde el backup a nivel sistema de archivo hasta copias replicadas en línea para cambiar el servidor de base de datos y continuar operando sin interrupciones.

Dado que el actual proyecto está involucrado dentro del marco de un trabajo académico, que no cuenta con presupuesto alguno y que el RDBMS utilizado carece de características avanzadas de backup y recuperación, se recomienda un esquema de backup diario (o la frecuencia que sea acorde a su uso) basado en copias del sistema de archivos en CD-ROM.

#### 4.1.6.5. Recuperación y reanudación de trabajos

Los trabajos no pueden ser recuperados o reanudados. Simplemente, cada operación puede ejecutarse satisfactoriamente, o no. En caso de una interrupción en el servicio sucederá lo indicado en la Tabla DSI 4.7:

Origen de la interrupción	Consecuencias
Fallo en el cliente que ocasiona la caída del sistema	El sistema no guarda estado de sus objetos en el cliente. Con lo cual, cualquier operación no terminada (enviada al servidor) será deshecha por completo. Al reanudar el uso del sistema, el usuario deberá volver a ejecutar todos los pasos hechos anteriormente.
Fallo en el componente base de datos, que ocasiona la caída del mismo	La base de datos posee lógica de gestión de transacciones. Por lo tanto, una caída en este componente resultará en volver las transacciones abiertas al último estado consistente anterior.

Tabla DSI 4.7: Consecuencias de la interrupción del servicio

## 4.2. Diseño de la Arquitectura de Soporte

En esta actividad se lleva a cabo la especificación de la arquitectura de soporte, que comprende el diseño de los subsistemas de soporte identificados en la actividad de Definición de la Arquitectura del Sistema (DSI 1), y la determinación de los mecanismos genéricos de diseño. Estos últimos sirven de guía en la utilización de diferentes estilos de diseño, tanto en el ámbito global del sistema de información, como en el diseño de detalle.

El diseño de los subsistemas de soporte, conceptualmente, es similar al diseño de los subsistemas específicos, aunque debe cumplir con unos objetivos claros de reutilización. De esta manera, se consigue simplificar y abstraer el diseño de los subsistemas específicos de la complejidad del entorno tecnológico, dotando al sistema de información de una mayor independencia de la infraestructura que le da soporte. Con

este fin, se aconseja la consulta de los datos de otros proyectos existentes, disponible en el Histórico de Proyectos. Si esto no fuera suficiente, se puede contar en esta actividad con la participación de perfiles técnicos, con una visión global de la instalación.

Esta actividad se realiza en paralelo al diseño detallado, debido a que existe una constante realimentación, tanto en la especificación de los subsistemas con sus interfaces y dependencias, como en la aplicación de esqueletos o patrones en el diseño.

Los productos resultantes de esta actividad son:

- Diseño Detallado de los Subsistemas de Soporte.
- Mecanismos Genéricos de Diseño y Construcción.

#### **4.2.1. Diseño de Subsistemas de Soporte**

El objetivo de esta tarea es la especificación y diseño de los módulos/clases que forman parte de los subsistemas de soporte, identificados en la tarea Identificación de Subsistemas de Diseño. Se lleva a cabo siempre y cuando no se disponga en la instalación de servicios comunes que respondan satisfactoriamente a los requisitos planteados.

El nivel de reutilización de los subsistemas de soporte y sus servicios es potencialmente alto, de modo que se debe intentar emplear, en la medida de lo posible, los subsistemas que ya existan en la instalación y se consideren viables. La información relativa a dichos subsistemas podrá obtenerse del Histórico de Proyectos. En cualquier caso, cuando proceda realizar el diseño de los subsistemas de soporte, se recomienda hacerlo con ese fin. El diseño sigue las mismas pautas que las establecidas para los subsistemas específicos, aunque con las siguientes particularidades:

- Generalmente, será necesaria una descomposición de los subsistemas de soporte en servicios, entendiendo como tales módulos o clases independientes y reutilizables.

- Se recomienda realizar una descripción de la interfaz y del comportamiento de cada servicio, previa a su diseño de detalle, que permita completar el diseño de los subsistemas específicos.
- La especificación y diseño de cada servicio, módulo o clase, se realiza con las técnicas habituales de especificación y diseño de módulos o clases, o incluso opcionalmente, si la simplicidad de los elementos lo aconseja, otros lenguajes de especificación, pseudocódigo o lenguaje natural.

A medida que se lleva a cabo esta tarea pueden surgir comportamientos de excepción que deberán contemplarse igualmente en el diseño, y que en función del nivel de especificación que se haya establecido, se incorporan al catálogo de excepciones.

### **4.3. Diseño de Casos de Uso Reales**

Esta actividad, que se realiza solo en el caso de Diseño Orientado a Objetos, tiene como propósito especificar el comportamiento del sistema de información para un caso de uso, mediante objetos o subsistemas de diseño que interactúan, y determinar las operaciones de las clases e interfaces de los distintos subsistemas de diseño.

Para ello, una vez identificadas las clases participantes dentro de un caso de uso, es necesario completar los escenarios que se recogen del análisis, incluyendo las clases de diseño que correspondan y teniendo en cuenta las restricciones del entorno tecnológico, esto es, detalles relacionados con la implementación del sistema. Es necesario analizar los comportamientos de excepción para dichos escenarios. Algunos de ellos pueden haber sido identificados en el proceso de análisis, aunque no se resuelven hasta este momento. Dichas excepciones se añadirán al catálogo de excepciones para facilitar las pruebas.

Algunos de los escenarios detallados requerirán una nueva interfaz de usuario. Por este motivo es necesario diseñar el formato de cada una de las pantallas o impresos identificados.

Es importante validar que los subsistemas definidos en la tarea Identificación de Subsistemas de Diseño tienen la mínima interfaz con otros subsistemas. Por este motivo, se elaboran los escenarios al nivel de subsistemas y, de esta forma, se delimitan las interfaces necesarias para cada uno de ellos, teniendo en cuenta toda la funcionalidad del sistema que recogen los casos de uso. Además, durante esta actividad pueden surgir requisitos de implementación, que se recogen en el catálogo de requisitos.

Las tareas de esta actividad se realizan en paralelo con las de Diseño de Clases .

#### **4.3.1. Identificación de Clases Asociadas a un Caso de Uso**

El objetivo de esta tarea es identificar las clases que intervienen en cada caso de uso, a partir del conjunto de clases definidas en la tarea Identificación de Clases Adicionales, ya que, como se ha señalado en la introducción de esta actividad, las actividades Diseño de casos de uso reales y Diseño de clases se realizan en paralelo. Dichas clases se identifican a partir de las clases del modelo del análisis y de aquellas clases adicionales necesarias para el escenario que se está diseñando.

A su vez, a medida que se va estudiando la descripción de los casos de uso, pueden aparecer nuevas clases de diseño que no hayan sido identificadas anteriormente y que se incorporan al modelo de clases en la tarea Identificación de Clases Adicionales.

##### **4.3.1.1. Clases Asociadas a los Casos de Usos**

A continuación, en las figuras DSI 4.3 a DSI 4.16, se exponen de manera esquemática las relaciones entre los casos de uso, los actores y las clases, para cada caso de uso:

a) Caso de uso Ingreso al sistema (RF01):

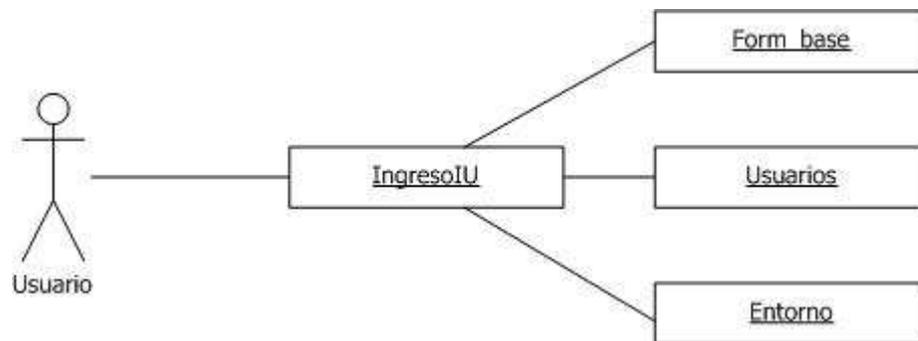


Figura DSI 4.3: Esquema de relaciones del caso de uso Ingreso al sistema.

b) Caso de Uso Agregar Usuario (RF03):

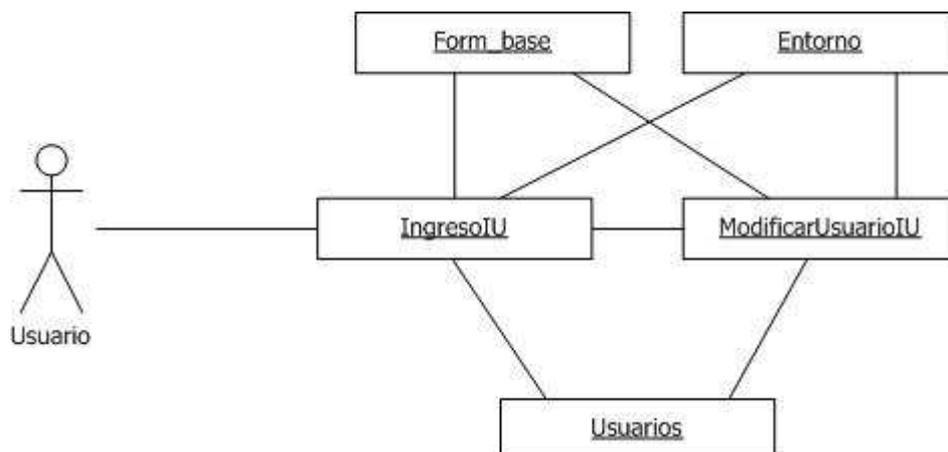


Figura DSI 4.4: Esquema de relaciones del caso de uso Agregar Usuario.

c) Caso de Uso Modificar Usuario (RF04):

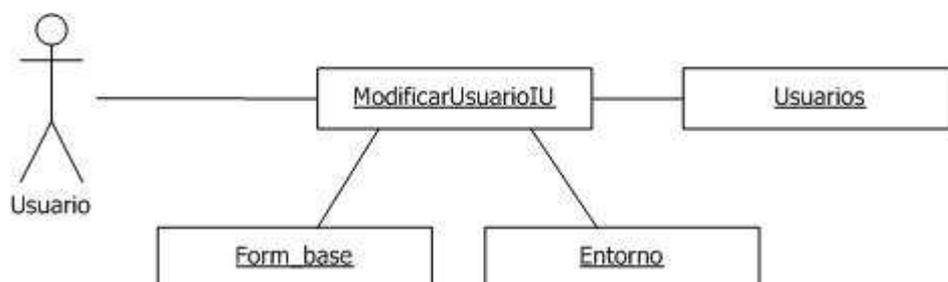


Figura DSI 4.5: Esquema de relaciones del caso de uso Modificar Usuario.

d) Caso de Uso Crear Proyecto (RF05):

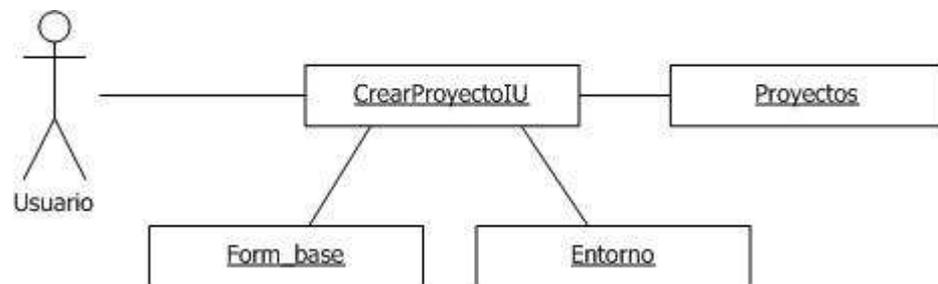


Figura DSI 4.6: Esquema de relaciones del caso de uso Crear Proyecto.

e) Caso de Uso Eliminar Proyecto (RF06):

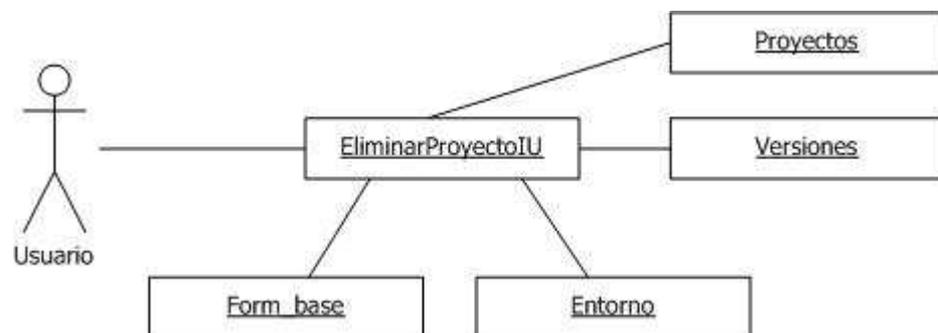


Figura DSI 4.7: Esquema de relaciones del caso de uso Eliminar Proyecto.

f) Caso de Uso Listar Proyectos (RF07):

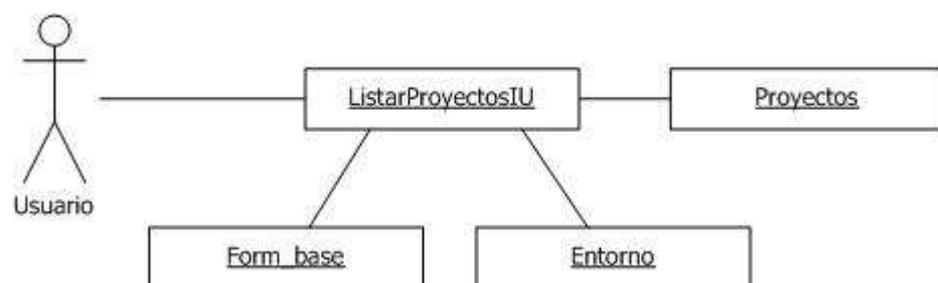


Figura DSI 4.8: Esquema de relaciones del caso de uso Listar Proyectos.

g) Caso de Uso Salir del Sistema (RF02):



Figura DSI 4.9: Esquema de relaciones del caso de uso Salir del Sistema.

h) Caso de Uso Crear Versiones de un Proyecto (RF08):

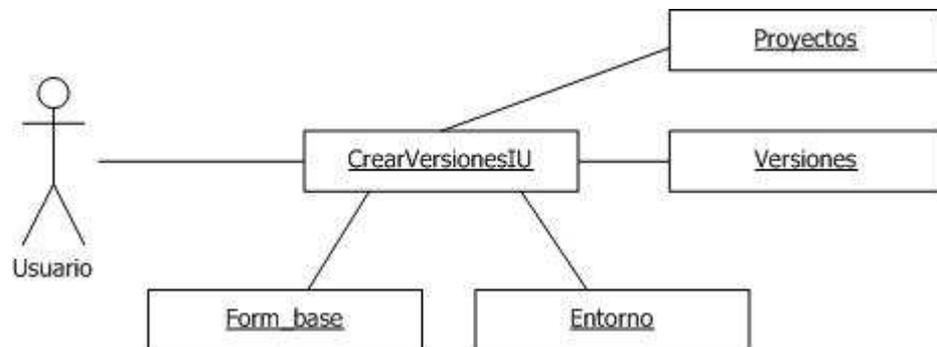


Figura DSI 4.10: Esquema de relaciones del caso de uso Crear Versiones de un Proyecto.

i) Caso de Uso Abrir Versión (RF09):

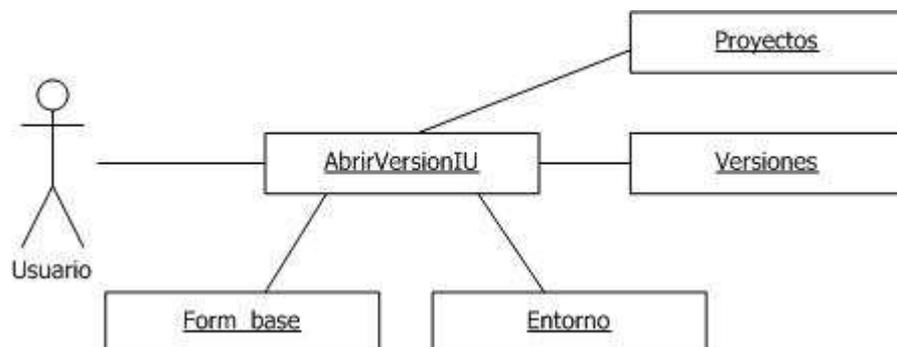


Figura DSI 4.11: Esquema de relaciones del caso de uso Abrir Versión.

j) Caso de Uso Eliminar Versión (RF10):

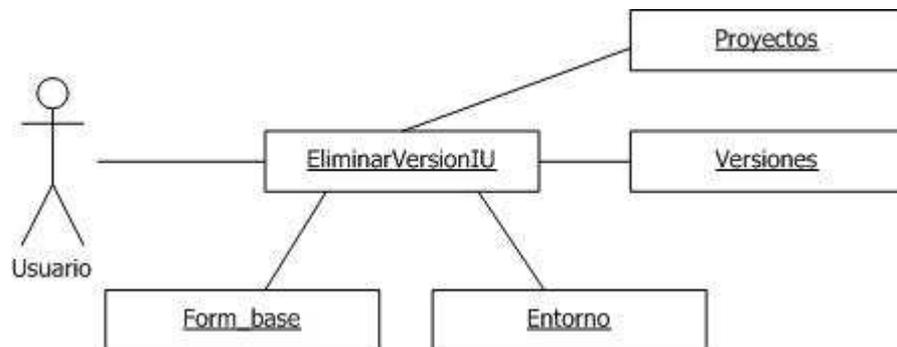


Figura DSI 4.12: Esquema de relaciones del caso de uso Eliminar Versión.

k) Caso de Uso Listar Versiones (RF11):

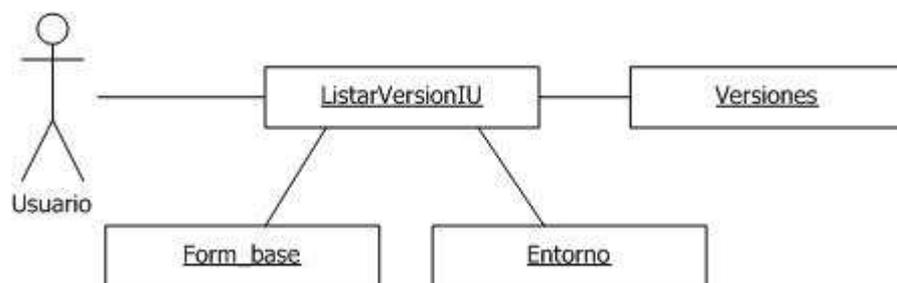


Figura DSI 4.13: Esquema de relaciones del caso de uso Listar Versiones.

l) Caso de Uso Establecer Parámetros de SOM (RF12):

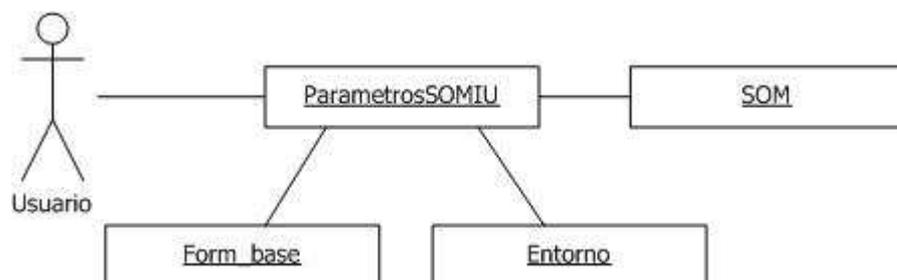


Figura DSI 4.14: Esquema de relaciones del caso de uso Establecer Parámetros de SOM.

m) Caso de Uso Establecer Parámetros de C5.0 (RF13):

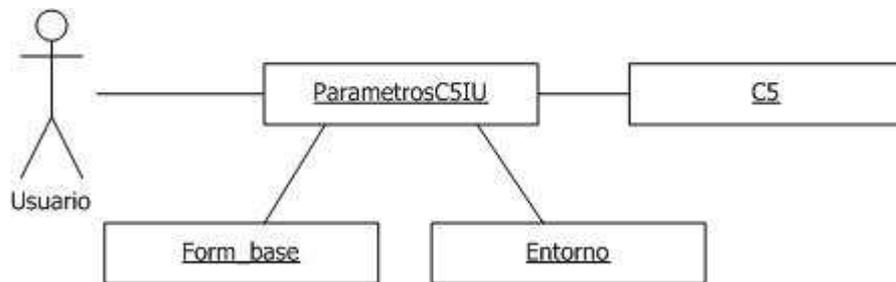


Figura DSI 4.15: Esquema de relaciones del caso de uso Establecer Parámetros de C5.0.

n) Caso de Uso Ejecutar Proceso de Explotación de Datos (RF14):

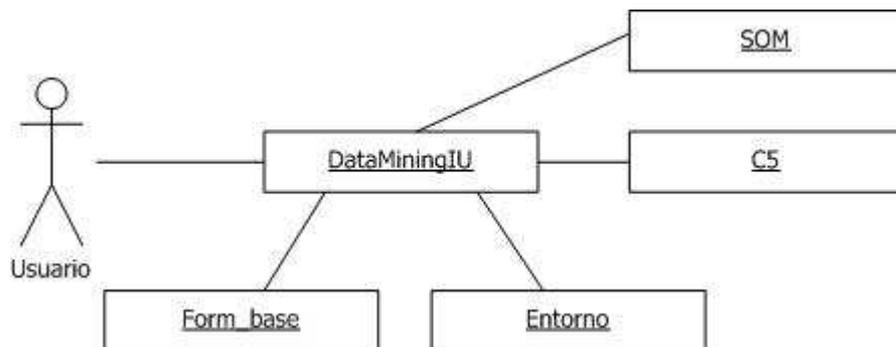


Figura DSI 4.16: Esquema de relaciones del caso de uso Ejecutar Proceso de Explotación de Datos.

#### 4.3.2. Revisión de la Interfaz de Usuario

El objetivo de esta tarea es realizar el diseño detallado del comportamiento de la interfaz de usuario a partir de la especificación de la misma, obtenida en el proceso de análisis, y de acuerdo con el entorno tecnológico definido. Si se hubiera realizado un prototipo de la interfaz de usuario, éste se tomaría como punto de partida para el diseño.

Además, se incluyen las ventanas alternativas o elementos de diseño surgidos como consecuencia del diseño de los escenarios definidos en la tarea anterior. Además, se revisa: la interfaz de usuario, la navegación entre ventanas, los elementos que forman cada interfaz, sus características (que deben ser consistentes con los atributos con los

que están relacionadas), su disposición, y cómo se gestionan los eventos relacionados con los objetos.

En aquellos casos en los que se realizan modificaciones significativas sobre la interfaz de usuario, es conveniente que éste las valide, siendo opcional la realización de un nuevo prototipo.

#### **4.3.2.1. Resultado de la revisión de la Interfaz de Usuario**

Como puede verse en el apartado anterior, no aparecen nuevas clases de interfaz que deban ser tratadas en este punto. Igualmente se ha realizado una comparación entre las clases de interfaz antes definidas y las actuales clases de diseño y no se han encontrado diferencias significativas que justifiquen la modificación del actual modelo.

#### **4.4. Diseño de Clases**

El propósito de esta actividad, que se realiza sólo en el caso de Diseño Orientado a Objetos, es transformar el modelo de clases lógico, que proviene del análisis, en un modelo de clases de diseño. Dicho modelo recoge la especificación detallada de cada una de las clases, es decir, sus atributos, operaciones, métodos, y el diseño preciso de las relaciones establecidas entre ellas, bien sean de agregación, asociación o jerarquía. Para llevar a cabo todos estos puntos, se tienen en cuenta las decisiones tomadas sobre el entorno tecnológico y el entorno de desarrollo elegido para la implementación.

Se identifican las clases de diseño, que denominamos clases adicionales, en función del estudio de los escenarios de los casos de uso, que se está realizando en paralelo en la actividad Diseño de Casos de Uso Reales, y aplicando los mecanismos genéricos de diseño que se consideren convenientes por el tipo de especificaciones tecnológicas y de desarrollo. Entre ellas se encuentran clases abstractas, que integran características comunes con el objetivo de especializarlas en clases derivadas. Se diseñan las clases de interfaz de usuario, que provienen del análisis. Como consecuencia del estudio de los escenarios secundarios que se está realizando, pueden aparecer nuevas clases de interfaz.

También hay que considerar que, por el diseño de las asociaciones y agregaciones, pueden aparecer nuevas clases, o desaparecer incluyendo sus atributos y métodos en otras, si se considera conveniente por temas de optimización.

La jerarquía entre las clases se va estableciendo a lo largo de esta actividad, a medida que se van identificando comportamientos comunes en las clases, aunque haya una tarea propia de diseño de la jerarquía.

Otro de los objetivos del diseño de las clases es identificar para cada clase, los atributos, las operaciones que cubren las responsabilidades que se identificaron en el análisis, y la especificación de los métodos que implementan esas operaciones, analizando los escenarios del Diseño de Casos de Uso Reales, luego, se determina la visibilidad de los atributos y operaciones de cada clase, con respecto a las otras clases del modelo.

Una vez que se ha elaborado el modelo de clases, se define la estructura física de los datos correspondiente a ese modelo, en la actividad Diseño Físico de Datos.

Además, en los casos en que sea necesaria una migración de datos de otros sistemas o una carga inicial de información, se realizará su especificación a partir del modelo de clases y las estructuras de datos de los sistemas origen.

Como resultado de todo lo anterior se actualiza el modelo de clases del análisis, una vez recogidas las decisiones de diseño.

#### **4.4.1. Identificación de Atributos de las Clases**

El objetivo de esta tarea es identificar y describir, una vez que se ha especificado el entorno de desarrollo, los atributos de las clases.

Para identificar los atributos se revisa el modelo de clases obtenido en el proceso de Análisis del Sistema de Información (ASI), considerando que, a partir de uno de ellos, puede ser necesario definir atributos adicionales. Para cada atributo identificado se define su tipo, con formatos específicos, y si existieran, las restricciones asociadas a ese atributo. Asimismo, se analiza la posibilidad de convertir un atributo en clase en

aquellos casos en los que:

- El atributo se defina en varias clases de diseño.
- La complejidad del atributo aumente la dificultad para comprender la clase a la que pertenece.

#### 4.4.1.1. Atributos de Clases

Antes de pasar a definir los atributos de las clases, se mostrará, en la Tabla DSI 4.8, como se vinculan las clases de análisis y diseño.

<b>Clases de Análisis</b>	<b>Clases de diseño</b>
Proyecto	Proyecto
Version	Version
Usuario	Usuario
SOM	SOM
C5	C5
-	Entorno
-	Form_base

Tabla DSI 4.8: Comparación entre clases de Análisis y Diseño

En la figura DSI 4.17 se muestra el diagrama de clases de diseño del sistema en cuestión. Se especifica para cada clase sus campos y mensajes:

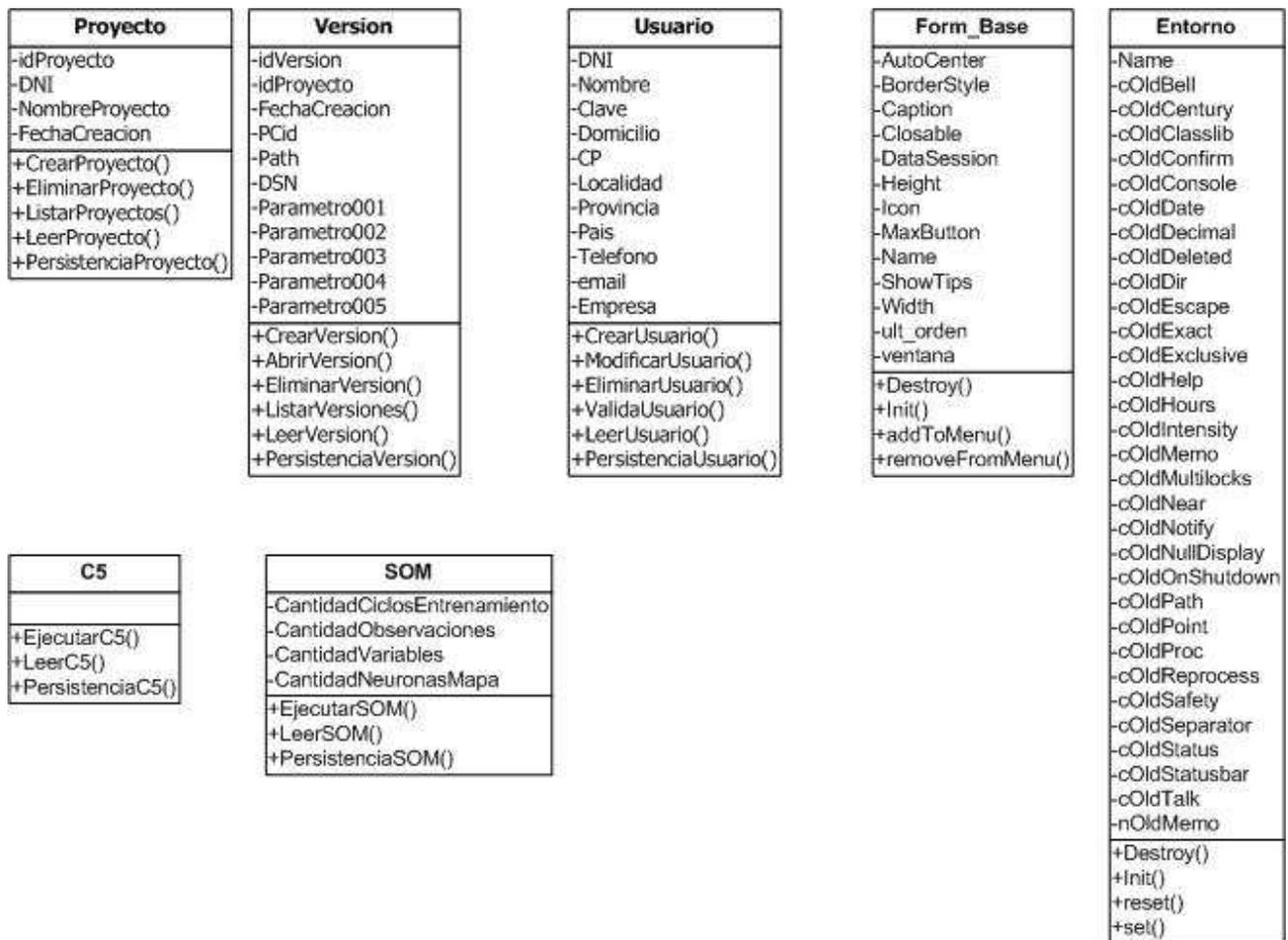


Figura DSI 4.17: diagrama de clases de diseño

A continuación, en la Tabla DSI 4.9, se completa la información de los atributos de las clases identificados en la fase de análisis:

Clase	Atributos	Tipo
C5	LimpiezaAtributos	Bit
	Reglas	Bit
	OrdUtilidad	Bit
	OrdUtilidadBandas	Int
	Boost	Bit
	BoostEj	Int
	SubcjoValores	Bit
	UsarEj	Bit
UsarEjPorc	Int	

<b>Clase</b>	<b>Atributos</b>	<b>Tipo</b>
	LockEj ValidacionCruzada ValidacionCruzadaHojas IgnorarArchivoCostos FuzzyThresholds PodaGlobal PodaCF MinimoCasos	Bit Bit Int Bit Bit Int Int
Entorno	(*)	
Form_base	(*)	
Proyecto	IdProyecto DNI NombreProyecto FechaCreacion	Int String String Date
SOM	CantidadCiclosEntrenamiento CantidadObservaciones CantidadVariable CantidadNeuronasMapa	Int Int Int Int
Usuario	DNI Nombre Clave Domicilio CP Localidad Provincia Pais Telefono email Empresa	Int String String String String String Int String String String String
Version	idVersion idProyecto FechaCreacion PCid Path DSN Parametro001	Int Int Date String String String String

Clase	Atributos	Tipo
	Parametro002	String
	Parametro003	String
	Parametro004	String
	Parametro005	String

(\*) son clases solo de diseño

Tabla DSI 4.9: Atributos de las clases.

#### 4.4.2. Identificación de Operaciones de las Clases

El objetivo de esta tarea es definir, de forma detallada, las operaciones de cada clase de diseño. Para ello, se toma como punto de partida el modelo de clases generado en el análisis, así como el diseño de los casos de uso reales y los requisitos de diseño que pueden aparecer al definir el entorno de desarrollo.

Las operaciones de las clases de diseño surgen para dar respuesta a las responsabilidades de las clases de análisis y, además, para definir las interfaces que ofrece esa clase.

Según el entorno de desarrollo utilizado, se describe cada operación especificando: su nombre, parámetros y visibilidad (pública, privada, protegida). Si el entorno de desarrollo lo permite, se tiene en cuenta la posibilidad de simplificar el modelo de clases haciendo uso del polimorfismo y la sobrecarga de operaciones.

Para identificar las operaciones de aquellos objetos que presenten distintos estados, por lo que su comportamiento depende del estado en el que se encuentren, es recomendable realizar un diagrama de transición de estados, y traducir cada acción o actividad del mismo en una de estas operaciones.

##### 4.4.2.1. Operaciones de Clases

En el presente trabajo, no se encuentran objetos cuyo comportamiento cambie en función de su estado, por lo tanto, no se desarrollarán diagrama de transición de estados.

Para comenzar con el análisis de los métodos, hay que tener en cuenta para facilitar el entendimiento del modelo, se documentarán solo los métodos no triviales, que son aquellos que agregan valor al sistema, es decir, modelan reglas de negocio y comportamiento de las clases del sistema. Estos métodos si serán diseñados y explicados en las tablas DSI 4.10 a DSI 4.14:

a) Proyecto:

<b>Método</b>	<b>Tipo</b>	<b>Descripción</b>
CrearProyecto	Público	Da de alta a un proyecto vacío para el usuario activo en ese momento.
	Parámetros	IdProyecto, DNI, NombreProyecto, FechaCreacion
	Resultado	CodigoRetorno
EliminarProyecto	Público	Borra el proyecto elegido, como así también todas sus versiones.
	Parámetros	IdProyecto, DNI
	Resultado	CodigoRetorno
ListarProyectos	Público	Recupera la lista de todos los proyectos del usuario activo.
	Parámetros	DNI
	Resultado	IdProyecto, DNI, NombreProyecto, FechaCreacion
LeerProyecto	Público	Recupera todos los datos de un proyecto en particular.
	Parámetros	IdProyecto, DNI.
	Resultado	IdProyecto, DNI, NombreProyecto, FechaCreacion
PersistenciaProyecto	Público	Inserta o actualiza en una tabla de la base de datos los datos de un proyecto.
	Parámetros	IdProyecto, DNI, NombreProyecto, FechaCreacion
	Resultado	CodigoRetorno

Tabla DSI 4.10: Descripción de los métodos de la clase Proyecto.

b) Version:

<b>Método</b>	<b>Tipo</b>	<b>Descripción</b>
CrearVersion	Público	Da de alta una versión en el proyecto activo en ese momento.
	Parámetros	IdVersion, IdProyecto, FechaCreacion, PCId, Path, DSN, Parametro001, Parametro002, Parametro003, Parametro004, Parametro005
	Resultado	CodigoRetorno
AbrirVersion	Público	Establece una versión de un proyecto como activa para trabajar en ella.
	Parámetros	IdVersion, IdProyecto, DNI
	Resultado	CodigoRetorno
EliminarVersion	Público	Borra la versión elegida del proyecto.
	Parámetros	IdVersion, IdProyecto, DNI
	Resultado	CodigoRetorno
ListarVersion	Público	Recupera la lista de todas las versiones de los proyectos del usuario activo.
	Parámetros	DNI
	Resultado	IdVersion, IdProyecto, FechaCreacion
LeerVersion	Público	Recupera todos los datos de una versión de proyecto en particular.
	Parámetros	IdVersion, IdProyecto, DNI.
	Resultado	IdVersion, IdProyecto, FechaCreacion, PCId, Path, DSN, Parametro001, Parametro002, Parametro003, Parametro004, Parametro005
PersistenciaVersion	Público	Inserta o actualiza en una tabla de la base de datos los datos de la versión de proyecto.
	Parámetros	IdVersion, IdProyecto, FechaCreacion, PCId, Path, DSN, Parametro001, Parametro002, Parametro003, Parametro004, Parametro005
	Resultado	CodigoRetorno

Tabla DSI 4.11: Descripción de los métodos de la clase Version.

c) Usuario:

<b>Método</b>	<b>Tipo</b>	<b>Descripción</b>
CrearUsuario	Público	Crea un nuevo usuario para el sistema.
	Parámetros	DNI, Nombre, Clave, Domicilio, CP, Localidad, Provincia, Pais, Telefono, email, Empresa
	Resultado	CodigoRetorno
ModificarUsuario	Público	Modifica alguno de los atributos (excepto DNI) del usuario activo.
	Parámetros	Nombre, Clave, Domicilio, CP, Localidad, Provincia, Pais, Telefono, email, Empresa
	Resultado	CodigoRetorno
EliminarUsuario	Público	Elimina a un usuario de la nomina.
	Parámetros	DNI
	Resultado	CodigoRetorno
ValidaUsuario	Público	Verifica que la clave de ingreso al sistema coincida con la correspondiente a ese DNI.
	Parámetros	DNI, Clave
	Resultado	CodigoRetorno
LeerUsuario	Público	Trae de base de datos los datos correspondientes a un usuario.
	Parámetros	DNI
	Resultado	Nombre, Clave, Domicilio, CP, Localidad, Provincia, Pais, Telefono, email, Empresa
PersistenciaUsuario	Público	Graba en una base de datos los datos correspondientes a un usuario.
	Parámetros	Nombre, Clave, Domicilio, CP, Localidad, Provincia, Pais, Telefono, email, Empresa
	Resultado	CodigoRetorno

Tabla DSI 4.12: Descripción de los métodos de la clase Usuario.

d) SOM:

<b>Método</b>	<b>Tipo</b>	<b>Descripción</b>
EjecutarSOM	Público	Ejecuta los pasos del algoritmo SOM.
	Parámetros	DNI, IdProyecto, IdVersion, CantidadCiclosEntrenamiento, CantidadObservaciones, CantidadVariable, CantidadNeuronasMapa
	Resultado	CodigoRetorno
LeerSOM	Público	Trae de base de datos los datos correspondientes a un objeto SOM.
	Parámetros	DNI, IdProyecto, IdVersion
	Resultado	CantidadCiclosEntrenamiento, CantidadObservaciones, CantidadVariable, CantidadNeuronasMapa
PersistenciaSOM	Público	Graba en una base de datos los datos correspondientes a un objeto SOM.
	Parámetros	DNI, IdProyecto, IdVersion, CantidadCiclosEntrenamiento, CantidadObservaciones, CantidadVariable, CantidadNeuronasMapa
	Resultado	CodigoRetorno

Tabla DSI 4.13: Descripción de los métodos de la clase SOM.

e) C5:

<b>Método</b>	<b>Tipo</b>	<b>Descripción</b>
EjecutarC5	Público	Ejecuta los pasos del algoritmo C5.
	Parámetros	DNI, IdProyecto, IdVersion, LimpiezaAtributos, Reglas, OrdUtilidad, OrdUtilidadBandas, Boost, BoostEj, SubcjoValores, UsarEj, UsarEjPorc, LockEj, ValidacionCruzada, ValidacionCruzadaHojas, IgnorarArchivoCostos, FuzzyThresholds, PodaGlobal, PodaCF, MinimoCasos

Método	Tipo	Descripción
	Resultado	CodigoRetorno
LeerC5	Público	Trae de base de datos los datos correspondientes a un objeto C5.
	Parámetros	DNI, IdProyecto, IdVersion
	Resultado	LimpiezaAtributos, Reglas, OrdUtilidad, OrdUtilidadBandas, Boost, BoostEj, SubcjoValores, UsarEj, UsarEjPorc, LockEj, ValidacionCruzada, ValidacionCruzadaHojas, IgnorarArchivoCostos, FuzzyThresholds, PodaGlobal, PodaCF, MinimoCasos
PersistenciaC5	Público	Graba en una base de datos los datos correspondientes a un objeto C5.
	Parámetros	DNI, IdProyecto, IdVersion, LimpiezaAtributos, Reglas, OrdUtilidad, OrdUtilidadBandas, Boost, BoostEj, SubcjoValores, UsarEj, UsarEjPorc, LockEj, ValidacionCruzada, ValidacionCruzadaHojas, IgnorarArchivoCostos, FuzzyThresholds, PodaGlobal, PodaCF, MinimoCasos
	Resultado	CodigoRetorno

Tabla DSI 4.14: Descripción de los métodos de la clase C5.

## f) Entorno:

Método	Tipo	Descripción
Destroy	Público	Se ejecuta cuando se libera el objeto y se encarga de llamar al método que Reset.
	Parámetros	
	Resultado	
Init	Público	Se ejecuta cuando se crea el objeto y se encarga de llamar al método Set.
	Parámetros	
	Resultado	
Reset	Público	Carga las variables de entorno de la aplicación en sus valores originales.

<b>Método</b>	<b>Tipo</b>	<b>Descripción</b>
	Parámetros	
	Resultado	
Set	Público	Carga las variables de entorno de la aplicación en un valor deseado y guarda los originales de las mismas.
	Parámetros	
	Resultado	

Tabla DSI 4.15: Descripción de los métodos de la clase Entorno.

g) **Form\_base:**

<b>Método</b>	<b>Tipo</b>	<b>Descripción</b>
Destroy	Público	Se ejecuta cuando se libera el objeto y se encarga de liberar elementos de los formularios de la aplicación.
	Parámetros	
	Resultado	
Init	Público	Se ejecuta cuando se crea el objeto y se encarga de armar los elementos básicos de los formularios de la aplicación.
	Parámetros	
	Resultado	
addToMenu	Público	Carga el nombre del formulario en cuestión en la opción de menú "Ventana", para ser accedido una vez creado el mismo.
	Parámetros	
	Resultado	
removeFromMenu	Público	Quita el nombre del formulario en cuestión de la opción de menú "Ventana".
	Parámetros	
	Resultado	

Tabla DSI 4.16: Descripción de los métodos de la clase Form\_base.

#### 4.4.3. Diagrama de Clases de Diseño

Antes de desarrollar el diagrama de clases de diseño, es necesario mencionar el tema de paquetes. Los paquetes son una forma de agrupar clases. Si bien físicamente no tienen por que conservar paralelismo alguno con cuestiones físicas de librería, sirven para agrupar clases en base a criterios de cohesión o acoplamiento conceptual. Para el presente proyecto, y dado que este es el diseño detallado, se divide el sistema en tres paquetes principales, los cuales se describen en la Tabla DSI 4.17:

<b>Paquete</b>	<b>Descripción</b>
Clases de Dominio	Estas clases dominan fuertemente el dominio del negocio. Cada clase se corresponde con un concepto del negocio que es necesario modelar. Las relaciones entre estas clases, modelan las relaciones, restricciones y reglas del negocio.
Clases de Proceso	Estas clases modelan procesos dentro del dominio. Por lo general, estos procesos son de una importancia o complejidad tal que vale la pena su diseño e implementación fuera de las clases de dominio.
Clases de Interfaz	Estas clases modelan la interfaz gráfica de usuario. Este tipo de clases están altamente ligadas con las características del lenguaje de implementación del sistema.

Tabla DSI 4.17: Paquetes principales del sistema

A continuación, en la figura DSI 4.18, se detalla en nomenclatura UML la relación de los paquetes descritos en la Tabla DSI 4.15:

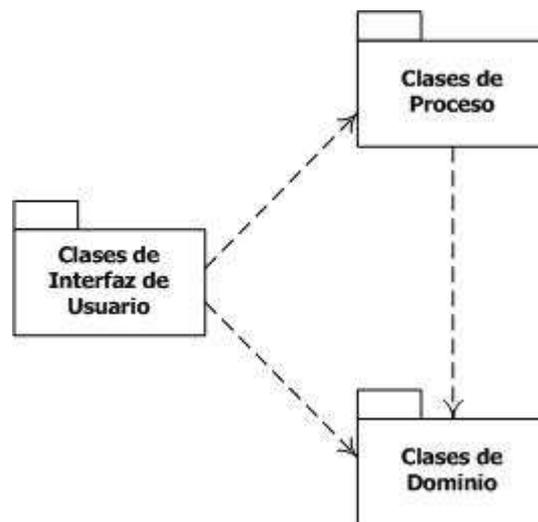
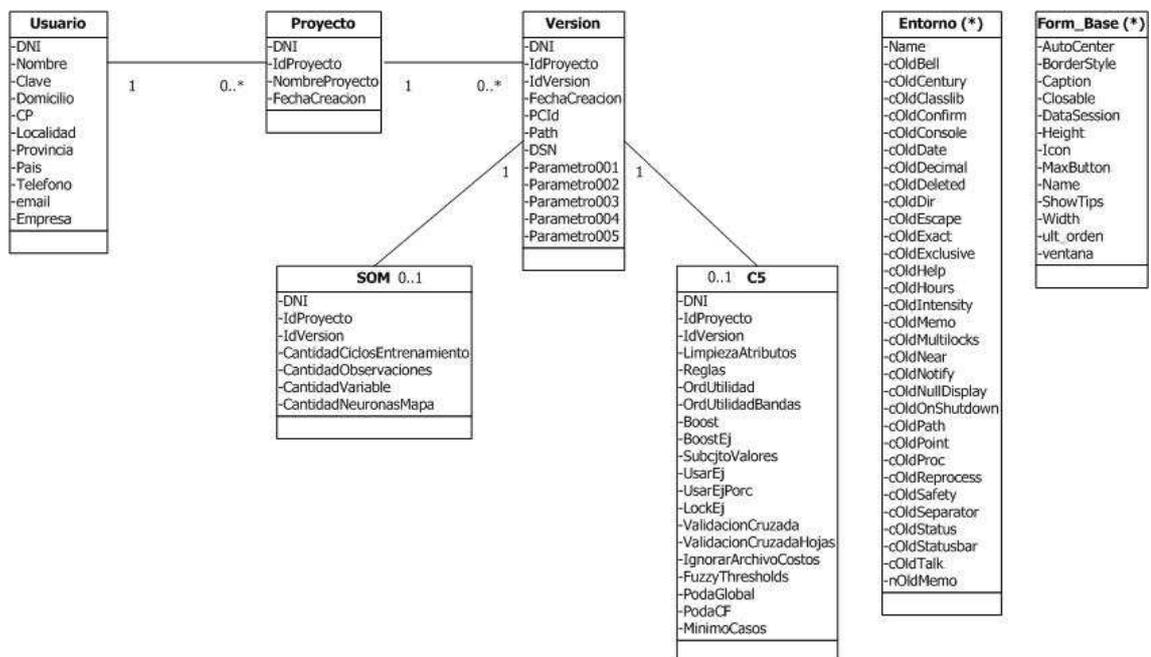


Figura DSI 4.18: Relación entre paquetes del sistema

A continuación; en las figuras DSI 4.19, DSI 4.20 y DSI 4.21; se describen los diagramas de clases correspondientes a cada uno de los paquetes:

a) Diagrama de clases para las clases de dominio:



(\*) no están conectadas porque pertenecen solo al diseño del sistema.

Figura DSI 4.19: Diagrama de clases del paquete de dominio

b) Diagrama de clases para las clases de proceso:

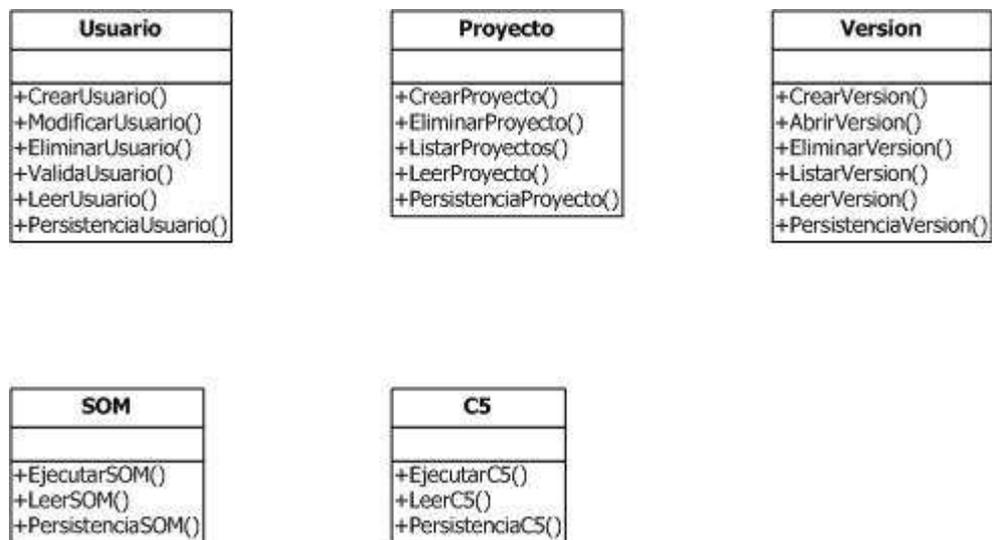


Figura DSI 4.20: Diagrama de clases del paquete de proceso

c) Clases de interfaz. Las clases de interfaz de usuario serán implementadas a través de clases visuales de formularios de Visual FoxPro.

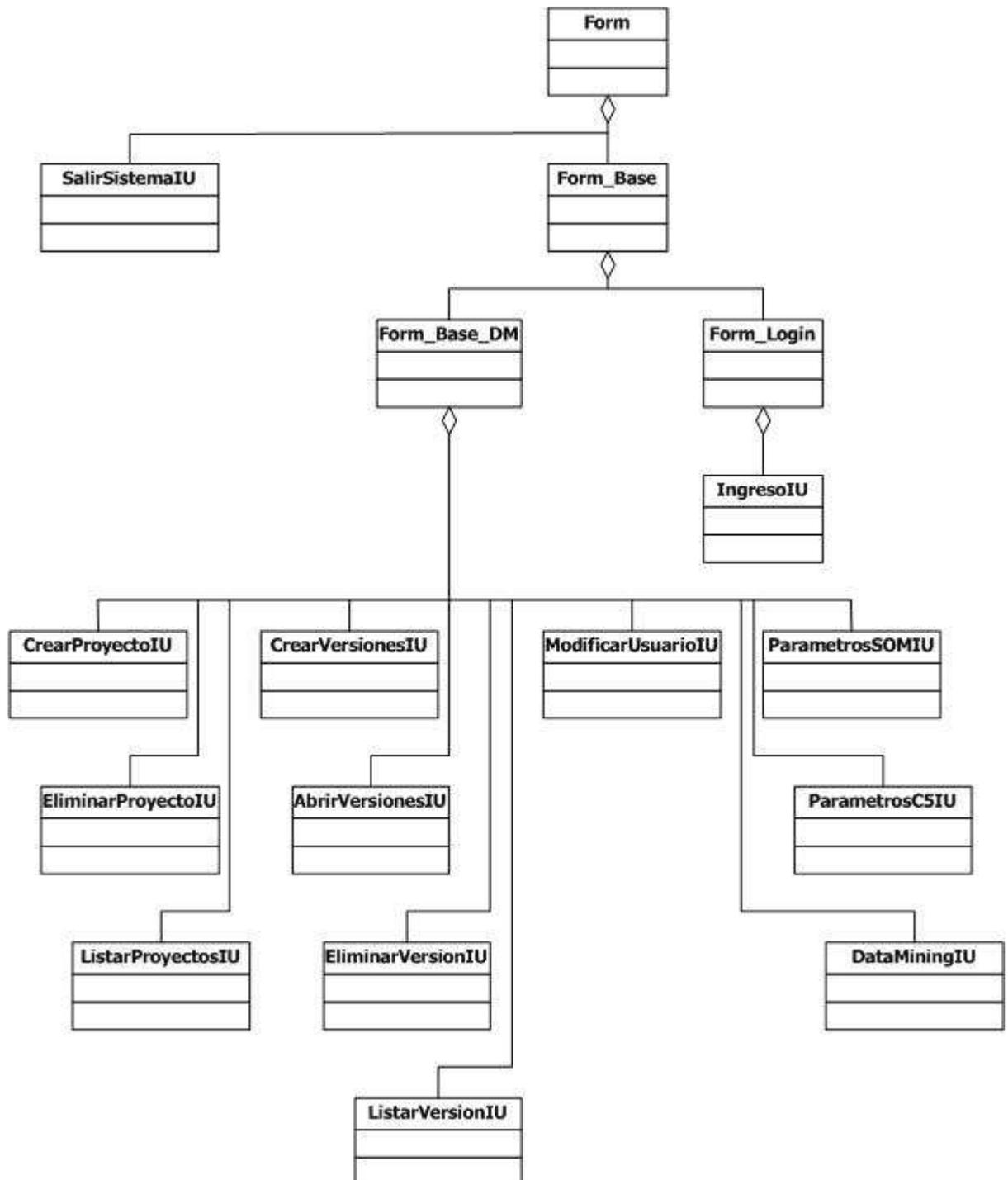


Figura DSI 4.21: Diagrama de clases del paquete de Interfaz de Usuario

#### **4.4.4. Especificación de Necesidades de Migración y Carga Inicial de Datos**

En esta tarea se realiza una primera especificación de las necesidades de migración o carga inicial de los datos requeridos por el sistema, que se completa en la actividad Diseño de la Migración y Carga Inicial de Datos.

##### **4.4.4.1. Carga Inicial de Datos**

Para el presente proyecto no es necesario plantear un esquema de migración dado, ya que no se prevé que el actual sistema tome datos de otros sistemas existentes en las organizaciones en las cuales se implemente.

#### **4.5. Diseño Físico de Datos**

En esta actividad se define la estructura física de datos que utilizará el sistema, a partir del modelo lógico de datos normalizado o modelo de clases, de manera que teniendo presentes las características específicas del sistema de gestión de datos concreto a utilizar, los requisitos establecidos para el sistema de información, y las particularidades del entorno tecnológico, se consiga una mayor eficiencia en el tratamiento de los datos.

También se analizan los caminos de acceso a los datos utilizados por cada módulo/clase del sistema en consultas y actualizaciones, con el fin de mejorar los tiempos de respuesta y optimizar los recursos de máquina.

Las tareas de esta actividad se realizan de forma iterativa y en paralelo con las realizadas en las actividades Definición de la Arquitectura del Sistema, dónde se especifican los detalles de arquitectura e infraestructura y la planificación de capacidades, Diseño de la Arquitectura de Soporte, dónde se determinan y diseñan los servicios comunes que pueden estar relacionados con la gestión de datos (acceso a bases de datos, ficheros, áreas temporales, sincronización de bases de datos, etc.), Diseño de Casos de Uso Reales y de Clases, para desarrollo orientado a objetos, y Diseño de la Arquitectura de Módulos del Sistema, para desarrollo estructurado, dónde se especifica

la lógica de tratamiento y las interfaces utilizadas.

En el caso de diseño orientado a objetos, esta actividad también es necesaria. La obtención del modelo físico de datos se realiza aplicando una serie de reglas de transformación a cada elemento del modelo de clases que se está generando en la actividad Diseño de Clases.

Asimismo, en esta actividad hay que considerar los estándares y normas establecidos para el diseño aplicando, cuando proceda, los mecanismos genéricos de diseño identificados en la tarea Identificación de Mecanismos Genéricos de Diseño.

#### **4.5.1. Diseño del Modelo Físico de Datos**

El objetivo de esta tarea es realizar el diseño del modelo físico de datos a partir del modelo lógico de datos normalizado o del modelo de clases, en el caso de diseño orientado a objetos.

Como paso previo al diseño de la estructura física de datos, se analizan las peculiaridades técnicas del gestor de bases de datos o sistema de ficheros a utilizar, y las estimaciones sobre la utilización y volumen de las ocurrencias de cada entidad / clase del modelo lógico de datos normalizado o modelo de clases. Además, si se ha establecido la necesidad de llevar a cabo una migración de datos, se deben tener en cuenta también los volúmenes de las estructuras de datos implicadas en la conversión. Esta información sirve para decidir la mejor implementación del modelo lógico de datos/modelo de clases, así como para hacer una estimación del espacio de almacenamiento.

De acuerdo al análisis anterior, se determina cómo se van a convertir las entidades/clases en tablas, considerando las relaciones existentes entre ellas y los identificadores, definiendo sus claves primarias, ajenas, alternativas u otros medios de acceso en general.

También se definen aquellos elementos que, en función del gestor o sistemas de ficheros a utilizar, se considere necesario implementar. Entre estos elementos podemos citar los siguientes:

- Bloqueo y comprensión de datos.
- Agrupamientos (cluster).
- Punteros.
- Otros.

#### 4.5.1.1. Diseño del modelo físico de datos

En primer lugar, para facilitar el entendimiento de como se vinculan las distintas tablas del sistema, en la figura DSI 4.22, se describe el diagrama de entidad relación:

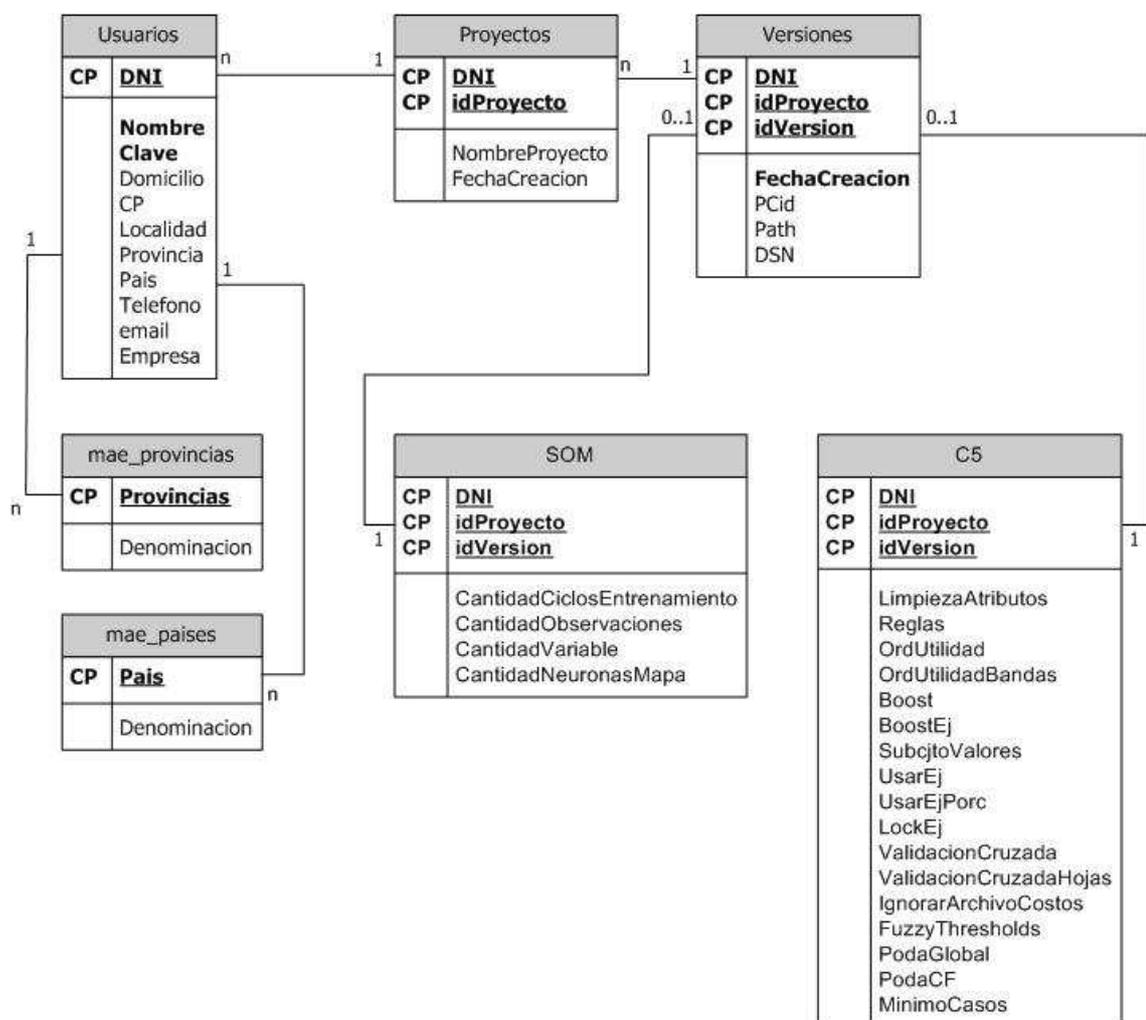


Figura DSI 4.22: Diagrama de entidad relación del proyecto

A continuación, en la Tabla DSI 4.18, se describe en detalle cada una de las tablas descriptas en la gráfica de la figura 26:

<b>Tabla</b>	<b>Columna</b>	<b>Tipo</b>	<b>Condicionante</b>
<b>C5</b>	DNI	Entero	
	idProyecto	Entero	
	idVersion	Entero	
	LimpiezaAtributos	Bit	
	Reglas	Bit	
	OrdUtilidad	Bit	
	OrdUtilidadBandas	Entero	
	Boost	Bit	
	BoostEj	Entero	
	SubejtoValores	Bit	
	UsarEj	Bit	
	UsarEjPorc	Entero	
	LockEj	Bit	
	ValidacionCruzada	Bit	
	ValidacionCruzadaHojas	Int	
	IgnorarArchivoCostos	Bit	
FuzzyThresholds	Bit		
PodaGlobal	Bit		
PodaCF	Entero		
MinimoCasos	Entero		
<b>Mae_paises</b>	Pais	Carácter (2)	
	Denominación	Carácter (55)	
<b>Mae_provincias</b>	Provincias	Entero	
	Denominación	Carácter (50)	
<b>Proyectos</b>	DNI	Entero	
	IdProyecto	Entero	
	NombreProyecto	Carácter (40)	
	FechaCreacion	Fecha	
<b>SOM</b>	DNI	Entero	
	idProyecto	Entero	
	idVersion	Entero	
	CantidadCiclosEntrenamiento	Entero	

Tabla	Columna	Tipo	Condicionante
	CantidadObservaciones	Entero	
	CantidadVariable	Entero	
	CantidadNeuronasMapa	Entero	
<b>Usuarios</b>	DNI	Entero	
	Nombre	Carácter (40)	
	Clave	Carácter (10)	
	Domicilio	Carácter (40)	
	CP	Carácter (8)	
	Localidad	Carácter (40)	
	Provincia	Entero	
	Pais	Carácter (2)	
	Telefono	Carácter (20)	
	email	Carácter (30)	
	Empresa	Carácter (40)	
<b>Versiones</b>	DNI	Entero	
	idProyecto	Entero	
	idVersion	Entero	
	FechaCreacion	Fecha	
	PCid	Carácter (20)	
	Path	Carácter (10)	
	DSN	Carácter (10)	
	Parametro001	Carácter (10)	
	Parametro002	Carácter (10)	
	Parametro003	Carácter (10)	
	Parametro004	Carácter (10)	
	Parametro005	Carácter (10)	

Tabla DSI 4.18: Descripción de las tablas del sistema

#### 4.6. Verificación y Aceptación de la Arquitectura del Sistema

El objetivo de esta actividad es garantizar la calidad de las especificaciones del diseño del sistema de información y la viabilidad del mismo, como paso previo a la generación de las especificaciones de construcción.

Para cumplir dicho objetivo, se llevan a cabo las siguientes acciones:

- Verificación de la calidad técnica de cada modelo o especificación
- Aseguramiento de la coherencia entre los distintos modelos
- Aceptación del diseño de la arquitectura por parte de Explotación y Sistemas.

#### **4.6.1. Verificación de la Especificación de Diseño**

El objetivo de esta tarea es asegurar la calidad formal de los distintos modelos, conforme a la técnica seguida para la elaboración de cada producto y a las normas y estándares especificados en el catálogo de normas.

##### **4.6.1.1. Resultado de la Verificación de la Especificación de Diseño**

Se han verificado los distintos modelos generados juntamente con la directora de Tesis y se han aplicado las modificaciones correspondientes.

#### **4.6.2. Análisis de Consistencia de las Especificaciones de Diseño**

El objetivo de esta tarea es asegurar que las especificaciones del diseño son coherentes entre sí, comprobando la falta de ambigüedades o duplicación de información. Esta consistencia se asegura entre especificaciones de diseño, y con respecto a los modelos del análisis.

Las diferentes comprobaciones se fundamentan generalmente en técnicas matriciales o de revisión entre los elementos comunes de los distintos modelos.

El análisis de consistencia relativo a la arquitectura del sistema es común para desarrollo estructurado y orientado a objetos, aunque respecto a los productos del diseño detallado es específico para cada uno de los enfoques.

#### 4.6.2.1. Consistencia de las Especificaciones de Diseño

Se comprueba la coherencia entre los distintos modelos de acuerdo a las trazabilidades que se presentan en la figura DSI 4.23:

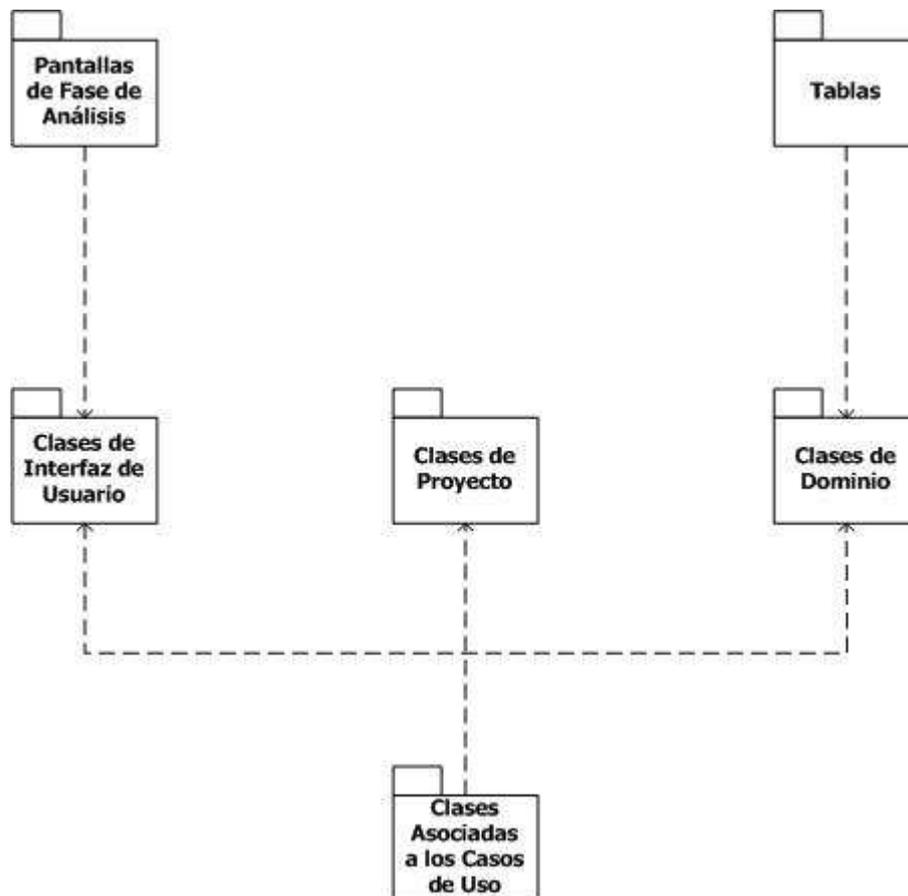


Figura DSI 4.23: Diagrama de consistencia del proyecto

#### **Clases Asociadas a los Casos de Uso vs. Clases de Interfaz de Usuario / Clases de Proceso / Clases de Dominio**

La Matriz de la Tabla DSI 4.19 muestra dos columnas, en la de la izquierda, se describen las clases de los paquetes de Clases de Interfaz de Usuario / Clases de Proceso / Clases de Dominio y en la columna de la derecha las clases asociadas a los Casos de Uso, como puede observarse, existe una relación uno a uno entre ambas clases:

Clases asociadas a los paquetes de clases de Interfaz de Usuario / Clases de Procesos / Clases de Dominios	Clases asociadas a los casos de uso
AbrirVersionesIU	Abrir Versión
CrearProyectoIU	Crear Proyecto
CrearVersionesIU	Crear Versiones de un Proyecto
DataMiningIU	Ejecutar Proceso de Explotación de Datos
EliminarProyectoIU	Eliminar Proyecto
EliminarVersiónIU	Eliminar Versión
IngresoIU	Ingreso al sistema
ListarProyectosIU	Listar Proyectos
ListarVersionIU	Listar Versiones
ModificarUsuarioIU	Modificar Usuario / Incorporar Usuario
Parámetros C5IU	Establecer Parámetros de C5.0
ParámetrosSOMIU	Establecer Parámetros de SOM
	Salir del Sistema

Tabla DSI 4.19, Trazabilidades entre Clases Asociadas a los Casos de Uso versus Clases de Interfaz de Usuario / Clases de Proceso / Clases de Dominio

### **Tablas vs. Clases de Dominio**

La Matriz de la Tabla DSI 4.20 muestra en las filas las tablas y en las columnas las clases asociadas al paquete de dominio, como puede observarse, todas las clases tiene su correspondencia con la tablas:

<b>Tablas / Clases</b>	Proyecto	Version	Usuario	SOM	C5
C5					X
Mae_paises					
Mae_provincias					
Proyectos	X				
SOM				X	

Tablas / Clases	Proyecto	Version	Usuario	SOM	C5
Versiones		X			
Usuarios			X		

Tabla DSI 4.20, Trazabilidades entre tablas y las Clases de Dominio

### **Clases de Interfaz de Usuario vs. Pantalla de la fase de análisis**

La Matriz de la Tabla DSI 4.21 muestra dos columnas, en la de la izquierda, se describen las clases del paquetes de Clases de Interfaz de Usuario y en la columna de la derecha las pantallas de la fase de análisis, como puede observarse, existe una relación uno a uno entre ambos componentes:

Clases de Interfaz de Usuario	Pantallas de la fase de análisis
AbrirVersionesIU	Abrir versiones de un proyecto
CrearProyectoIU	Crear Proyecto Nuevo
CrearVersionesIU	Crear versiones de un proyecto
DataMiningIU	Ejecutar Datamining
EliminarProyectoIU	Eliminar Proyectos
EliminarVersiónIU	Eliminar versión de un proyecto
IngresoIU	Ingreso al sistema
ListarProyectosIU	Listar Proyectos
ListarVersionIU	Listar versiones de proyectos
ModificarUsuarioIU	Modificar / ingresar datos de usuario
Parámetros C5IU	Parámetros de C5.0
ParámetrosSOMIU	Parámetros de SOM

Tabla DSI 4.21. Trazabilidades entre las Clases de Interfaz de Usuario y las pantallas de la fase de análisis

#### **4.7. Aceptación de la Arquitectura del Sistema**

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

En una reunión mantenida entre el tesista y la Directora del proyecto se dio por aprobada la fase de Análisis del Sistema de Información.

#### **4.8. Generación de Especificaciones de Construcción**

En esta actividad se generan las especificaciones para la construcción del sistema de información, a partir del diseño detallado.

Estas especificaciones definen la construcción del sistema de información a partir de las unidades básicas de construcción (en adelante, componentes), entendiendo como tales unidades independientes y coherentes de construcción y ejecución, que se corresponden con un empaquetamiento físico de los elementos del diseño de detalle, como pueden ser módulos, clases o especificaciones de interfaz.

La división del sistema de información en subsistemas de diseño proporciona, por continuidad, una primera división en subsistemas de construcción, definiendo para cada uno de ellos los componentes que lo integran. Si se considera necesario, un subsistema de diseño se podrá dividir a su vez en sucesivos niveles para mayor claridad de las especificaciones de construcción.

Las dependencias entre subsistemas de diseño proporcionan información para establecer las dependencias entre los subsistemas de construcción y, por lo tanto, definir el orden o secuencia que se debe seguir en la construcción y en la realización de las pruebas.

También se generan las especificaciones necesarias para la creación de las estructuras de datos en los gestores de bases de datos o sistemas de ficheros.

El producto resultante de esta actividad es el conjunto de las especificaciones de construcción del sistema de información, que comprende:

- Especificación del entorno de construcción.
- Descripción de subsistemas de construcción y dependencias.
- Descripción de componentes.
- Plan de integración del sistema de información.
- Especificación detallada de componentes.
- Especificación de la estructura física de datos.

#### **4.8.1. Especificación del Entorno de Construcción**

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

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

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

A continuación, en la Tabla DSI 4.22, se describen las especificaciones del entorno tecnológico:

<b>Concepto</b>	<b>Definición</b>
Entorno tecnológico: Hardware, software y comunicaciones	El equipo de desarrollo será un PC – Intel 2,8 Ghz, con 256 Mb de memoria RAM y un disco rígido de 80GB. El sistema operativo es Windows XP. La base de datos es Microsoft SQL. Server 2000
Herramientas de construcción, generadores de código, compiladores, etc.	Microsoft Visual FoxPro 9.0
Restricciones técnicas del entorno	No se observan
Planificación de capacidades previstas, o la información que estime oportuno el departamento de sistemas para efectuar dicha planificación	No se observan
Requisitos de operación y seguridad del entorno de construcción	No se observan

Tabla DSI 4.22. Especificaciones del entorno tecnológico

#### **4.8.2. Definición de Componentes y Subsistemas de Construcción**

La especificación de los subsistemas de construcción se realiza a partir de los subsistemas de diseño, con una continuidad directa, permitiéndose a su vez un mayor nivel de detalle agrupando componentes en subsistemas dentro de un subsistema de construcción.

Los componentes se definen mediante la agrupación de elementos del diseño de detalle de cada subsistema de diseño. En principio, cada módulo o clase y cada formato individual de interfaz se corresponden con un componente, aunque se pueden agrupar o redistribuir módulos o clases en componentes, siguiendo otros criterios más oportunos, como pueden ser:

- Optimización de recursos.
- Características comunes de funcionalidad o de acceso a datos.
- Necesidades especiales de ejecución: elementos críticos, accesos costosos a datos, etc.

Los subsistemas de construcción y las dependencias entre subsistemas y entre componentes de un subsistema recogen aspectos prácticos relativos a la plataforma concreta de construcción y ejecución. Entre estos aspectos se pueden citar, por ejemplo:

- Secuencia de compilación entre componentes.
- Agrupación de elementos en librerías o packages (por ejemplo, DLL en el entorno Windows, packages en Java).

La asignación de subsistemas de construcción a nodos, por continuidad con el diseño, determina la distribución de los componentes que lo integran.

Opcionalmente, se propone la realización de un plan de integración del sistema de información, especificando la secuencia y organización de la construcción y prueba de los subsistemas de construcción y de los componentes, desde un punto de vista técnico.

#### 4.8.2.1. Componentes y Subsistemas de Construcción

Para el presente trabajo, se ha decidido dejar las clases tal y como se han diseñado. De esta forma cada clase que compone el diseño se encontrará representada por una clase en el dominio de la implementación y viceversa. A continuación, en la figura DSI 4.24, se muestra el pertinente diagrama UML:

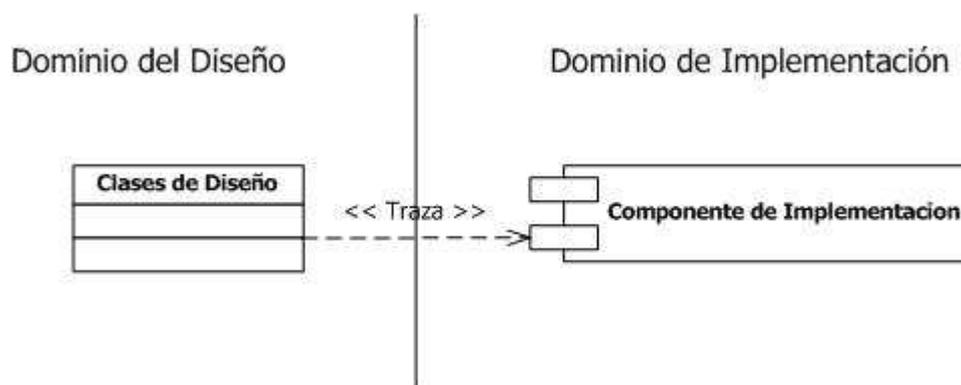


Figura DSI 4.24: Relación entre los dominios de diseño y implementación

#### 4.9. Especificación Técnica del Plan de Pruebas

En esta actividad se realiza la especificación de detalle del plan de pruebas del sistema de información para cada uno de los niveles de prueba establecidos en el proceso Análisis del Sistema de Información:

- Pruebas unitarias.
- Pruebas de integración.
- Pruebas del sistema.
- Pruebas de implantación.
- Pruebas de aceptación.

Para ello se toma como referencia el plan de pruebas, que recoge los objetivos de la prueba de un sistema, establece y coordina una estrategia de trabajo, y provee del marco adecuado para planificar paso a paso las actividades de prueba. También puede ser una referencia el plan de integración del sistema de información propuesto, opcionalmente, en la tarea Definición de Componentes y Subsistemas de Construcción.

El catálogo de requisitos, el catálogo de excepciones y el diseño detallado del sistema de información, permiten la definición de las verificaciones que deben realizarse en cada nivel de prueba para comprobar que el sistema responde a los requisitos planteados. La asociación de las distintas verificaciones a componentes, grupos de componentes y subsistemas, o al sistema de información completo, determina las distintas verificaciones de cada nivel de prueba establecido.

Las pruebas unitarias comprenden las verificaciones asociadas a cada componente del sistema de información. Su realización tiene como objetivo verificar la funcionalidad y estructura de cada componente individual.

Las pruebas de integración comprenden verificaciones asociadas a grupos de componentes, generalmente reflejados en la definición de subsistemas de construcción o en el plan de integración del sistema de información. Tienen por objetivo verificar el

correcto ensamblaje entre los distintos componentes.

Las pruebas del sistema, de implantación y de aceptación corresponden a verificaciones asociadas al sistema de información, y reflejan distintos propósitos en cada tipo de prueba:

- Las pruebas del sistema son pruebas de integración del sistema de información completo. Permiten probar el sistema en su conjunto y con otros sistemas con los que se relaciona para verificar que las especificaciones funcionales y técnicas se cumplen.
- Las pruebas de implantación incluyen las verificaciones necesarias para asegurar que el sistema funcionará correctamente en el entorno de operación al responder satisfactoriamente a los requisitos de rendimiento, seguridad y operación, y coexistencia con el resto de los sistemas de la instalación, y conseguir la aceptación del sistema por parte del usuario de operación.
- Las pruebas de aceptación van dirigidas a validar que el sistema cumple los requisitos de funcionamiento esperado, recogidos en el catálogo de requisitos y en los criterios de aceptación del sistema de información, y conseguir la aceptación final del sistema por parte del usuario.

Las pruebas unitarias, de integración y del sistema se llevan a cabo en el proceso Construcción del Sistema de Información (CSI), mientras que las pruebas de implantación y aceptación se realizan en el proceso Implantación y Aceptación del Sistema (IAS).

Como resultado de esta actividad se actualiza el plan de pruebas con la información siguiente:

- Especificación del entorno de pruebas.
- Especificación técnica de niveles de prueba.
- Planificación de las pruebas.

### **4.9.1. Especificación del Entorno de Pruebas**

El objetivo de esta tarea es la definición detallada y completa del entorno necesario para la realización de las pruebas del sistema: unitarias, de integración, de implantación y de aceptación.

Se propone considerar los siguientes conceptos en la especificación del entorno:

- Entorno tecnológico: hardware, software y comunicaciones.
- Restricciones técnicas del entorno.
- Requisitos de operación y seguridad del entorno de pruebas.
- Herramientas de prueba relacionadas con la extracción de juegos de ensayo, análisis de resultados, utilidades de gestión del entorno, etc.
- Planificación de capacidades previstas, o la información que estime oportuno el departamento técnico para efectuar dicha planificación.
- Procedimientos de promoción de elementos entre entornos (desarrollo, pruebas, explotación, etc.).
- Procedimientos de emergencia y de recuperación, así como de vuelta atrás.

#### **4.9.1.1. Entorno de Pruebas**

Para la realizar los casos de pruebas, no se requerirá especificar nuevos elementos de equipamiento, tanto a nivel de hardware como de software, a los ya explicados en las fases anteriores. A continuación se describe cual será el mecanismo de promoción de elementos entre entornos y procedimientos de emergencia y recuperación en caso de fallo:

- a) Procedimientos de promoción: El sistema, para sus pruebas, será instalado en un equipo donde se cuente con las librerías de Microsoft Visual FoxPro. De esta forma, una probado y aprobado, el sistema será resguardado para su puesta en producción.

b) Procedimientos de emergencia y de recuperación, así como de vuelta atrás: Se define como emergencia en la que haga falta una recuperación del sistema, a aquel caso en el que el servidor se ve dañado físicamente y por ende genera el mal funcionamiento de la aplicación. En este caso deberá ser necesario recuperar el sistema con el siguiente curso de acción:

1. Tomar los archivos de instalación del sistema.
2. Proceder a instalar el sistema en un servidor o bien en el servidor reparado.
3. Recuperar la copia de resguardo (backup) de la base de datos. Este punto varía de acuerdo a la base de datos en cuestión. Para el caso del servidor Microsoft SQL Server 2000, esta recuperación consiste en reemplazar el archivo de datos con la versión del backup.
4. Iniciar el sistema

#### **4.9.2. Revisión de la Planificación de Pruebas**

En esta tarea se completa y especifica la planificación de las pruebas, determinando los distintos perfiles implicados en la preparación y ejecución de las pruebas y en la evaluación de los resultados, así como el tiempo estimado para la realización de cada uno de los niveles de prueba, de acuerdo a la estrategia de integración establecida.

##### **4.9.2.1. Planificación de Pruebas**

Teniendo en cuenta que el sistema ha sido desarrollado con un enfoque de casos de uso, se genera un plan de pruebas que apunte a probar funcionalmente el sistema desde el punto de vista de los casos de uso, componentes de infraestructura y pruebas globales del sistema. Los tipos de prueba a realizar son:

- Pruebas Unitarias: Prueban componentes específicos del sistema. Se preparará un plan de pruebas para el componente de comunicación.

- Pruebas de Integración: Abarca las pruebas por casos de uso (ya integrados al componente de comunicaciones y ejecutándose contra el servidor).
- Pruebas de Sistema: las pruebas serán ejecutadas y obtenidos sus resultados utilizado al propio sistema como herramienta y probando el circuito total.

#### **4.9.2.2. Pruebas Unitarias**

El único componente a probar de forma independiente es el componente de comunicaciones. Las pruebas tienen como objetivo la ejecución de una transacción específica y evaluar el funcionamiento del componente tal de una transacción y evaluar el funcionamiento del componente como de detalla en su diseño y en el catálogo de requisitos y excepciones. Para ello, se procederá desde la ventana de comandos de la herramienta Visual FoxPro 9.0 a probar la autenticación de usuario. El programa recibirá por línea de comando un par de usuario-clave y ejecutará la transacción de autenticación en el servidor. Como resultado mostrará por consola un entero indicando el estado de la conexión.

Además, desde el SQL se ingresará a la base de datos un usuario con los siguientes datos:

- DNI = 22336911
- Nombre = Sal Esteban
- Clave = 123123
- Domicilio = Uriburu 1921
- CP = C11126ABC
- Localidad = Ciudad de Buenos Aires
- Provincia = Ciudad de Buenos Aires
- Pais = Argentina

- Telefono = 4311-4117
- email = esteban.sal@gmail.com
- Empresa = Desarrollo SA

A continuación, en la tabla DSI 4.23, se describen en detalle los casos de prueba a realizar:

<b>CP-0001</b>	
Objetivo	Probar una autenticación exitosa
Entrada	22336911 123123
Salida	1
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ubicarse en la pantalla de comandos del producto Visual FoxPro 9.0 2- Ejecutar la instrucción: crear objeto obj (CreateObject('usuario')) y ejecutar el método obj.conectar('usuario','clave') 3- El resultado de la conexión se almacenará en la propiedad obj.conexion y si es un entero positivo la misma se realizó con éxito
Prerrequisitos	Levantar el servidor de base de datos

Tabla DSI 4.23: Caso de Pruebas CP-0001

#### 4.9.2.3. Pruebas de Integración

Las pruebas de integración tienen como objetivo encontrar fallas en el funcionamiento de los componentes y subsistemas del sistema, al funcionar en conjunto para proveer la funcionalidad deseada.

Dada la topología y funcionalidad del presente desarrollo se harán las pruebas necesarias dentro de las pruebas del sistema. Esta definición se basa en la siguiente premisa:

- Los componentes de funcionalidad clientes no pueden ser probados sin el uso del componente servidor y los medios de comunicación (ya probados en el apartado anterior).

#### 4.9.2.4. Pruebas del Sistema

Las pruebas de integración tienen como objetivo probar cada uno de los casos de uso implementados en la aplicación.

El proceso de pruebas se iniciará con una instalación nueva del producto, es decir sin ningún usuario ni proyecto cargado, tal como si una analista del negocio recibiera el soporte con el software para instalarlo en su PC. Asimismo el ingreso de los casos de prueba se hace en el orden en que se detalla en la Tabla DSI 4.24.

<b>Código</b>	<b>Caso de Uso</b>	<b>Objetivo</b>
CP-0002	Validar Usuario	Probar un ingreso exitoso al sistema
CP-0003	Validar Usuario	Intentar ingresar al sistema con un usuario inexistente
CP-0004	Validar Usuario	Intentar ingresar al sistema con una clave errónea
CP-0005	Cambiar Clave	Cambiar exitosamente la clave de acceso de un usuario
CP-0006	Cambiar Clave	Intentar cambiar erróneamente la clave de acceso al sistema (cargar una clave anterior incorrecta)
CP-0007	Cambiar Clave	Intentar cambiar erróneamente la clave de acceso al sistema (cargar erróneamente la verificación de la nueva clave)
CP-0008	Incorporar Usuario	Ingresar correctamente un nuevo usuario
CP-0009	Incorporar Usuario	Intentar ingresar un usuario existente
CP-0010	Modificar Usuario	Modificar correctamente los datos de un usuario
CP-0011	Crear Proyecto	Ingresar un nuevo Proyecto
CP-0012	Crear Proyecto	Intentar ingresar un proyecto con nombre ya existente
CP-0013	Crear Proyecto	Intentar ingresar un proyecto sin nombre

<b>Código</b>	<b>Caso de Uso</b>	<b>Objetivo</b>
CP-0014 – CP-0016	Crear Proyecto	Para poder desarrollar los siguientes casos de pruebas, se debe ingresar tres nuevos proyectos
CP-0017	Eliminar Proyecto	Eliminar correctamente un proyecto
CP-0018	Listar Proyecto	Mostrar todos los proyectos de un usuario
CP-0019	Crear Versión	Crear una nueva versión de un proyecto
CP-0020	Abrir Versión	Abrir una versión de un proyecto
CP-0021	Listar Versión	Mostrar todas las versiones de proyectos de un usuario
CP-0022	Establecer Parámetros de SOM	Ingresar parámetros permitidos para el algoritmo SOM
CP-0023	Establecer Parámetros de SOM	Ingresar parámetros no permitidos para el algoritmo SOM
CP-0024	Establecer Parámetros de C5.0	Ingresar parámetros permitidos para el algoritmo C5.0
CP-0025	Establecer Parámetros de C5.0	Ingresar parámetros no permitidos para el algoritmo C5.0
CP-0026	Ejecutar Proceso de Explotación de Datos	Intentar ejecutar los dos algoritmos en una versión de un proyecto
CP-0027	Ejecutar Proceso de Explotación de Datos	Intentar ejecutar el algoritmo SOM en una versión de un proyecto
CP-0028	Ejecutar Proceso de Explotación de Datos	Intentar ejecutar el algoritmo C5.0 en una versión de un proyecto
CP-0029	Ejecutar Proceso de Explotación de Datos	Intentar ejecutar sin haber seleccionado un algoritmo

Tabla DSI 4.24: Secuencia de ingreso de casos de Prueba

A continuación se descripción de los casos de prueba asociados a los casos de uso, según el orden definido en la Tabla DSI 4.24:

a) Casos de prueba asociados al caso de uso Validar Usuario (ver tablas DSI 4.25 a DSI 4.27):

<b>CP-0002</b>	
Objetivo	Probar un ingreso exitoso al sistema
Entrada	Usuario = 22336911

<b>CP-0002</b>	
	Clave = 123456
Salida	Menú principal del sistema
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ejecutar dm.exe 2- Cuando aparezca la pantalla de ingreso teclear: Usuario = 22336911 Clave = 123456 3- Presionar el botón “Aceptar”
Prerrequisitos	Levantar el servidor de Base de Datos

Tabla DSI 4.25: Probar un ingreso exitoso al sistema

<b>CP-0003</b>	
Objetivo	Intentar ingresar al sistema con un usuario inexistente
Entrada	Usuario = 22336912 Clave = 123456
Salida	Mensaje de usuario o clave inválido
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ejecutar dm.exe 2- Cuando aparezca la pantalla de ingreso teclear: Usuario = 22336912 Clave = 123456 3- Presionar el botón “Aceptar”
Prerrequisitos	Levantar el servidor de Base de Datos

Tabla DSI 4.26: Intentar ingresar al sistema con un usuario inexistente

<b>CP-0004</b>	
Objetivo	Intentar ingresar al sistema con una clave errónea
Entrada	Usuario = 22.336.911 Clave = 123457
Salida	Mensaje de usuario o clave inválido
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ejecutar dm.exe 2- Cuando aparezca la pantalla de ingreso teclear: Usuario = 22336911 Clave = 123457 3- Presionar el botón “Aceptar”
Prerrequisitos	Levantar el servidor de Base de Datos

Tabla DSI 4.27: Intentar ingresar al sistema con una clave errónea

b) Casos de prueba asociados al caso de uso Cambiar Clave (ver tablas DSI 4.28 a DSI 4.30):

<b>CP-0005</b>	
Objetivo	Cambiar exitosamente la clave de acceso de un usuario (desde la opción del menú “Detalles de Usuario”)
Entrada	Clave actual = 123456 Clave nueva = 123123
Salida	Se ha modificado la clave
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Seleccionar la opción del menú “Detalles de Usuario”. 2- Cuando aparezca la pantalla, en los campos destinados a claves ingresar: Clave Actual = 123456 Clave Nueva = 123123

<b>CP-0005</b>	
	Reingrese Clave Nueva = 123123 3- Presionar el botón “Actualizar”
Prerrequisitos	Ingresar al sistema, ingresando usuario = esteban y clave = 123456

Tabla DSI 4.28: Cambiar exitosamente la clave de acceso de un usuario

<b>CP-0006</b>	
Objetivo	Intentar cambiar erróneamente la clave de acceso al sistema (ingresar un valor de clave anterior incorrecto)
Entrada	Clave actual = 123456 Ingreso Clave nueva = 123789 Reingreso Clave nueva = 123789
Salida	Clave actual incorrecta
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Seleccionar la opción del menú “Detalles de Usuario”. 2- Cuando aparezca la pantalla, en los campos destinados a claves ingresar: Clave Actual = 123456 Clave Nueva = 123789 Reingrese Clave Nueva = 123789 3- Presionar el botón “Actualizar”
Prerrequisitos	Ingresar a la pantalla de ingreso al sistema, ingresando usuario = esteban y clave = 123123

Tabla DSI 4.29: Intentar cambiar erróneamente la clave de acceso al sistema

<b>CP-0007</b>	
Objetivo	Intentar cambiar erróneamente la clave de acceso al sistema (reingresar erróneamente la clave nueva)
Entrada	Clave actual = 123123

<b>CP-0007</b>	
	Ingreso Clave nueva = 123789 Reingreso Clave nueva = 123777
Salida	Reingreso de clave incorrecto
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Seleccionar la opción del menú “Detalles de Usuario”. 2- Cuando aparezca la pantalla, en los campos destinados a claves ingresar: Clave Actual = 123123 Clave Nueva = 123789 Reingrese Clave Nueva = 123777 3- Presionar el botón “Actualizar”
Prerrequisitos	Ingresar a la pantalla de ingreso al sistema, ingresando usuario = esteban y clave = 123123

Tabla DSI 4.30: Intentar cambiar erróneamente la clave de acceso al sistema

c) Casos de prueba asociados al caso de uso Incorporar Usuario (ver tablas DSI 4.31 y DSI 4.32):

<b>CP-0008</b>	
Objetivo	Ingresar correctamente un nuevo usuario
Entrada	DNI = 20001002 Apellido y Nombre = Benegas Alberto Domicilio = Alberdi 1854 Localidad = Ciudad de Buenos Aires CP = C1050AAA Provincia = Ciudad de Buenos Aires Teléfono = 4852-7412 e-mail = abenegas@gmail.com País = Argentina Fecha de nacimiento = 01/12/1970 Empresa = Hayeck SA
Salida	Usuario ingresado

<b>CP-0008</b>	
Objetivo	Ingresar correctamente un nuevo usuario
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	<p>1- Ingresar desde la pantalla inicial “Datos de acceso...”, la opción “Usuario nuevo”</p> <p>2- Cuando aparezca la pantalla de ingreso de usuario teclear:</p> <p>DNI = 20001002</p> <p>Apellido y Nombre = Benegas Alberto</p> <p>Domicilio = Alberdi 1854</p> <p>Localidad = Ciudad de Buenos Aires</p> <p>CP = C1050AAA</p> <p>Provincia = Ciudad de Buenos Aires</p> <p>Teléfono = 4852-7412</p> <p>e-mail = abenegas@gmail.com</p> <p>País = Argentina</p> <p>Fecha de nacimiento = 01/12/1970</p> <p>Empresa = Hayek SA</p> <p>3- Presionar el botón “Insertar”</p>
Prerrequisitos	Ingresar al sistema

Tabla DSI 4.31: Prueba de un ingreso correcto de un usuario

<b>CP-0009</b>	
Objetivo	Intentar ingresar un usuario existente
Entrada	<p>DNI = 22336911</p> <p>Apellido y Nombre = Sal Esteban</p> <p>Domicilio = Uriburu 1921</p> <p>Localidad = Ciudad de Buenos Aires</p> <p>CP = C11126ABC</p> <p>Provincia = Ciudad de Buenos Aires</p> <p>Teléfono = 4311-4117</p> <p>e-mail = esteban.sal@gmail.com</p> <p>País = Argentina</p> <p>Fecha de nacimiento = 10/03/1972</p> <p>Empresa = Desarrollo SA</p>

<b>CP-0009</b>	
Objetivo	Intentar ingresar un usuario existente
Salida	El usuario ingresado ya existe
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	<p>1- Ingresar desde la pantalla inicial “Datos de acceso...”, la opción “Usuario nuevo”</p> <p>2- Cuando aparezca la pantalla de ingreso de usuario teclear:</p> <p>DNI = 22336911</p> <p>Apellido y Nombre = Sal Esteban</p> <p>Domicilio = Uriburu 1921</p> <p>Localidad = Ciudad de Buenos Aires</p> <p>CP = C11126ABC</p> <p>Provincia = Ciudad de Buenos Aires</p> <p>Teléfono = 4311-4117</p> <p>e-mail = esteban.sal@gmail.com</p> <p>País = Argentina</p> <p>Fecha de nacimiento = 10/03/1972</p> <p>Empresa = Desarrollo SA</p> <p>3- Presionar el botón “Insertar”</p>
Prerrequisitos	Ingresar al sistema

Tabla DSI 4.32: Prueba de un ingreso de usuario ya existente

d) Casos de prueba asociados al caso de uso Modificar Usuario (ver tabla DSI 4.33):

<b>CP-0010</b>	
Objetivo	Modificar correctamente los datos de un usuario
Entrada	Domicilio = Uriburu 1921 – 6to 32
Salida	Modificación aceptada
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	<p>1- Ingresar desde menú principal al menú “Usuarios” seleccionar la opción “Modificar datos de un usuario”</p> <p>2- Cuando aparezca la pantalla de ingreso de</p>

<b>CP-0010</b>	
	usuario teclear: Domicilio = Uriburu 1921 – 6to 32 3- Presionar el botón Enter 4- Presionar el botón “Actualizar”
Prerrequisitos	Ingresar al ingresar al sistema con usuario = 22336911 y clave = 123123

Tabla DSI 4.33: Prueba de modificar correctamente los datos de un usuario

f) Casos de prueba asociados al caso de uso Crear Proyecto (ver tablas DSI 4.34 a DSI 4.39):

<b>CP-0011</b>	
Objetivo	Ingresar un nuevo proyecto
Entrada	Nombre Proyecto = prueba1
Salida	Proyecto ingresado
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “Proyectos” seleccionar la opción “Crear Proyecto” 2- Cuando aparezca la pantalla de ingreso del proyecto teclear: Nombre Proyecto = prueba1 3- Presionar el botón “Aceptar”
Prerrequisitos	Ingresar al ingresar al sistema con DNI = 22336911 y clave = 123123

Tabla DSI 4.34: Prueba de ingresar un nuevo proyecto

<b>CP-0012</b>	
Objetivo	Intentar ingresar un proyecto con un nombre ya existente
Entrada	Nombre Proyecto = prueba1

<b>CP-0012</b>	
Salida	El proyecto ya existe
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “Proyectos” seleccionar la opción “Crear Proyecto” 2- Cuando aparezca la pantalla de ingreso del proyecto teclear: Nombre Proyecto = prueba1 3- Presionar el botón “Aceptar”
Prerrequisitos	1- Levantar el servidor de aplicaciones 2- Ingresar al ingresar al sistema con DNI = 22336911 y clave = 123123

Tabla DSI 4.35: Prueba de ingresar un proyecto con un nombre ya existente

<b>CP-0013</b>	
Objetivo	Intentar ingresar un proyecto sin nombre
Entrada	Nombre Proyecto =
Salida	Falta ingresar el nombre del proyecto
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “Proyectos” seleccionar la opción “Crear Proyecto” 2- Cuando aparezca la pantalla de ingreso del proyecto teclear: Nombre Proyecto = 3- Presionar el botón “Aceptar”
Prerrequisitos	Ingresar al ingresar al sistema con DNI = 22336911 y clave = 123123

Tabla DSI 4.36: Prueba de ingresar un proyecto sin nombre

<b>CP-0014</b>	
Objetivo	Ingresar un nuevo proyecto
Entrada	Nombre Proyecto = prueba2
Salida	Proyecto ingresado
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “Proyectos” seleccionar la opción “Crear Proyecto” 2- Cuando aparezca la pantalla de ingreso del proyecto teclear: Nombre Proyecto = prueba2 3- Presionar el botón “Aceptar”
Prerrequisitos	Ingresar al ingresar al sistema con DNI = 20001002 y clave = 111111

Tabla DSI 4.37: Prueba de ingresar un nuevo proyecto

<b>CP-0015</b>	
Objetivo	Ingresar un nuevo proyecto
Entrada	Nombre Proyecto = prueba3
Salida	Proyecto ingresado
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “Proyectos” seleccionar la opción “Crear Proyecto” 2- Cuando aparezca la pantalla de ingreso del proyecto teclear: Nombre Proyecto = prueba3 3- Presionar el botón “Aceptar”
Prerrequisitos	Ingresar al ingresar al sistema con usuario = 22336911 y clave = 123123

Tabla DSI 4.38: Prueba de ingresar un nuevo proyecto

<b>CP-0017</b>	
Objetivo	Ingresar un nuevo proyecto
Entrada	Nombre Proyecto = prueba4
Salida	Proyecto ingresado
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “Proyectos” seleccionar la opción “Crear Proyecto” 2- Cuando aparezca la pantalla de ingreso del proyecto teclear: Nombre Proyecto = prueba4 3- Presionar el botón “Aceptar”
Prerrequisitos	Ingresar al ingresar al sistema con usuario = 20001002 y clave = 111111

Tabla DSI 4.39: Prueba de ingresar un nuevo proyecto

g) Casos de prueba asociados al caso de uso Eliminar Proyecto (ver tabla DSI 4.40):

<b>CP-0017</b>	
Objetivo	Eliminar correctamente un proyecto
Entrada	Nombre Proyecto = prueba1
Salida	Proyecto eliminado
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “Proyectos” seleccionar la opción “Eliminar Proyecto” 2- Cuando aparezca la pantalla donde se despliegan los proyectos existentes, seleccionar el proyecto: prueba1 3- Presionar el botón “Eliminar”
Prerrequisitos	Ingresar al ingresar al sistema con usuario = 22336911 y clave = 123123

Tabla DSI 4.40: Prueba de eliminación de proyecto

h) Casos de prueba asociados al caso de uso Listar Proyecto (ver tabla DSI 4.41):

<b>CP-0018</b>	
Objetivo	Mostrar todos los proyectos de un usuario
Entrada	Elegir la opción de menú “Listar Proyectos”
Salida	Pantalla donde se listan todos los proyectos pertenecientes al usuario activo
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “Proyectos” seleccionar la opción “Listar Proyectos”
Prerrequisitos	Ingresar al ingresar al sistema con usuario = 22336911 y clave = 123123

Tabla DSI 4.41: Prueba de mostrar todos los proyectos de un usuario

i) Casos de prueba asociados al caso de uso Crear Versión (ver tabla DSI 4.42):

<b>CP-0019</b>	
Objetivo	Crear una nueva versión de un proyecto
Entrada	Nombre Proyecto = prueba1
Salida	Confirmación de la creación de la versión del proyecto elegido
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “Versiones” seleccionar la opción “Crear Versiones de un Proyecto” 2- Cuando aparezca la pantalla donde se despliegan los proyectos existentes, seleccionar el proyecto: prueba1 y presionar el botón “Aceptar” 3- Cuando aparezca la pantalla indicando la creación de la nueva versión y su número y fecha de creación, presionar el botón “Aceptar”

<b>CP-0019</b>	
Prerrequisitos	Ingresar al ingresar al sistema con usuario = 22336911 y clave = 123456

Tabla DSI 4.42: Prueba de crear una nueva versión de un proyecto

j) Casos de prueba asociados al caso de uso Abrir Versión (ver tabla DSI 4.43):

<b>CP-0020</b>	
Objetivo	Abrir una versión de un proyecto
Entrada	Nombre Proyecto = prueba1 Versión = 1
Salida	Se activa la versión de proyecto elegida
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “Versiones” seleccionar la opción “Abrir Versiones de un Proyecto” 2- Cuando aparezca la pantalla donde se despliegan las versiones de los proyectos existentes, seleccionar la versión: 1, el proyecto: prueba1 3- Presionar el botón “Abrir”
Prerrequisitos	Ingresar al ingresar al sistema con usuario = 22336911 y clave = 123456

Tabla DSI 4.43: Prueba de abrir una versión de un proyecto

k) Casos de prueba asociados al caso de uso Listar Versión (ver tabla DSI 4.44):

<b>CP-0021</b>	
Objetivo	Mostrar todas las versiones de proyectos de un usuario
Entrada	Elegir la opción de menú “Listar Versiones de Proyectos”
Salida	Pantalla donde se listan todos las versiones de

<b>CP-0021</b>	
	los proyectos pertenecientes al usuario activo
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “Versiones” seleccionar la opción “Listar Versiones de Proyectos”
Prerrequisitos	Ingresar al ingresar al sistema con usuario = 22336911 y clave = 123123

Tabla DSI 4.44: Prueba de mostrar todas las versiones de proyectos de un usuario

l) Casos de prueba asociados al caso de uso Establecer Parámetros de SOM (ver tablas DSI 4.45 y 4.46):

<b>CP-0022</b>	
Objetivo	Ingresar parámetros permitidos para el algoritmo SOM
Entrada	Elegir la opción de menú “Parámetros de SOM” CantidadCiclosEntrenamiento=3 CantidadObservaciones=50 CantidadVariable=5 CantidadNeuronasMapa=3
Salida	Parámetros para el algoritmo SOM cargados
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “DataMining” seleccionar la opción “Parámetros de SOM” 2- Cuando aparezca la pantalla de ingreso de valores de parámetros de SOM cargar: CantidadCiclosEntrenamiento=3 CantidadObservaciones=50 CantidadVariable=5 CantidadNeuronasMapa=3 3- Presionar el botón “Aceptar”

<b>CP-0022</b>	
Prerrequisitos	Ingresar al ingresar al sistema con usuario = 22336911 y clave = 123123

Tabla DSI 4.45: Prueba de carga de parámetros de SOM

<b>CP-0023</b>	
Objetivo	Ingresar parámetros no permitidos para el algoritmo SOM
Entrada	Elegir la opción de menú “Parámetros de SOM” CantidadCiclosEntrenamiento=3 CantidadObservaciones=0 CantidadVariable=5 CantidadNeuronasMapa=3
Salida	Error en valores de parámetros para el algoritmo SOM
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “DataMining” seleccionar la opción “Parámetros de SOM” 2- Cuando aparezca la pantalla de ingreso de valores de parámetros de SOM cargar: CantidadCiclosEntrenamiento=3 CantidadObservaciones=0 CantidadVariable=5 CantidadNeuronasMapa=3 3- Presionar el botón “Aceptar”
Prerrequisitos	Ingresar al ingresar al sistema con usuario = 22336911 y clave = 123123

Tabla DSI 4.46: Prueba de carga de parámetros de SOM

m) Casos de prueba asociados al caso de uso Establecer Parámetros de C5.0 (ver tablas DSI 4.47 y 4.48):

<b>CP-0024</b>	
Objetivo	Ingresar parámetros permitidos para el algoritmo C5.0
Entrada	Elegir la opción de menú “Parámetros de C5.0” LimpiezaAtributos = No Reglas = No OrdUtilidad = No OrdUtilidadBandas = 0 Boost = No BoostEj = 0 SubcjoValores = No UsarEj = No UsarEjPorc = 0 LockEj = No ValidacionCruzada = No ValidacionCruzadaHojas = 0 IgnorarArchivoCostos = No FuzzyThresholds = No PodaGlobal = Si PodaCF = 25 MinimoCasos = 2
Salida	Parámetros para el algoritmo C5.0 cargados
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “DataMining” seleccionar la opción “Parámetros de C5.0” 2- Cuando aparezca la pantalla de ingreso de valores de parámetros de C5.0 cargar: LimpiezaAtributos = No Reglas = No OrdUtilidad = No OrdUtilidadBandas = 0 Boost = No

<b>CP-0024</b>	
	BoostEj = 0 SubcjoValores = No UsarEj = No UsarEjPorc = 0 LockEj = No ValidacionCruzada = No ValidacionCruzadaHojas = 0 IgnorarArchivoCostos = No FuzzyThresholds = No PodaGlobal = Si PodaCF = 25 MinimoCasos = 2 3- Presionar el botón “Aceptar”
Prerrequisitos	Ingresar al ingresar al sistema con usuario = 22336911 y clave = 123123

Tabla DSI 4.47: Prueba de carga de parámetros de C5.0

<b>CP-0025</b>	
Objetivo	Ingresar parámetros no permitidos para el algoritmo C5.0
Entrada	Elegir la opción de menú “Parámetros de C5.0” LimpiezaAtributos = No Reglas = No OrdUtilidad = No OrdUtilidadBandas = 0 Boost = Si BoostEj = 0 SubcjoValores = No UsarEj = No UsarEjPorc = 0 LockEj = No ValidacionCruzada = No ValidacionCruzadaHojas = 0 IgnorarArchivoCostos = No FuzzyThresholds = No

<b>CP-0025</b>	
	PodaGlobal = Si PodaCF = 25 MinimoCasos = 0
Salida	Error en valores de parámetros para el algoritmo C5.0
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “DataMining” seleccionar la opción “Parámetros de C5.0” 2- Cuando aparezca la pantalla de ingreso de valores de parámetros de C5.0 cargar: LimpiezaAtributos = No Reglas = No OrdUtilidad = No OrdUtilidadBandas = 0 Boost = Si BoostEj = 0 SubcjoValores = No UsarEj = No UsarEjPorc = 0 LockEj = No ValidacionCruzada = No ValidacionCruzadaHojas = 0 IgnorarArchivoCostos = No FuzzyThresholds = No PodaGlobal = Si PodaCF = 25 MinimoCasos = 0 3- Presionar el botón “Aceptar”
Prerrequisitos	Ingresar al ingresar al sistema con usuario = 22336911 y clave = 123123

Tabla DSI 4.48: Prueba de carga de parámetros de C5.0

n) Casos de prueba asociados al caso de uso Ejecutar Proceso de Explotación de Datos (ver tablas DSI 4.49 y 4.52):

<b>CP-0026</b>	
Objetivo	Intentar ejecutar los dos algoritmos en una versión de un proyecto
Entrada	Marcar las casillas correspondientes a SOM y a C5.0
Salida	Resultado de ejecutar los algoritmos SOM y C5.0 sobre un conjunto de datos.
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú "DataMining" seleccionar la opción "Ejecutar DataMining" 2- Cuando aparezca la pantalla donde se despliegan los algoritmos disponibles, marcar SOM y C5.0. 3- Presionar el botón "Ejecutar"
Prerrequisitos	Ingresar al ingresar al sistema con usuario = 22336911 y clave = 123456

Tabla DSI 4.49: Prueba de ejecutar los dos algoritmos en una versión de un proyecto

<b>CP-0027</b>	
Objetivo	Intentar ejecutar el algoritmo SOM en una versión de un proyecto
Entrada	Marcar la casilla correspondientes a SOM
Salida	Resultado de ejecutar los algoritmos SOM sobre un conjunto de datos.
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú "DataMining" seleccionar la opción "Ejecutar DataMining" 2- Cuando aparezca la pantalla donde se

<b>CP-0027</b>	
	despliegan los algoritmos disponibles, marcar SOM. 3- Presionar el botón “Ejecutar”
Prerrequisitos	Ingresar al ingresar al sistema con usuario = 22336911 y clave = 123456

Tabla DSI 4.50: Prueba de ejecutar el algoritmo SOM en una versión de un proyecto

<b>CP-0028</b>	
Objetivo	Intentar ejecutar el algoritmo C5.0 en una versión de un proyecto
Entrada	Marcar las casillas correspondientes a C5.0
Salida	Resultado de ejecutar los algoritmos C5.0 sobre un conjunto de datos.
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “DataMining” seleccionar la opción “Ejecutar DataMining” 2- Cuando aparezca la pantalla donde se despliegan los algoritmos disponibles, marcar C5.0. 3- Presionar el botón “Ejecutar”
Prerrequisitos	Ingresar al ingresar al sistema con usuario = 22336911 y clave = 123456

Tabla DSI 4.51: Prueba de ejecutar el algoritmo C5.0 en una versión de un proyecto

<b>CP-0029</b>	
Objetivo	Intentar ejecutar sin haber seleccionado un algoritmo
Entrada	Desmarcar las casillas correspondientes a SOM y a C5.0
Salida	Mensaje que no se selecciono algoritmo de DataMining.

<b>CP-0029</b>	
Condiciones	No se permite el acceso a la base de datos de otros programas mientras se realizan las pruebas
Procedimiento	1- Ingresar desde menú principal al menú “DataMining” seleccionar la opción “Ejecutar DataMining” 2- Cuando aparezca la pantalla donde se despliegan los algoritmos disponibles, desmarcar SOM y C5.0. 3- Presionar el botón “Ejecutar”
Prerrequisitos	Ingresar al ingresar al sistema con usuario = 22336911 y clave = 123456

Tabla DSI 4.52: Prueba de ejecutar sin haber seleccionado un algoritmo

#### **4.10. Aprobación del Diseño del Sistema de Información**

##### **4.10.1. Presentación y Aprobación del Diseño del Sistema de Información**

En esta tarea se realiza la presentación del diseño del sistema de información ante la directora de Tesis para la aprobación final del mismo.

En una reunión mantenida entre el tesista y la Directora del proyecto se dio por aprobada la fase de Diseño del Sistema de Información.

## **CAPÍTULO 5 – CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN**

En este proceso se genera el código de los componentes del Sistema de Información, se desarrollan todos los procedimientos de operación y seguridad y se elaboran todos los manuales de usuario final y de explotación con el objetivo de asegurar el correcto funcionamiento del Sistema para su posterior implantación.

Para conseguir dicho objetivo, en este proceso se realizan las pruebas unitarias, las pruebas de integración de los subsistemas y componentes y las pruebas del sistema, de acuerdo al plan de pruebas establecido.

Asimismo, se define la formación de usuario final y, si procede, se construyen los procedimientos de migración y carga inicial de datos.

Al ser MÉTRICA Versión 3 una metodología que cubre tanto desarrollos estructurados como orientados a objetos, las actividades de ambas aproximaciones están integradas en una estructura común.

La fase Especificaciones de Construcción del Sistema de Información, obtenida en la actividad de Generación de Especificaciones de Construcción, es la base para la construcción del sistema de información. En dicho producto se recoge la información relativa al entorno de construcción del sistema de información, la especificación detallada de los componentes y la descripción de la estructura física de datos, tanto bases de datos como sistemas de ficheros.

En la actividad Preparación del Entorno de Generación y Construcción, se asegura la disponibilidad de la infraestructura necesaria para la generación del código de los componentes y procedimientos del sistema de información. Una vez configurado el entorno de construcción, se realiza la codificación y las pruebas de los distintos componentes que conforman el sistema de información, en las actividades:

- Generación del Código de los Componentes y Procedimientos, que se hace según las especificaciones de construcción del sistema de información, y conforme al plan de integración del sistema de información

- Ejecución de las Pruebas Unitarias, dónde se llevan a cabo las verificaciones definidas en el plan de pruebas para cada uno de los componentes
- Ejecución de las Pruebas de Integración, que incluye la ejecución de las verificaciones asociadas a los subsistemas y componentes, a partir de los componentes verificados individualmente, y la evaluación de los resultados.

Una vez construido el sistema de información y realizadas las verificaciones correspondientes, se lleva a cabo la integración final del sistema de información en la actividad Ejecución de las Pruebas del Sistema, comprobando tanto las interfaces entre subsistemas y sistemas externos como los requisitos, de acuerdo a las verificaciones establecidas en el plan de pruebas para el nivel de pruebas del sistema.

Si se ha establecido la necesidad de realizar una migración de datos, la construcción y pruebas de los componentes y procedimientos relativos a dicha migración y a la carga inicial de datos se realiza en la actividad Construcción de los Componentes y Procedimientos de Migración y Carga Inicial de Datos.

### **5.1. Preparación del Entorno de Generación y Construcción**

El objetivo de esta actividad es asegurar la disponibilidad de todos los medios y facilidades para que se pueda llevar a cabo la construcción del sistema de información. Entre estos medios, cabe destacar la preparación de los puestos de trabajo, equipos físicos y lógicos, gestores de bases de datos, bibliotecas de programas, herramientas de generación de código, bases de datos o ficheros de prueba, entre otros.

Las características del entorno de construcción y sus requisitos de operación y seguridad, así como las especificaciones de construcción de la estructura física de datos, se establecen en la actividad Generación de Especificaciones de Construcción, y constituyen el punto de partida para la realización de esta actividad.

### 5.1.1. Implantación de la Base de Datos Física o Ficheros

En esta tarea se debe:

- Crear los elementos del sistema gestor de base de datos o sistema de ficheros
- Reservar el espacio de almacenamiento, definiendo, entre otros, los dispositivos físicos a emplear, tamaño de los bloques, tipo de registro físico, zona de desbordamiento, opciones de almacenamiento de datos, etc.
- Inicializar la base de datos o ficheros, cargando los datos considerados necesarios en el espacio de almacenamiento previamente definido.

#### 5.1.1.1. Creación de la Base de Datos

Para el presente proyecto, esta tarea consiste en instalar la base de datos Microsoft SQL Server, crear la base de datos “dm”, dentro de la cual se crearán las tablas del sistema.

Para la instalación de la base de datos se seguirán las indicaciones del fabricante del producto.

Una vez instalada la base de datos se procederá a crear la base de datos dm, para lo cual se deben seguir los pasos que se indican en la tabla CSI 5.1:

Paso	Tarea
1	Ejecutar la interfaz de comandos: acceso directo a “Analizador de consultas”
2	Crear la base de datos: create database dm
3	Para el uso de la base, a partir de ahora habrá que usar la siguiente instrucción: use dm

Tabla CSI 5.1: Pasos a seguir para la creación de la Base de Datos

Con la base de datos creada y en ejecución, se deberá crear las tablas del sistema, lo cual se hará a través de instrucciones SQL en base a los datos definidos en la

fase de diseño del sistema.

### **5.1.2. Preparación del Entorno de Construcción**

En esta tarea se prepara el entorno en el que se construirán los componentes del sistema de información, contemplando aspectos tales como:

- Bibliotecas o librerías a utilizar
- Herramientas: generadores de código, editores, compiladores, verificadores sintácticos, montadores de enlace
- Puestos de trabajo
- Implementación de los procedimientos de operación y seguridad propios del entorno de construcción, de acuerdo a los requisitos de seguridad y operación establecidos en la tarea Especificación del Entorno de Construcción.

#### **5.1.2.1. Generación del Entorno de trabajo**

Para el presente proyecto, se deberán instalar el producto Visual FoxPro 9.0 en su versión típica y el controlador ODBC de acceso a la base de datos Microsoft SQL Server. Para esta tarea se recomienda consultar el manual de instalación de los mencionados productos.

## **5.2. Generación del Código de los Componentes y Procedimientos**

El objetivo de esta actividad es la codificación de los componentes del sistema de información, a partir de las especificaciones de construcción obtenidas en el proceso Diseño del Sistema de Información, así como la construcción de los procedimientos de operación y seguridad establecidos para el mismo.

En paralelo a esta actividad, se desarrollan las actividades relacionadas con las pruebas unitarias y de integración del sistema de información. Esto permite una

construcción incremental, en el caso de que así se haya especificado en el plan de pruebas y en el plan de integración del sistema de información.

### **5.2.1. Generación del Código de Componentes**

En esta tarea se genera el código correspondiente a cada uno de los componentes del sistema de información, identificados en la tarea Definición de Componentes y Subsistemas de Construcción.

Para generar el código fuente se tienen en cuenta los estándares de nomenclatura, codificación y calidad utilizados por la organización y recogidos en el catálogo de normas. Con el fin de verificar que el código fuente especifica de forma correcta el componente, se realiza su ensamblaje o compilación, verificando y corrigiendo los errores sintácticos, y el enlace del código objeto obtenido con las correspondientes bibliotecas.

### **5.2.2. Generación del Código de los Procedimientos de Operación y Seguridad**

El objetivo de esta tarea es generar los procedimientos de operación y administración del sistema de información, así como los procedimientos de seguridad y control de acceso, necesarios para ejecutar el sistema una vez que se haya implantado y esté en producción.

#### **5.2.2.1. Generación de Procedimientos de Operación y Seguridad**

Como ya se ha indicado antes, el sistema posee dos componentes físicos: cliente y Servidor. En el presente apartado, se van a especificar aspectos técnicos de seguridad del componente servidor, ya que dentro del mismo es donde se encuentran las transacciones que reflejan las reglas de negocio y la base de datos del sistema.

Para la ejecución de los servicios desde el servidor, se deberá crear un grupo de trabajo, dentro de los grupos de usuarios del sistema operativo, llamado dm. Dentro de este grupo se deberá incluir a todos aquellos usuarios del sistema que deban operar con el presente gestor de proyectos, dicho grupo tendrá permisos de acceso a las carpetas donde se crearán los directorios de trabajo que contengan los documentos del proyecto.

La aplicación necesitará una cuenta para conectarse a la base de datos. Para ello se recomienda la creación de un usuario “dm” dentro del servidor de Base de Datos. Esta cuenta debe tener permisos de “db\_owner” sobre la base de datos donde residen las tablas del sistema.

Para la ejecución del sistema, se recomienda la creación de un usuario adicional que solamente tenga permisos de lectura sobre las tablas del sistema.

### **5.3. Ejecución de las Pruebas Unitarias**

En esta actividad se realizan las pruebas unitarias de cada uno de los componentes del sistema de información, una vez codificados, con el objeto de comprobar que su estructura es correcta y que se ajustan a la funcionalidad establecida.

En el plan de pruebas se ha definido el entorno necesario para la realización de cada nivel de prueba, así como las verificaciones asociadas a las pruebas unitarias, la coordinación y secuencia a seguir en la ejecución de las mismas y los criterios de registro y aceptación de los resultados (Ver punto DSI 4.9).

#### **5.3.1. Realización y Evaluación de las Pruebas Unitarias**

El objetivo de esta tarea es comprobar el correcto funcionamiento de los componentes del sistema de información, codificados en la actividad Generación del Código de los Componentes y Procedimientos, conforme a las verificaciones establecidas en el plan de pruebas para el nivel de pruebas unitarias, en la actividad Especificación Técnica del Plan de Pruebas.

Para cada verificación establecida, se realizan las pruebas con los casos de pruebas asociados, efectuando el correspondiente análisis y evaluación de los resultados, y generando un registro conforme a los criterios establecidos en el plan de pruebas.

Seguidamente, se analizan los resultados de las pruebas unitarias, evaluándose las mismas para comprobar que los resultados son los esperados. Si los resultados no son los esperados hay que proceder a realizar las correcciones pertinentes.

#### **5.3.1.1. Resultado de la Realización de las Pruebas Unitarias**

A continuación, en la tabla CSI 5.2, se detalla el resultado de las pruebas unitarias luego de realizar dos iteraciones de modificación sobre el desarrollo de los componentes de comunicación:

<b>Código</b>	<b>Componente</b>	<b>Resultado</b>
CP-0001	Comunicación	Correcto

Tabla CSI 5.2: Resultado de la ejecución de los casos de prueba unitarios

### **5.4. Ejecución de las Pruebas del Sistema**

El objetivo de las pruebas del sistema es comprobar la integración del sistema de información globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de sistemas de información con los que se comunica.

En la realización de estas pruebas es importante comprobar la cobertura de los requisitos, dado que su incumplimiento puede comprometer la aceptación del sistema por el equipo de operación responsable de realizar las pruebas de implantación del sistema, que se llevarán a cabo en el proceso Implantación y Aceptación del Sistema.

#### **5.4.1. Realización de las Pruebas del Sistema**

El objetivo de esta tarea es comprobar la integración de todos los subsistemas y componentes del sistema de información, así como la interacción del mismo con otros sistemas de información con los que se relaciona, de acuerdo a las verificaciones establecidas para el nivel de pruebas del sistema.

Para cada verificación establecida, se realizan las pruebas con los casos de pruebas asociados, efectuando el correspondiente análisis e informe de los resultados y generando un registro conforme a los criterios establecidos en el plan de pruebas.

#### 5.4.1.1. Resultado de la Realización de las Pruebas de Sistema

A continuación, en la tabla CSI 5.3, se detalla el resultado de las pruebas a nivel de sistemas luego de realizar tres iteraciones de modificación sobre el desarrollo del sistema:

<b>Código</b>	<b>Caso de Uso</b>	<b>Objetivo</b>
CP-0002	Validar Usuario	Correcto
CP-0003	Validar Usuario	Correcto
CP-0004	Validar Usuario	Correcto
CP-0005	Cambiar Clave	Correcto
CP-0006	Cambiar Clave	Correcto
CP-0007	Cambiar Clave	Correcto
CP-0008	Incorporar Usuario	Correcto
CP-0009	Incorporar Usuario	Correcto
CP-0010	Modificar Usuario	Correcto
CP-0011	Crear Proyecto	Correcto
CP-0012	Crear Proyecto	Correcto
CP-0013	Crear Proyecto	Correcto
CP-0014	Crear Proyecto	Correcto
CP-0015	Crear Proyecto	Correcto
CP-0016	Crear Proyecto	Correcto
CP-0017	Eliminar Proyecto	Correcto
CP-0018	Listar Proyecto	Correcto
CP-0019	Crear Versión	Correcto
CP-0020	Abrir Versión	Correcto
CP-0021	Listar Versión	Correcto
CP-0022	Establecer Parámetros de SOM	Correcto

<b>Código</b>	<b>Caso de Uso</b>	<b>Objetivo</b>
CP-0023	Establecer Parámetros de SOM	Correcto
CP-0024	Establecer Parámetros de C5.0	Correcto
CP-0025	Establecer Parámetros de C5.0	Correcto
CP-0026	Ejecutar Proceso de Explotación de Datos	Correcto
CP-0027	Ejecutar Proceso de Explotación de Datos	Correcto
CP-0028	Ejecutar Proceso de Explotación de Datos	Correcto
CP-0029	Ejecutar Proceso de Explotación de Datos	Correcto

Tabla CSI 5.3: Resultado de la ejecución de los casos de prueba unitarios

#### **5.4.2. Evaluación del Resultado de las Pruebas del Sistema**

El objetivo de esta actividad es analizar los resultados de las pruebas del sistema de información y efectuar su evaluación. Dicha evaluación recoge el grado de cumplimiento de las mismas.

##### **5.4.2.1. Resultado de la Evaluación de los Resultados de las Pruebas de Sistema**

En función del análisis de los resultado de los casos de prueba indicado en la tabla CSI 5.4 podemos decir que el sistema a alcanzado los niveles de calidad deseados, dado que todas las salidas se encuentran dentro de los parámetros de valores esperado.

#### **5.5. Aprobación del Sistema de Información**

En esta tarea se recopilan los productos del sistema de información y se presentan al Comité de Seguimiento para su aprobación.

En una reunión mantenida entre el tesista y la Directora del proyecto se dio por aprobada la fase de Construcción del Sistema de Información.

## CAPÍTULO 6 – IMPLANTACIÓN Y ACEPTACIÓN DEL SISTEMA

Este proceso tiene como objetivo principal la entrega y aceptación del sistema en su totalidad, y la realización de todas las actividades necesarias para el paso a producción del mismo.

En primer lugar, se revisa la estrategia de implantación que ya se determinó en el proceso Estudio de Viabilidad del Sistema (EVS), presentado en el capítulo 1. Se estudia su alcance y, en función de sus características, se define un plan de implantación y se especifica el equipo que lo va a llevar a cabo. Conviene señalar la participación del usuario de operación en las pruebas de implantación, del usuario final en las pruebas de aceptación, y del responsable de mantenimiento.

Las actividades previas al inicio de la producción incluyen la preparación de la infraestructura necesaria para configurar el entorno, la instalación de los componentes, la activación de los procedimientos manuales y automáticos asociados y, cuando proceda, la migración o carga inicial de datos. Para ello se toman como punto de partida los productos software probados, obtenidos en el proceso Construcción del Sistema de Información (CSI) (ver capítulo 4) y su documentación asociada.

Se realizan las pruebas de implantación y de aceptación del sistema en su totalidad, las mismas responden a los siguientes propósitos:

- *Las pruebas de implantación* cubren un rango muy amplio, que va desde la comprobación de cualquier detalle de diseño interno hasta aspectos tales como las comunicaciones. Se debe comprobar que el sistema puede gestionar los volúmenes de información requeridos, se ajusta a los tiempos de respuesta deseados y que los procedimientos de respaldo, seguridad e interfaces con otros sistemas funcionan correctamente. Se debe verificar también el comportamiento del sistema bajo las condiciones más extremas.
- *Las pruebas de aceptación* se realizan por y para los usuarios. Tienen como objetivo validar formalmente que el sistema se ajusta a sus necesidades.

Asimismo, se llevan a cabo las tareas necesarias para la preparación del mantenimiento, siempre y cuando se haya decidido que éste va a efectuarse. En cualquier caso, es necesario que la persona que vaya a asumir el mantenimiento conozca el sistema, antes de su incorporación al entorno de producción.

Además hay que determinar los servicios (y niveles para cada uno) que requiere el sistema que se va a implantar, y el acuerdo que se adquiere una vez que se inicie la producción. Hay que distinguir entre servicios de gestión de operaciones (servicios por lotes, seguridad, comunicaciones, etc.) y servicios al cliente (servicio de atención a usuario, mantenimiento, etc.) que se deben negociar en cuanto a recursos, horarios, coste, etc. Se fija el nivel con el que se prestará el servicio como indicador de la calidad del mismo.

Conviene señalar que la implantación puede ser un proceso iterativo que se realiza de acuerdo al plan establecido para el comienzo de la producción del sistema en su entorno de operación. Para establecer este plan se tiene en cuenta:

- El cumplimiento de los requisitos de implantación definidos en la actividad Establecimiento de Requisitos y especificados en la actividad Establecimiento de Requisitos de Implantación.
- La estrategia de transición del sistema antiguo al nuevo.

Finalmente, se realizan las acciones necesarias para el inicio de la producción.

### **6.1. Establecimiento del Plan de Implantación**

En esta actividad se revisa la estrategia de implantación para el sistema, establecida inicialmente en el proceso Estudio de Viabilidad del Sistema (EVS), presentado en el capítulo 1. Se identifican los distintos sistemas de información que forman parte del sistema objeto de la implantación. Para cada sistema se analizan las posibles dependencias con otros proyectos, que puedan condicionar el plan de implantación.

Una vez estudiado el alcance y los condicionantes de la implantación, se decide

si ésta se puede llevar a cabo. Será preciso establecer, en su caso, la estrategia que se concretará de forma definitiva en el plan de implantación.

Se constituye el equipo de implantación, determinando los recursos humanos necesarios para la propia instalación del sistema, para las pruebas de implantación y aceptación, y para la preparación del mantenimiento. Se identifican, para cada uno de ellos, sus perfiles y niveles de responsabilidad.

### **6.1.1. Definición del Plan de Implantación**

La estrategia de implantación del sistema se habrá determinado en la tarea Evaluación de las Alternativas y Selección del proceso Estudio de Viabilidad del Sistema, presentada en el capítulo 1, en función de la envergadura del sistema, es decir el número de sistemas de información implicados en la implantación y la cobertura geográfica, cuyo alcance depende de las características y complejidad de los sistemas de información que conforman el sistema objeto de la implantación.

Se revisan los requisitos de implantación (instalación, infraestructura, formación) establecidos en la tarea Especificación de Requisitos de Implantación y los procedimientos implicados en la implantación, establecidos para cada uno de los sistemas de información en la tarea Especificación de Requisitos de Operación y Seguridad, con el fin de asegurar su adecuación a la estrategia global de implantación.

Una vez analizada la información anterior, se define un plan de implantación que permita calcular adecuadamente el esfuerzo y los recursos necesarios para llevar a cabo con éxito la implantación. Dicho plan debe contemplar todas las tareas relacionadas con:

- La formación necesaria para la implantación, tanto a usuarios finales como al equipo que se encarga de realizar las pruebas de implantación y aceptación del sistema.
- La preparación de la infraestructura necesaria para la incorporación del sistema al entorno de operación.

- La instalación de todos los componentes y procedimientos manuales y automáticos asociados a cada sistema de información implicado en la implantación.
- La ejecución de los procedimientos de carga inicial y migración de datos, si se determinó su necesidad.
- La realización de las pruebas de implantación y aceptación del sistema.
- La formalización del plan de mantenimiento.

#### **6.1.1.1. Formación de usuarios finales y equipo de pruebas**

Se prevé capacitar a un usuario en el uso del sistema de información de forma tal que pueda utilizarlo para verificar que el mismo cumple con sus requisitos para posteriormente aceptar el sistema. Dicho usuario debe ser una persona con experiencia en el negocio y en el desarrollo de proceso de Explotación de Datos.

#### **6.1.1.2. Preparación de la infraestructura necesaria para la incorporación del sistema al entorno de operación**

Como ya se ha dicho en la fase de Diseño y Construcción del Sistema, será necesario instalar los componentes de cliente y servidor de la aplicación. Para la correcta implementación del sistema, se deben contemplar varios roles que para el presente proyecto serán llevados a delante por el Tesis. Entre las tareas a desarrollar se encuadran las actividades descriptas en la tabla IAS 6.1:

<b>Tarea</b>	<b>Rol</b>
Implementación de la base de datos que necesita la aplicación	Administrador de Base de Datos
Instalación de los clientes de aplicación	Administrador de aplicación e infraestructura
Instalación y verificación del correcto funcionamiento de las cuestiones de seguridad y comunicación	Administrador de seguridad y comunicación

Tabla IAS 6.1: Tareas a desarrollar para la implementación del sistema

### 6.1.1.3. Ejecución de carga inicial

Para que el usuario pueda probar el sistema de información, será necesario hacer una carga inicial de datos, para ello se ejecutará los scripts SQL necesario para la inserción de los datos del sistema, esta tarea se describe en la tabla IAS 6.2:

Tarea	Rol
Ejecución de los procedimientos de carga inicial	Administrador de Base de Datos

Tabla IAS 6.2: Tareas a desarrollar para la carga inicial de datos

### 6.1.1.4. Realización de las pruebas de implementación y aceptación del sistema

Se necesitará generar un perfil de usuario con permisos para acceder a las carpetas y base de datos del sistema. Esta tarea se describe en la tabla IAS 6.3:

Tarea	Rol
Creación de las cuentas del usuario	Administrador de aplicación e infraestructura

Tabla IAS 6.3: Tareas a desarrollar para la operación del sistema por parte del usuario

### 6.1.1.5. Formalización del plan de mantenimiento

La etapa de mantenimiento del sistema de información excede los límites del proyecto de tesis. De todas formas, en la tabla IAS 6.4, se indican como se resolverán las cuestiones de mantenimiento que con mayor frecuencia aparecen en la etapa de mantenimiento del sistema cuando estos se encuentran en producción:

Concepto	Explicación
Espacio de almacenamiento	No se prevé que las bases de datos del sistema crezcan de forma desmesurada, por tal motivo no se considera necesario plantar esquemas de mantenimiento específicos para este concepto.

<b>Concepto</b>	<b>Explicación</b>
Performance del motor de Base de Datos	No se prevé que las bases de datos del sistema crezcan de forma desmesurada, ni tampoco que la cantidad de usuarios conectados al sistema pueda afectar los tiempos de respuesta de la misma.
Accesos indebidos al sistema	Para controlar este factor y poder proveer algún ajuste al sistema, se ha previsto la generación de un archivo de log que contendrá el detalle de todas las operaciones realizadas dentro del mismo. Por tal motivo, será conveniente revisar este archivo periódicamente
Control de las copias de seguridad	Las copia de seguridad son el único recurso para reestablecer el sistema en caso de una caída del mismo. Es por ello importante verificar que las mismas se realicen correctamente para lo cual se aconseja recuperarlas periódicamente en otro equipo y probar que el sistema se encuentre disponible como así también la consistencia de los datos

Tabla IAS 6.4: Aspectos a tener en cuenta para el mantenimiento del sistema

### **6.1.2. Especificación del Equipo de Implantación**

Se constituye el equipo de trabajo necesario para llevar a cabo la implantación y aceptación del sistema, según el plan de implantación establecido en la tarea anterior. Para ello se identifican, en función del nivel de esfuerzo requerido, los distintos participantes implicados en la implantación del sistema (usuarios, equipo técnico y responsable de mantenimiento), determinando previamente sus perfiles, responsabilidades, nivel de implicación y fechas previstas de participación a lo largo de toda la implantación.

#### **6.1.2.1. Equipo de Implantación**

Si bien el presente trabajo ha sido desarrollado por el tesista con la colaboración de la Directora del proyecto, se prevé para esta instancia la incorporación de uno de los alumnos de la Carrera Especialidad en Procesos de Explotación de Datos que hará las veces de usuario final y colaborará en la realización de la prueba de aceptación del sistema. A continuación, en la tabla IAS 6.5, se describen los involucrados en el desarrollo de esta fase y su función:

<b>Rol</b>	<b>Perfil</b>
Usuario Final	Es el especialista en Explotación de Datos que será el encargado de ejecutar el sistema
Administrador de Base de Datos	Esta función está a cargo del tesista y consiste en instalar, crear y administrar los recursos de la base de datos
Administrador de Seguridad y comunicaciones	Esta función está a cargo del tesista y consiste en instala y verificar el correcto funcionamiento del componente de comunicación del sistema.
Administrador de Aplicaciones e Infraestructura	Esta función está a cargo del tesista y consiste en instalar el aplicativo y asignar los permisos, desde el sistema operativo, para que los distintos usuarios puedan utilizar el sistema.

Tabla IAS 6.5: Descripción del equipo de implementación

## 6.2. Incorporación del Sistema al Entorno de Operación

En esta actividad se realizan todas las tareas necesarias para la incorporación del sistema al entorno de operación en el que se van a llevar a cabo las pruebas de implantación y aceptación del sistema.

Mientras que las pruebas unitarias, de integración y del sistema se pueden ejecutar en un entorno distinto de aquél en el que finalmente se implantará, las pruebas de implantación y aceptación del sistema deben ejecutarse en el entorno real de operación. El propósito es comprobar que el sistema satisface todos los requisitos especificados por el usuario en las mismas condiciones que cuando se inicie la producción.

Por tanto, como paso previo a la realización de dichas pruebas y de acuerdo al plan de implantación establecido, se verifica que los recursos necesarios están disponibles para que se pueda realizar, adecuadamente, la instalación de todos los componentes que integran el sistema, así como la creación y puesta a punto de las bases de datos en el entorno de operación. Asimismo, se establecen los procedimientos de explotación y uso de las bases de datos de acuerdo a la normativa existente en dicho

entorno.

### **6.2.1. Preparación de la Instalación**

En esta tarea se verifica que está disponible la infraestructura necesaria para configurar el entorno. Dicha infraestructura debe cumplir los requisitos de implantación (instalación e infraestructura) y tener en cuenta los procedimientos de seguridad y control de acceso (mantenimiento de la integridad y confidencialidad de los datos, control de accesos al sistema, copias de seguridad y recuperación de datos, etc.), y operación y administración del sistema (estándares, recuperación y reanudación de trabajos, planificación de trabajos, etc.).

Además, si alguno de los sistemas de información implicados en la implantación lleva implícita una migración de datos habrá que tener en cuenta, también, las características del entorno y los procedimientos propios de la migración establecidos en el plan de migración y carga inicial de datos, obtenido en la actividad Diseño de la Migración y Carga Inicial de Datos (DSI 9).

Una vez comprobada la idoneidad de los distintos elementos relacionados con la infraestructura, se realiza la instalación del software de base necesario para la incorporación posterior de los componentes asociados a los sistemas de información implicados en la implantación.

#### **6.2.1.1. Descripción de la Instalación**

Dentro del contexto de desarrollo del presente trabajo, se deberá proceder a la instalación del siguiente software:

- Sistema Operativo Microsoft Windows 2000 o XP Server, en el equipo Servidor
- Sistema Operativo Microsoft Windows 2000 o XP, en el equipo cliente
- Microsoft SQLServer 2005 en el equipo servidor

### **6.2.2. Realización de la Instalación**

Se realiza la instalación de todos los componentes del nuevo sistema, incluidos los procedimientos manuales y automáticos, de acuerdo al plan de implantación y a su ubicación física, establecida en el proceso Diseño del Sistema de Información. Se deben tener en cuenta los estándares y normativas por los que se rige la organización en los entornos de operación.

Asimismo, se prepara el entorno de datos identificando los sistemas de información que forman parte del sistema objeto de la implantación. Para cada uno de ellos:

- Se crean las bases de datos a partir del esquema físico elaborado en el proceso de construcción.
- Se establecen los procedimientos de explotación y uso de las bases de datos, es decir, la normativa necesaria para la utilización de las bases de datos, actualización, consulta, etc.
- Se revisan los procedimientos necesarios para realizar las copias de seguridad de los datos y de restauración de las copias indicando su frecuencia, así como los procedimientos de consolidación y sincronización de la información, éstos últimos cuando proceda.
- Se preparan las autorizaciones de acceso a los datos para los distintos perfiles de usuarios.

Una vez comprobada la correcta instalación del nuevo sistema, se activan los procedimientos de operación, de administración del sistema, de seguridad y de control de acceso. Incluyen el arranque y cierre del sistema según la frecuencia establecida, la planificación de trabajos, su recuperación y reanudación, las autorizaciones de acceso al sistema según los distintos perfiles de usuario, etc. Asimismo, si es necesaria una migración de datos se activarán también los procedimientos asociados.

### **6.2.2.1. Instalación del sistema**

Se procedió a realizar la instalación del sistema en lo que sería un ambiente de producción típico y la misma resultó completamente exitosa.

## **6.3. Carga de Datos al Entorno de Operación**

Teniendo en cuenta que los sistemas de información que forman parte del sistema a implantar pueden mejorar, ampliar o sustituir a otros ya existentes en la organización, puede ser necesaria una carga inicial y/o una migración de datos cuyo alcance dependerá de las características y cobertura de cada sistema de información implicado. Por tanto, la necesidad de una migración de datos puede venir determinada desde el proceso Estudio de Viabilidad del Sistema (EVS), en la actividad Selección de la Solución. Allí se habrá establecido la estrategia a seguir en la sustitución, evaluando las opciones del enfoque de desarrollo e instalación más apropiados para llevarlo a cabo.

En cualquier caso, en la actividad Diseño de la Migración y Carga Inicial de Datos se habrán definido y planificado los procesos y procedimientos necesarios para llevar a cabo la migración, realizándose su codificación en la actividad Construcción de los Componentes y Procedimientos de Migración y Carga Inicial de Datos.

### **6.3.1. Migración y Carga inicial de Datos**

Se realiza la carga inicial de datos del nuevo sistema, y se comprueba que ha finalizado correctamente.

#### **6.3.1.1. Instalación del sistema**

Se procedió a ejecutar los scripts de carga de datos iniciales y los mismos se ejecutaron con éxito, a partir de este momento el sistema se encuentra correctamente instalado y operable.

## **6.4. Pruebas de Implantación del Sistema**

La finalidad de las pruebas de implantación es doble:

- Comprobar el funcionamiento correcto del mismo en el entorno de operación.
- Permitir que el usuario determine, desde el punto de vista de operación, la aceptación del sistema instalado en su entorno real, según el cumplimiento de los requisitos especificados.

Para ello, el responsable de implantación revisa el plan de pruebas de implantación, descrito en el ítem 4.9.2.1 y los criterios de aceptación del sistema, previamente elaborados. Las pruebas las realizan los técnicos de sistemas y de operación, que forman parte del grupo de usuarios técnicos que ha recibido la formación necesaria para llevarlas a cabo.

Una vez ejecutadas estas pruebas, el equipo de usuarios técnicos informa de las incidencias detectadas al responsable de implantación, el cual analiza la información y toma las medidas correctoras que considere necesarias para que el sistema dé respuesta a las especificaciones previstas, momento en el que el equipo de operación lo da por probado.

### **6.4.1. Preparación de las Pruebas de Implantación**

Se comprueba la disponibilidad de los recursos humanos y técnicos necesarios para realizar las pruebas de implantación. Se revisan las verificaciones establecidas en el plan de pruebas, especificados en el ítem 4.9.2.1. Se preparan las condiciones que permitan simular las situaciones límite previstas para las pruebas, formalmente, se comunica el plan de pruebas de implantación al equipo responsable de llevarlas a cabo.

#### **6.4.1.1. Pruebas de Implantación**

Luego de revisar el esquema de pruebas definido en la fase de diseño y aplicado durante la etapa de construcción vistas en el capítulo 5, se considera que el mismo posee

una adecuada cobertura de la funciones del sistema, y por tal motivo no se considera necesario la generación de un nuevo plan de pruebas.

#### **6.4.2. Realización de las Pruebas de implantación**

Se realizan las pruebas de implantación, de acuerdo a las verificaciones establecidas en el plan de pruebas definido en la actividad Especificación Técnica del Plan de Pruebas (ítem 9, capítulo 4).

##### **6.4.2.1. Prueba de Implantación**

El usuario cargó todos los casos de prueba en el entorno de producción, y la ejecución de los mismos fue exitosa en todos los casos.

#### **6.4.3. Evaluación del Resultado de las Pruebas de Implantación**

Se evalúan los resultados de las pruebas analizando las incidencias recibidas y comprobando que se han llevado a cabo todos los casos de pruebas establecidos en el plan de pruebas.

##### **6.4.3.1. Evaluación de la Prueba de Implantación**

Como el usuario no ha registrado anomalías en la carga de los casos de prueba, se da por aprobada la prueba de implementación del sistema.

### **6.5. Pruebas de Aceptación del Sistema**

Las pruebas de aceptación tienen como fin validar que el sistema cumple los requisitos básicos de funcionamiento esperado y permitir que el usuario determine la aceptación del sistema. Por este motivo, estas pruebas son realizadas por el usuario final que, durante este periodo de tiempo, debe plantear todas las deficiencias o errores que encuentre antes de dar por aprobado el sistema definitivamente.

Los Directores de los Usuarios revisan los criterios de aceptación, especificados previamente en el plan de pruebas del sistema, y dirigen las pruebas de aceptación final

que llevan a cabo los usuarios expertos. A su vez, éstos últimos deben elaborar un informe que los Directores de los Usuarios analizan y evalúan para determinar la aceptación o rechazo del sistema.

### **6.5.1. Realización de las Pruebas de Aceptación**

Se llevan a cabo las pruebas de aceptación final del sistema para asegurar que todos los componentes responden a los criterios de aceptación especificados.

Se registra la realización de las pruebas, incluyendo un informe que recoja la desviación de los requisitos establecidos y los problemas que quedan sin resolver.

#### **6.5.1.1. Pruebas de Aceptación**

Las pruebas de aceptación del sistema se han llevado a cabo entre el usuario experto en procesos de Explotación de Datos juntamente con la prueba de implementación y el resultado de la misma ha sido exitoso.

### **6.6. Presentación y Aprobación del Sistema**

Una vez que se han efectuado las pruebas de implantación y de aceptación, y que se ha fijado el acuerdo de nivel de servicio, el Comité de Dirección debe formalizar la aprobación del sistema. Para esto, se lleva a cabo una presentación general del sistema al Comité de Dirección y se espera la confirmación de su aprobación.

En una reunión mantenida entre el tesista y la Directora del proyecto se dio por aprobada la fase de Implementación del Sistema de Información, no obstante, como el presente trabajo forma parte de la tesis de Maestría, la aprobación final del sistema consistirá en la defensa del mismo ante un tribunal evaluador oportunamente reunido a tal fin.

## **CAPÍTULO 7 – CONCLUSIÓN Y FUTURAS LÍNEAS DE TRABAJO**

### **7.1. Conclusión**

Respecto de las conclusiones obtenidas con el presente trabajo, las dividiría en dos orientaciones, por un lado las relacionadas con la Métrica Versión 3 y por otro las que tienen que ver con la herramienta desarrollada.

En cuanto a la aplicación de la metodología Métrica Versión 3 se destaca por un lado en la construcción en general del sistema y el mantenimiento de un orden en cada fase del proyecto, y por otra parte en el hecho de la familiarización con éste esquema de trabajo, el cual es de gran utilidad en el desarrollo de proyectos de Ingeniería de Software a través de la definición de un marco estratégico para el desarrollo de Sistemas de Información, el establecimiento de una mayor importancia al análisis de requisitos, el incremento de la productividad, permitiendo una mayor capacidad de adaptación a los cambios y teniendo en cuenta la reutilización en la medida de lo posible y la facilitación de la operación, mantenimiento y uso de los productos software obtenidos.

En relación con la herramienta construida los analistas del negocio tienen la posibilidad de realizar una explotación de datos en forma sencilla, teniendo la facilidad del versionado de proyectos, fundamentalmente para que realice una variedad de estudios sobre los datos sin resignar las diferentes alternativas.

La herramienta además brinda al analista, en esta primera versión, dos importantes algoritmos de explotación de datos, SOM y C5.0, facilitando a los usuarios del sistema su aprendizaje a través de valores por defecto para poder correr los mismos en una primera etapa sin tener que analizar todas las alternativas posibles en sus parámetros. Se puso especial énfasis en las facilidades de reportes y salidas para los analistas del negocio.

## 7.2. Futuras Líneas de Trabajo

Como futuras líneas de trabajo se han identificado los siguiente tópicos:

- Incluir mas algoritmos de Explotación de Datos en esta herramienta.
- Mejorar los gráficos de salida de los algoritmos implementados.
- Internacionalización del sistema, incluyendo diferentes idiomas para su interacción.
- Modificación del sistema para que pueda ser multiplataforma.

---

## CAPÍTULO 8 – BIBLIOGRAFÍA Y GLOSARIO DE TÉRMINOS

### 8.1. Bibliografía

- [Cabena, P. et al; 1998]      Discovering Data Mining from Concept to Implementation. Prentice Hall.
- [García Martínez et al; 2003]      García Martínez, R.; Servente, M.; Pasquín, D.; 2003. *Sistemas Inteligentes*, Capítulo 1: “Aprendizaje Automático”, Capítulo 2 “Redes Neuronales Artificiales”; Nueva Librería, Buenos Aires, Argentina.
- [Mena, J.; 1999]      Data Mining Your Website. Digital Press.
- [Nnclust; 1998]      Angshuman Saha; Tesis Doctoral *Application of Ridge Regression for Improved Estimation of Parameters in Compartmental Models*; Departamento de Estadística; Universidad de Washintong; Agosto 1998  
<http://www.geocities.com/adotsaha/NN/SOMinExcel.html>.
- [RT-ITBA; 2004].      Material del Modulo I del la carrera de Magister en Ingeniería del Software – ITBA – UPM.
- [See5/C5.0; 2004]      RuleQuest Research - datamining tools. Australia; Noviembre 2004.  
<http://www.rulequest.com/see5-info.html>.
- [Segovia, J.; 2004]      Modulo VII – Temas Complementarios de Inteligencia Artificial – Unidad 32 – Redes Neuronales Artificiales. Material MIS.
- [Viñuela, I; etal; 2004]      Redes Neuronales Artificiales – Un Enfoque Práctico. Editorial Pearson – Prentice Hall. 2004.

### 8.2. Glosario

#### Acrónimos y Abreviaturas

- DNI: Documento Nacional de Identidad.

- DSN: Data Source Name (Orígenes de datos de usuario).
- IEEE: Institute of Electrical and Electronics Engineers (Instituto de ingenieros eléctricos y electrónicos).
- NTFS: New Technology File System (sistema de archivos nueva tecnología).
- ODBC: Open DataBase Connectivity (conectividad abierta de base de datos).
- SGBD: Sistema Gestor de Base de Datos.
- SGBDR: Sistema de Gestor de Base de Datos Relacional.
- SOM: Self Organizing Maps (mapas auto organizados).
- TCP/IP: Transport Control Protocol (Protocolo de Control de Transporte) / Internet Protocol (Protocolo de Internet).

## ANEXO I – CONTROL DE CONFIGURACIÓN

Elementos de Configuración de Software:

ECS	Versión	Ultima modif.	Estado	Observaciones
DM_CAP.01_Plan de sistemas de información	1.0	08/07/2005	Aprobado	
DM_CAP.01_Plan de sistemas de información	2.0	20/07/2005	Aprobado	
<b>DM_CAP.01_Plan de sistemas de información</b>	<b>3.0</b>	<b>29/07/2005</b>	<b>Aprobado</b>	<b>Capitulo 1 de Tesis</b>
DM_CAP.02_Estudio de Viabilidad del Sistema	1.0	10/08/2005	Aprobado	
DM_CAP.02_Estudio de Viabilidad del Sistema	2.0	17/08/2005	Aprobado	
DM_CAP.02_Estudio de Viabilidad del Sistema	3.0	25/08/2005	Aprobado	
<b>DM_CAP.02_Estudio de Viabilidad del Sistema</b>	<b>4.0</b>	<b>31/08/2005</b>	<b>Aprobado</b>	<b>Capitulo 2 de Tesis</b>
DM_CAP.03_Modelo de análisis	1.0	22/09/2005	Aprobado	
DM_CAP.03_Modelo de análisis	2.0	18/10/2005	Aprobado	
DM_CAP.03_Modelo de análisis	3.0	21/11/2005	Aprobado	
DM_CAP.03_Modelo de análisis	4.0	01/12/2005	Aprobado	
DM_CAP.03_Modelo de análisis	5.0	15/12/2005	Aprobado	
<b>DM_CAP.03_Modelo de análisis</b>	<b>6.0</b>	<b>29/12/2005</b>	<b>Aprobado</b>	<b>Capitulo 3 de Tesis</b>
DM_CAP.04_Diseño del sistema de información	1.0	20/01/2006	Aprobado	
DM_CAP.04_Diseño del sistema de información	2.0	01/02/2006	Aprobado	
DM_CAP.04_Diseño del sistema de información	3.0	22/02/2006	Aprobado	
DM_CAP.04_Diseño del sistema de información	4.0	15/03/2006	Aprobado	
<b>DM_CAP.04_Diseño del sistema de información</b>	<b>5.0</b>	<b>28/04/2006</b>	<b>Aprobado</b>	<b>Capitulo 4 de Tesis</b>
DM_CAP.05_Construcción del sistema de información	1.0	22/05/2006	Aprobado	
DM_CAP.05_Construcción del sistema de información	2.0	15/06/2006	Aprobado	
DM_CAP.05_Construcción del sistema de información	3.0	06/07/2006	Aprobado	
DM_CAP.05_Construcción del sistema de información	4.0	28/07/2006	Aprobado	
DM_CAP.05_Construcción del sistema de información	5.0	14/08/2006	Aprobado	

<b>ECS</b>	<b>Versión</b>	<b>Ultima modif.</b>	<b>Estado</b>	<b>Observaciones</b>
<b>DM_CAP.05_Construcción del sistema de información</b>	<b>6.0</b>	<b>31/08/2006</b>	<b>Aprobado</b>	<b>Capitulo 5 de Tesis</b>
DM_CAP.06_Implantación y aceptación del sistema	1.0	15/09/2006	Aprobado	
DM_CAP.06_Implantación y aceptación del sistema	2.0	29/09/2006	Aprobado	
DM_CAP.06_Implantación y aceptación del sistema	3.0	05/10/2006	Aprobado	
<b>DM_CAP.06_Implantación y aceptación del sistema</b>	<b>4.0</b>	<b>13/10/2006</b>	<b>Aprobado</b>	<b>Capitulo 6 de Tesis</b>
ANX_5_Manuales de usuario	1.0	20/10/2006	Aprobado	
ANX_5_Manuales de usuario	2.0	06/11/2006	Aprobado	
<b>ANX_5_Manuales de usuario</b>	<b>3.0</b>	<b>14/11/2006</b>	<b>Aprobado</b>	
<b>Software</b>	<b>1.0</b>	<b>31/08/2006</b>	<b>Aprobado</b>	
<b>Registro de Instalación</b>	<b>1.0</b>	<b>05/10/2006</b>	<b>Aprobado</b>	
<b>Registro de Incidencias</b>	<b>1.0</b>	<b>05/10/2006</b>	<b>Aprobado</b>	

## **ANEXO II – PLAN DE ACCIÓN Y CONTINGENCIA FRENTE AL RIESGO**

### **II.1. Riesgo R01 Falla de seguridad en la aplicación**

#### **II.1.1 Aspectos a considerar**

- a. Por que el riesgo es importante.

Dadas las características del proyecto “DM”, es importante que su funcionamiento evite la perdida de confidencialidad e integración de la información.

- b. Que información se necesita para seguir el estado del riesgo.

Conocer los reportes de actividades y el registro de auditoria del sistema, en especial todo aquello relacionado con el acceso a los recursos que involucra el proyecto.

- c. Quien es el responsable de realizar las actividades del control del riesgo.

En este caso el responsable de coordinar las actividades de control de este riesgo es el autor del proyecto.

- d. Que recursos se necesitan para realizar las actividades del control del riesgo.

Se deben incorporar los recursos software de seguridad disponibles en cada funcionalidad de la aplicación que controle el acceso al sistema.

#### **II.1.2 Plan de Acción**

Para mantener controlado el riesgo R01 se deben desarrollar las siguientes actividades:

II.1.2.1 Asegurar que los canales de comunicación sean seguros.

II.1.2.2 Verificar la identidad de los usuarios del sistema.

II.1.2.3 Registrar los intentos de acceso fallidos.

II.1.2.4 Encriptar la fase de autenticación de usuarios del sistema.

II.1.2.5 Cumplir el plan de resguardo de información.

### **II.1.3 Plan de Contingencia**

En caso de que se verifique el siguiente evento disparador se adoptarán las siguientes medidas:

II.1.3.1 Disparador: Detección de intruso en el sistema.

II.1.3.2 Producir la inmediata detención del sistema, y la introducción de un aviso que indique sobre la imposibilidad de acceso al mismo.

II.1.3.3 Requerir la atención del personal de administración del sistema.

II.1.3.3 Verificar la actividad del día a efectos de determinar la magnitud de recursos potencialmente afectados.

## ANEXO III – ASEGURAMIENTO DE CALIDAD

### III.1. Listas de verificación

Lista de Verificación N° 1		
Proyecto: DM		Fecha: 04/12/2006
Integrantes: Esteban Sal		
Revisión de Requisitos		
Preguntas	Respuestas	
	Si	No
¿Los requisitos son entendibles?	✓	
¿Son factibles de implementar?	✓	
¿Existen requisitos superpuestos o que se complementan?	Ver Obs.1	
¿Son importantes para solucionar los problemas detectados?	✓	
¿Falta especificar algún requisito?		✓
<b>Observaciones:</b>		
1. Se ha determinado que los cálculos estadísticos se discriminen entre aquellos obtenidos usando los datos de las Solicitudes de la de los Pedidos.		

### Lista de Verificación N° 1

Lista de Verificación N° 2		
Proyecto: DM		Fecha: 04/12/2006
Integrantes: Esteban Sal		
Revisión de Requisitos		
Preguntas	Respuestas	
	Si	No
¿Todos los requisitos tienen asociado un caso de uso?	✓	

Lista de Verificación N° 2		
<b>Proyecto:</b> DM	<b>Fecha:</b> 04/12/2006	
<b>Integrantes:</b> Esteban Sal		
Revisión de Requisitos		
Preguntas	Respuestas	
	Si	No
¿Los casos de uso tienen su descripción completa?	✓	
¿Los casos de uso tienen su correspondiente clase de análisis?	✓	
¿Los casos de uso tienen definido su diagrama de colaboración correspondiente?	✓	
<b>Observaciones:</b>		

## Lista de Verificación N° 2

Lista de Verificación N° 3		
<b>Proyecto:</b> DM	<b>Fecha:</b> 04/12/2006	
<b>Integrantes:</b> Esteban Sal		
Revisión de Requisitos		
Preguntas	Respuestas	
	Si	No
¿Los casos de uso tienen su correspondiente diagrama de clases de análisis y de diseño?	✓	
¿Las clases de diseño tienen su correspondiente descripción?	✓	
¿Para cada requisito funcional existe su correspondiente caso de uso, diagrama de clase de diseño y su descripción?	✓	
<b>Observaciones:</b>		

## Lista de Verificación N° 3

<b>Lista de Verificación N° 4</b>		
<b>Proyecto:</b> DM	<b>Fecha:</b> 04/12/2006	
<b>Integrantes:</b> Esteban Sal		
<b>Revisión de Requisitos</b>		
<b>Preguntas</b>	<b>Respuestas</b>	
	<b>Si</b>	<b>No</b>
¿Son probados todos los requisitos?	✓	
¿Se verifican los límites de rango?	✓	
¿Se realiza la prueba de interfaz entre cada modulo?	✓	
¿Han sido verificados los valores representativos de las clases?	✓	
¿Los casos de prueba verifican todos los procesos?	✓	
<b>Observaciones:</b>		

Lista de Verificación N° 4

### III.2. Plan de Pruebas de Sistema

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

El plan de pruebas es un producto formal que define los objetivos de la prueba de un sistema, establece y coordina una estrategia de trabajo, y provee del marco adecuado para elaborar una planificación paso a paso de las actividades de prueba. El plan se inicia en el proceso Análisis del Sistema de Información (ASI), definiendo el marco general, y estableciendo los requisitos de prueba de aceptación, relacionados directamente con la especificación de requisitos.

Dicho plan se va completando y detallando a medida que se avanza en los restantes procesos del ciclo de vida del software, Diseño del Sistema de Información (DSI), Construcción del Sistema de Información (CSI) e Implantación y Aceptación del Sistema (IAS).

Se plantean los siguientes niveles de prueba:

- Pruebas unitarias.
- Pruebas de integración.
- Pruebas del sistema.
- Pruebas de implantación.
- Pruebas de aceptación.

En esta actividad también se avanza en la definición de las pruebas de aceptación del sistema. Con la información disponible, es posible establecer los criterios de aceptación de las pruebas incluidas en dicho nivel, al poseer la información sobre los requisitos que debe cumplir el sistema, recogidos en el catálogo de requisitos.

### **III.2.1. Plan de Pruebas**

La presente planificación de pruebas tiene como objetivo servir de guía para la realización de las pruebas, permitiendo verificar que el sistema construido cumple las necesidades establecidas dentro de un marco de garantía de calidad.

Para especificar las pruebas se ha adoptado el modelo de pruebas especificado en el estándar de Documentación de Pruebas de Software de la IEEE [IEEE 829,1983], el cual ha sido adaptado a las características del presente proyecto.

#### **III.2.1.1. Introducción**

El propósito de este plan es describir la estrategia, el alcance, la aproximación, el esquema de plazos y los recursos necesarios para llevar a cabo las actividades de prueba del Sistema de Explotación de Datos basados en las Metodologías SOM y C.5. Este

plan identifica los ítems a ser probados, las características de los mismos, las tareas a ser realizadas y los responsables de las mismas.

### **III.2.1.2. Alcance**

El alcance de este plan esta limitado a la definición de las actividades generales de prueba a realizarse para el Sistema de Explotación de Datos basados en las Metodologías SOM y C.5.

### **III.2.1.3. Ítems y características a probar**

El presente plan tiene como objetivo probar el Sistema de Explotación de Datos basados en las Metodologías SOM y C.5, para lo cual se estima pertinente realizar pruebas unitarias (para verificar como se realiza cada función del sistema) y una prueba global del Sistema.

### **III.2.1.4. Características que no van a ser probadas**

- Aplicaciones y herramientas no desarrolladas (por ejemplo, Sistema Operativo, soporte de impresión y almacenamiento).
- Performance del producto (por ejemplo, tiempo de respuesta de la aplicación e interfaz de usuario).
- Entorno de trabajo (por ejemplo, disponibilidad de red y tiempo de respuesta del equipo).
- Control de acceso a los documentos del proyecto por fuera del sistema.

### **III.2.2. Actividades a Realizar**

Las principales actividades de prueba a realizar están asociadas con las tareas de:

- Pruebas unitarias.
- Pruebas de sistema.

- Análisis y evaluación de la prueba.

### **III.2.2.1. Pruebas unitarias**

Esta actividad debe cubrir cada una de las clases creadas durante la etapa de codificación. Todas las entradas y salidas de una clase deben ser probadas, y en caso de existir la posibilidad de combinar varias al mismo tiempo, esto también debe ser probado.

Las pruebas unitarias deben desarrollarse de forma paralela a la codificación de la aplicación. Y solo cuando las actividades de pruebas unitarias hayan sido superadas exitosamente, se podrá pasar a la siguiente actividad de prueba.

### **III.2.2.2. Pruebas de sistema**

Las pruebas de sistema serán orientadas según la técnica de “caja negra”; utilizado particularmente los métodos de partición de equivalencias y análisis de valores límites. Una prueba de caja negra examina algunos aspectos externos del modelo del sistema sin tener en cuenta la estructura lógica interna del software.

Una vez que todos los casos de prueba han sido superados exitosamente, la aplicación estará lista para ser entregada.

### **III.2.2.3. Pruebas de aceptación**

Estas pruebas serán realizadas por la Directora del Proyecto, quien tomará como criterio de evaluación el cumplimiento, por parte del sistema, de los requisitos funcionales del mismo.

### **III.2.2.4. Pruebas unitarias**

Los resultados de las pruebas unitarias deberán ser almacenados en el documento de registro estándar y la ejecución de los mismos será manual.

### III.2.2.5. Pruebas de sistema

Los resultados de las pruebas deberán ser almacenados en el documento de registro estándar y la ejecución de las mismas será manual.

### III.2.2.6. Amplitud de las pruebas

La amplitud y criterio de completitud a emplear se basará en la cobertura realizada sobre la funcionalidad requerida.

### III.2.3. Reporte de fallas de las pruebas

Las fallas serán identificadas durante el análisis y evaluación de los resultados de la ejecución de las pruebas. A continuación, en la Figura III.1, se detalla el formato del reporte de pruebas a utilizar:

Reporte de Prueba N°.: _____		Fecha de Prueba ___/___/___
Objetivo Probado: _____		
_____		
_____		
Errores Encontrados:		
Id. Caso de Prueba	Nivel de Severidad	Descripción

Figura III.1: Documento de reporte de pruebas

### **III.2.4. Trazabilidad de requerimientos**

En el punto Análisis de consistencia entre métodos se detallan las matrices de trazabilidad que muestran la consistencia entre los distintos modelos generados en la fase de análisis.

#### **III.2.4.1. Disponibilidad de ítems de prueba**

Las actividades de prueba no podrán comenzar hasta que todas las unidades de prueba se encuentren disponibles.

#### **III.2.4.2. Disponibilidad de recursos para las pruebas**

A continuación se detallan los elementos necesarios para llevar a delante las pruebas unitarias y de sistema:

1 PC equipada mínimamente con:

- Procesador Pentium II
- 128 Mb de Memoria RAM
- 10 Mb libres en el disco rígido
- 1 Impresora (opcional).
- Sistema operativo Windows 2000 o XP.
- Conexión a Base de Datos vía ODBC (opcional).

### **III.2.5. Criterio de Paso/Falla**

A continuación se detallan los criterios a aplicar en la evaluación de las distintas instancias de prueba:

#### **III.2.5.1. Ítems**

El criterio a emplear sobre cada uno de los ítems es el siguiente:

**Paso:** Todas las pruebas realizadas sobre el ítem fueron exitosas.

**Fallo:** Al menos una de las pruebas realizadas no fue exitosa.

### III.2.5.2. Aplicación

El criterio a emplear sobre las pruebas de la aplicación es el siguiente:

- **Exitosa:** Todas las pruebas fueron realizados y no se encontraron defectos de severidad 1, 2 o 3. (Véase Tabla III.2 – Severidad de Defectos).
- **Fallida:** Al menos un defecto de severidad 1, 2 o 3 fue encontrado. (Véase Tabla III.2 - Severidad de Defectos).

Nivel de Severidad	Descripción
1	Sistema detenido
2	Fallas de funcionalidad
3	Una solución alternativa puede aplicarse
4	Error de documentación/ayuda
5	Cambios y mejoras

Tabla III.2: Severidad de Defectos

### III.2.6. Criterio de suspensión y reiniciación de pruebas

Las actividades de prueba deberían ser suspendidas si ocurre alguna de las situaciones:

- Se encuentra un problema en una prueba que impide la realización de la prueba.
- Se encuentra un problema en un ítem que impide la realización de más pruebas.

Cuando se encuentre un problema y el mismo no impida la realización de más pruebas, las mismas deben continuar.

Una vez solucionados los problemas encontrados, las pruebas deben reiniciarse, comenzando nuevamente por el primer caso de prueba para verificar que la solución del problema no haya generado nuevos inconvenientes.

### **III.2.7. Artefactos de las pruebas**

A continuación se detallan los elementos a generar como resultado de la realización de las pruebas:

- Plan de pruebas y Documento de diseño de la prueba.
- Especificación de los casos de prueba y Especificación del procedimiento de prueba.
- Informe de los casos de prueba.
- Informe de la prueba.

### **III.2.8. Actividades de prueba**

A continuación se detallan las actividades a realizarse para concretar cada uno de los ciclos de prueba:

- Actualizar el plan de pruebas y documento de diseño.
- Crear o actualizar casos y procedimientos de prueba.
- Ejecutar las pruebas y realizar el análisis, evaluación e informe de las mismas.
- Llevar a cabo la prueba de aceptación.

### **III.2.9. Procedimiento de pruebas**

A continuación, en la Figura III.2, se detalla el esquema de procedimiento de pruebas:

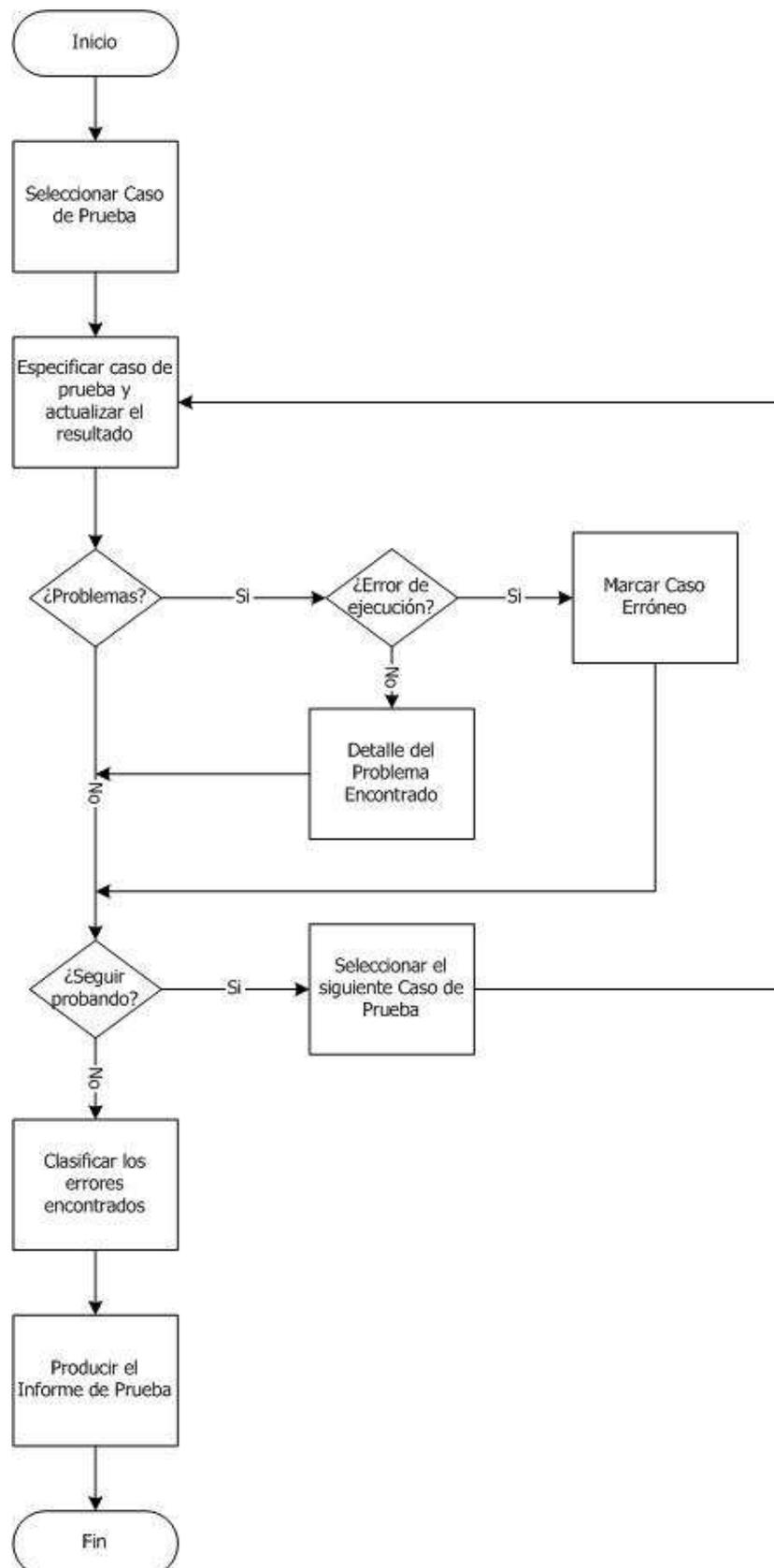


Figura III.2: Procedimiento de Pruebas

### **III.2.10. Necesidades de ambiente**

Existen dos entornos principales relacionados con las pruebas:

- Entorno local de pruebas.
- Entorno del cliente.

Ambos deben ser coincidentes y las propiedades mínimas requeridas para los mismos se especifican a continuación.

#### **III.2.10.1. Hardware**

PC Pentium II o superior, 128 MB de RAM y 10 MB libres en el disco rígido.

#### **III.2.10.2. Software**

ODBC para conectarse con algún repositorio de datos en caso de ser necesario.

#### **III.2.10.3. Seguridad**

El usuario debe contar con permisos de lectura/escritura sobre el directorio donde se ejecutará la aplicación y donde se desea ubicar los proyectos y versiones que se crean con el sistema.

#### **III.2.10.4. Modo de uso**

Referirse a Definición del Sistema y Establecimiento de Requisitos.

#### **III.2.10.5. Certificación de entorno**

No existen necesidades concretas de certificación de entorno.

### **III.2.11. Responsabilidades**

A continuación se detallan todos los roles a cubrir durante el desarrollo y la ejecución de las pruebas, estos roles, dentro del presente proyecto, serán llevados a cabo por el Tesista y la Directora de tesis, ya que los mismos son los únicos recursos

humanos asignados al proyecto.

Ingeniero de pruebas (Esta función está a cargo del tesista)

- Diseñar, preparar y realizar las actividades de prueba.
- Gestionar el entorno local de prueba.
- Recolectar y analizar los resultados obtenidos de la realización de las actividades de prueba.

Líder de desarrollo (Esta función está a cargo del tesista)

- Proveer de los ítems a probar.
- Revisar el plan de pruebas.
- Revisar las actividades de prueba.
- Revisar los resultados de las pruebas realizadas.

Líder de proyecto (Esta función está a cargo del tesista)

- Revisar el plan de pruebas.
- Revisar los resultados de las pruebas realizadas.
- Coordinar las actividades de prueba.

Especialista en calidad (Esta función está a cargo de la Directora de Tesis)

- Revisar el plan de pruebas.
- Revisar las actividades de prueba.

Cliente (Esta función está a cargo de la Directora de Tesis)

- Realizar las pruebas de aceptación.

### III.2.12. Riesgos y contingencias de pruebas

A continuación, en la tabla III.3, de detallan posibles riesgos a enfrentar en la implementación del sistema con sus respectivas contingencias

<b>Riesgos</b>	<b>Contingencias</b>
Diferencias entre el entorno de desarrollo y de pruebas y el entorno del cliente	Especificar como parte de la documentación a entregar los requerimientos de software y hardware y colaborar con el usuario en la verificación/instalación de los mismos
Falta de conocimiento del usuario de la herramienta para ejecutar la aplicación	Especificar como parte de la documentación a entregar los pasos para ejecutar el sistema y colaborar con el usuario en la realización de los mismos

Tabla III.3: Análisis de Riesgos y Contingencias