



UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA EN INFORMÁTICA

PROYECTO FIN DE CARRERA

**FRAMEWORK PARA LA CONEXIÓN DE SISTEMAS
HÍBRIDOS BASADOS EN REDES DE NEURONAS Y
SISTEMAS EXPERTOS**

Autor: *David Martín Muñoz*

Tutores: *Israel González Carrasco
José Luis López Cuadrado*

Octubre 2009



Tabla de contenido

ÍNDICE DE TABLAS	5
ÍNDICE DE FIGURAS	10
1 INTRODUCCIÓN	12
2 DEFINICIONES Y ACRÓNIMOS	15
2.1 DEFINICIONES	15
2.2 ACRÓNIMOS	17
3 ESTUDIO DE VIABILIDAD DEL SISTEMA.....	18
3.1 INTRODUCCIÓN	18
3.1.1 <i>Propósito del plan</i>	18
3.2 ESTABLECIMIENTO DEL ALCANCE DEL SISTEMA.....	19
3.2.1 <i>Estudio de la solicitud</i>	19
3.2.2 <i>Identificación del alcance del sistema</i>	20
3.2.3 <i>Identificación de los interesados en el sistema (stakeholders)</i>	21
3.3 ESTUDIO DE LA SITUACIÓN ACTUAL	22
3.3.1 <i>Valoración del estudio de la situación actual</i>	22
3.3.2 <i>Realización del diagnóstico actual</i>	23
3.4 DEFINICIÓN DE LOS REQUISITOS DEL SISTEMA.....	24
3.4.1 <i>Especificación de los casos de uso</i>	24
3.4.2 <i>Definición de los requisitos del sistema</i>	36
3.5 ESTUDIO DE ALTERNATIVAS DE SOLUCIÓN.....	55
3.6 VALORACIÓN DE LAS ALTERNATIVAS	56
3.7 SELECCIÓN DE LA SOLUCIÓN	56
4 GESTIÓN DEL PROYECTO.....	57
4.1 CICLO DE VIDA.....	57
4.2 ORGANIZACIÓN DEL PROYECTO	59
4.3 PLANIFICACIÓN.....	66
4.4 ESTIMACIÓN DE COSTES.....	71
5 ANÁLISIS DEL SISTEMA.....	73
5.1 INTRODUCCIÓN	73
5.1.1 <i>Objetivo del Análisis del Sistema</i>	73
5.1.2 <i>Alcance</i>	73
5.2 DEFINICIÓN DEL SISTEMA	74
5.2.1 <i>Determinación del alcance del sistema</i>	74
5.2.2 <i>Identificación del entorno tecnológico</i>	74
5.2.3 <i>Especificación de estándares y normas</i>	75
5.3 ESTABLECIMIENTO DE REQUISITOS SOFTWARE	78
5.3.1 <i>Obtención de requisitos</i>	78



5.4	IDENTIFICACIÓN DE SUBSISTEMAS DE ANÁLISIS	102
5.5	ANÁLISIS DE LOS CASOS DE USO.....	103
5.5.1	<i>Identificación de clases asociadas a un caso de uso</i>	<i>103</i>
5.5.2	<i>Descripción de la interacción de objetos</i>	<i>104</i>
5.6	ANÁLISIS DE CLASES.....	107
5.6.1	<i>Identificación de responsabilidades y atributos</i>	<i>109</i>
5.6.2	<i>Identificación de asociaciones</i>	<i>116</i>
5.6.3	<i>Identificación de generalizaciones.....</i>	<i>120</i>
5.7	DEFINICIÓN DE LAS INTERFACES DE USUARIO.....	122
5.7.1	<i>Especificación de principios generales de la interfaz</i>	<i>122</i>
5.7.2	<i>Identificación de perfiles y diálogos</i>	<i>122</i>
5.7.3	<i>Especificación de formatos individuales de la interfaz de la pantalla.....</i>	<i>122</i>
5.8	ANÁLISIS DE CONSISTENCIA Y ESPECIFICACIÓN DE REQUISITOS.....	128
5.8.1	<i>Verificación y análisis de consistencia entre modelos.....</i>	<i>128</i>
5.8.2	<i>Validación de los modelos</i>	<i>131</i>
6	DISEÑO DEL SISTEMA.....	132
6.1	INTRODUCCIÓN	132
6.1.1	<i>Propósito.....</i>	<i>132</i>
6.2	ALCANCE.....	133
6.3	DEFINICIÓN DE LA ARQUITECTURA	135
6.3.1	<i>Definición de los niveles de la arquitectura.....</i>	<i>135</i>
6.3.2	<i>Especificación de estándares y normas de diseño y construcción.....</i>	<i>135</i>
6.3.3	<i>Identificación de los subsistemas de la aplicación</i>	<i>137</i>
6.3.4	<i>Especificación del entorno tecnológico</i>	<i>137</i>
6.3.5	<i>Especificación de requisitos de seguridad y operación</i>	<i>138</i>
6.3.6	<i>Estudio de la seguridad requerida en el proceso de diseño del sistema ..</i>	<i>138</i>
6.3.7	<i>Análisis de los riesgos del entorno tecnológico</i>	<i>138</i>
6.4	DISEÑO DE LA ARQUITECTURA DE SOPORTE.....	139
6.5	DISEÑO DE CASOS DE USO REALES.....	143
6.5.1	<i>Identificación de clases asociadas a un caso de uso</i>	<i>143</i>
6.5.2	<i>Diseño de la realización de los casos de uso.....</i>	<i>144</i>
6.6	DISEÑO DE CLASES.....	147
6.6.1	<i>Identificación de clases de diseño.....</i>	<i>147</i>
6.6.2	<i>Identificación de atributos y métodos de clases de diseño</i>	<i>149</i>
6.7	DISEÑO FÍSICO DE DATOS	165
6.8	VERIFICACIÓN Y ACEPTACIÓN DE LA ARQUITECTURA DEL SISTEMA	166
6.8.1	<i>Análisis de consistencia de las especificaciones de diseño.....</i>	<i>166</i>
6.9	ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS.....	168
6.10	ESTABLECIMIENTO DE REQUISITOS DE IMPLANTACIÓN	169
6.10.1	<i>Especificación de Requisitos de Implantación</i>	<i>169</i>
7	PLAN DE VALIDACIÓN Y VERIFICACIÓN DEL SOFTWARE	170
7.1	INTRODUCCIÓN	170
7.1.1	<i>Propósito del documento.....</i>	<i>170</i>



7.1.2	<i>Visión general del Plan de Validación y Verificación del Sistema</i>	171
7.2	PLAN DE PRUEBA	172
7.2.1	<i>Elementos de prueba</i>	172
7.2.2	<i>Características que se probarán</i>	173
7.2.3	<i>Entregables</i>	173
7.2.4	<i>Tareas de prueba</i>	174
7.2.5	<i>Necesidades del entorno</i>	175
7.2.6	<i>Criterio de aceptación/rechazo de un caso de prueba</i>	175
7.3	PROCEDIMIENTOS DE PRUEBA.....	176
7.3.1	<i>Aplicación de tratamiento de ficheros</i>	176
7.3.2	<i>Simulador del sistema experto</i>	192
7.3.3	<i>Módulo de comunicación del sistema híbrido</i>	194
7.4	PLANTILLA DE INFORME DE PRUEBAS	204
7.5	MATRIZ DE TRAZABILIDAD PRUEBAS/REQUISITOS DE CAPACIDAD	205
8	IMPLEMENTACIÓN	208
8.1	APLICACIÓN PARA TRATAMIENTO DE FICHEROS.....	208
8.2	MÓDULO DE COMUNICACIÓN	210
8.3	SIMULADOR.....	211
8.4	INTERACCIÓN ENTRE MÓDULOS.....	213
9	PRUEBAS DE DOMINIO	215
9.1	PRUEBA DE DOMINIO A: CLASIFICACIÓN	215
9.2	PRUEBA DE DOMINIO B: REGRESIÓN	220
10	CONCLUSIONES	225
11	BIBLIOGRAFÍA	226
12	ANEXO A: MANUAL DE USUARIO APLICACIÓN TRATAMIENTO DE FICHEROS ..	228
12.1	ARRANQUE DE LA APLICACIÓN	228
12.2	LOCALIZACIÓN DE FICHEROS	230
12.2.1	<i>Fichero de entrada</i>	230
12.2.2	<i>Ruta de salida</i>	231
12.3	IMPORTAR DATOS	233
12.3.1	<i>Selección de atributos</i>	234
12.3.2	<i>Atributos de salida</i>	236
12.4	CONFIGURACIÓN DE EXPERIMENTOS	237
12.4.1	<i>Técnica de división</i>	237
12.4.2	<i>Número de pruebas</i>	239
12.4.3	<i>Conjuntos de instancias</i>	240
12.4.4	<i>Reparto de instancias</i>	240
12.4.5	<i>Añadir experimentos a preparar</i>	243
12.4.6	<i>Añadir experimentos de forma automatizada</i>	244
12.4.7	<i>Eliminar experimentos</i>	246
12.5	GENERACIÓN DE FICHEROS.....	248
12.6	GESTIÓN DE LA CONFIGURACIÓN	251



12.6.1	<i>Guardar configuración</i>	251
12.6.2	<i>Cargar configuración</i>	252
13	ANEXO B: FORMATO PARA FICHEROS DE ENTRADA	253
14	ANEXO C: CREACIÓN DE UNA RED DE NEURONAS EN NEUROSOLUTIONS	254
15	ANEXO D: MANUAL DE USUARIO MÓDULO DE COMUNICACIÓN	264
15.1	REQUISITOS DE FUNCIONAMIENTO	264
15.2	EJECUCIÓN	264
15.3	COMPROBACIÓN DE ERRORES	265
16	ANEXO E: MANUAL DE USUARIO SIMULADOR	266
16.1	ARRANQUE DE LA APLICACIÓN	266
16.2	INTRODUCCIÓN DE PARÁMETROS DE ENTRADA	268
16.3	EJECUCIÓN DEL SISTEMA.....	269



Índice de Tablas

TABLA 1. CASO DE USO CU-001	26
TABLA 2. CASO DE USO CU-002	27
TABLA 3. CASO DE USO CU-003	27
TABLA 4. CASO DE USO CU-004	28
TABLA 5. CASO DE USO CU-005	28
TABLA 6. CASO DE USO CU-006	29
TABLA 7. CASO DE USO CU-007	29
TABLA 8. CASO DE USO CU-008	30
TABLA 9. CASO DE USO CU-009	30
TABLA 10. CASO DE USO CU-010	31
TABLA 11. CASO DE USO CU-011	31
TABLA 12. CASO DE USO CU-012	32
TABLA 13. CASO DE USO CU-013	33
TABLA 14. CASO DE USO CU-014	34
TABLA 15. CASO DE USO CU-015	34
TABLA 16. CASO DE USO CU-016	35
TABLA 17. CASO DE USO CU-017	35
TABLA 18. REQUISITO DE USUARIO RU-C-001	38
TABLA 19. REQUISITO DE USUARIO RU-C-002	39
TABLA 20. REQUISITO DE USUARIO RU-C-003	39
TABLA 21. REQUISITO DE USUARIO RU-C-004	40
TABLA 22. REQUISITO DE USUARIO RU-C-005	40
TABLA 23. REQUISITO DE USUARIO RU-C-006	41
TABLA 24. REQUISITO DE USUARIO RU-C-007	41
TABLA 25. REQUISITO DE USUARIO RU-C-008	42
TABLA 26. REQUISITO DE USUARIO RU-C-009	42
TABLA 27. REQUISITO DE USUARIO RU-C-010	43
TABLA 28. REQUISITO DE USUARIO RU-C-011	43
TABLA 29. REQUISITO DE USUARIO RU-C-012	44
TABLA 30. REQUISITO DE USUARIO RU-C-013	44
TABLA 31. REQUISITO DE USUARIO RU-C-014	45
TABLA 32. REQUISITO DE USUARIO RU-C-015	45
TABLA 33. REQUISITO DE USUARIO RU-C-016	46
TABLA 34. REQUISITO DE USUARIO RU-C-017	46
TABLA 35. REQUISITO DE USUARIO RU-C-018	47
TABLA 36. REQUISITO DE USUARIO RU-C-019	47
TABLA 37. REQUISITO DE USUARIO RU-C-020	48
TABLA 38. REQUISITO DE USUARIO RU-C-021	48
TABLA 39. REQUISITO DE USUARIO RU-C-022	49
TABLA 40. REQUISITO DE USUARIO RU-R-001	49
TABLA 41. REQUISITO DE USUARIO RU-R-002	50
TABLA 42. REQUISITO DE USUARIO RU-R-003	50



TABLA 43. REQUISITO DE USUARIO RU-R-004	51
TABLA 44. REQUISITO DE USUARIO RU-R-005	51
TABLA 45. REQUISITO DE USUARIO RU-R-006	52
TABLA 46. REQUISITO DE USUARIO RU-R-007	52
TABLA 47. REQUISITO DE USUARIO RU-R-008	53
TABLA 48. REQUISITO DE USUARIO RU-R-009	53
TABLA 49. REQUISITO DE USUARIO RU-R-010	54
TABLA 50. REQUISITO DE USUARIO RU-R-011	54
TABLA 51. COSTES DE LICENCIAS.....	71
TABLA 52. COSTES DE RECURSOS HUMANOS	71
TABLA 53. COSTE TOTAL DEL PROYECTO	72
TABLA 54. REQUISITO SOFTWARE RS-F-001	80
TABLA 55. REQUISITO SOFTWARE RS-F-002	80
TABLA 56. REQUISITO SOFTWARE RS-F-003	81
TABLA 57. REQUISITO SOFTWARE RS-F-004	81
TABLA 58. REQUISITO SOFTWARE RS-F-005	82
TABLA 59. REQUISITO SOFTWARE RS-F-006	82
TABLA 60. REQUISITO SOFTWARE RS-F-007	83
TABLA 61. REQUISITO SOFTWARE RS-F-008	83
TABLA 62. REQUISITO SOFTWARE RS-F-009	84
TABLA 63. REQUISITO SOFTWARE RS-F-010	84
TABLA 64. REQUISITO SOFTWARE RS-F-011	85
TABLA 65. REQUISITO SOFTWARE RS-F-012	85
TABLA 66. REQUISITO SOFTWARE RS-F-013	86
TABLA 67. REQUISITO SOFTWARE RS-F-014	86
TABLA 68. REQUISITO SOFTWARE RS-F-015	87
TABLA 69. REQUISITO SOFTWARE RS-F-016	87
TABLA 70. REQUISITO SOFTWARE RS-F-017	88
TABLA 71. REQUISITO SOFTWARE RS-F-018	88
TABLA 72. REQUISITO SOFTWARE RS-F-019	89
TABLA 73. REQUISITO SOFTWARE RS-F-020	89
TABLA 74. REQUISITO SOFTWARE RS-F-021	90
TABLA 75. REQUISITO SOFTWARE RS-F-022	90
TABLA 76. REQUISITO SOFTWARE RS-F-023	91
TABLA 77. REQUISITO SOFTWARE RS-F-024	91
TABLA 78. REQUISITO SOFTWARE RS-F-025	92
TABLA 79. REQUISITO SOFTWARE RS-F-026	92
TABLA 80. REQUISITO SOFTWARE RS-F-027	93
TABLA 81. REQUISITO SOFTWARE RS-F-028	94
TABLA 82. REQUISITO SOFTWARE RS-I-001	94
TABLA 83. REQUISITO SOFTWARE RS-I-002	95
TABLA 84. REQUISITO SOFTWARE RS-O-001	95
TABLA 85. REQUISITO SOFTWARE RS-O-002	96
TABLA 86. REQUISITO SOFTWARE RS-O-003	96
TABLA 87. REQUISITO SOFTWARE RS-O-004	97



TABLA 88. REQUISITO SOFTWARE RS-O-005	97
TABLA 89. REQUISITO SOFTWARE RS-O-006	98
TABLA 90. REQUISITO SOFTWARE RS-O-007	98
TABLA 91. REQUISITO SOFTWARE RS-O-008	99
TABLA 92. REQUISITO SOFTWARE RS-O-009	99
TABLA 93. REQUISITO SOFTWARE RS-D-001.....	100
TABLA 94. REQUISITO SOFTWARE RS-C-001	100
TABLA 95. REQUISITO SOFTWARE RS-C-002	101
TABLA 96. CLASES ASOCIADAS A CASOS DE USO (I)	103
TABLA 97. CLASES ASOCIADAS A CASOS DE USO (II)	104
TABLA 98. CLASES ASOCIADAS A CASOS DE USO (III)	104
TABLA 99. CLASE INTERFAZPRINCIPAL	109
TABLA 100. CLASE INTERFAZIMPORTACION	109
TABLA 101. CLASE INTERFAZESPERA	110
TABLA 102. CLASE GESTORCONFIGURACION	110
TABLA 103. CLASE GESTOREXPERIMENTO	110
TABLA 104. CLASE DIVISIONKFCV	111
TABLA 105. CLASE DIVISIONLOOCV	111
TABLA 106. CLASE DIVISIONRBP	112
TABLA 107. CLASE DIVISIONRBPFIJO	112
TABLA 108. CLASE DIVISIONVKFCV	113
TABLA 109. CLASE INTERFAZPRINCIPAL	113
TABLA 110. CLASE PROCESADOR.....	114
TABLA 111. CLASE COMUNICACION	114
TABLA 112. CLASE EVALUACION.....	114
TABLA 113. CLASE PROCESO	115
TABLA 114. CLASE ENTRENAMIENTO.....	115
TABLA 115. CLASE VALIDACION	116
TABLA 116. CLASE PRODUCCION.....	116
TABLA 117. ASOCIACIÓN CLASES INTERFAZPRINCIPAL E INTERFAZIMPORTACION	116
TABLA 118. ASOCIACIÓN CLASES INTERFAZPRINCIPAL E INTERFAZESPERA	117
TABLA 119. ASOCIACIÓN CLASES INTERFAZPRINCIPAL Y GESTORCONFIGURACION	117
TABLA 120. ASOCIACIÓN CLASES INTERFAZPRINCIPAL Y GESTOREXPERIMENTO	117
TABLA 121. ASOCIACIÓN CLASES GESTOREXPERIMENTO Y DIVISIONKFCV.....	117
TABLA 122. ASOCIACIÓN CLASES GESTOREXPERIMENTO Y DIVISIONLOOCV	118
TABLA 123. ASOCIACIÓN CLASES GESTOREXPERIMENTO Y DIVISIONRBP.....	118
TABLA 124. ASOCIACIÓN CLASES GESTOREXPERIMENTO Y DIVISIONRBPFIJO	118
TABLA 125. ASOCIACIÓN CLASES GESTOREXPERIMENTO Y DIVISIONVKFCV	118
TABLA 126. ASOCIACIÓN CLASES INTERFAZPRINCIPAL Y PROCESADOR.....	119
TABLA 127. ASOCIACIÓN CLASES COMUNICACION Y PROCESO	119
TABLA 128. ASOCIACIÓN CLASES PROCESO Y EVALUACION	119
TABLA 129. GENERALIZACIÓN CLASE FUNCION	120
TABLA 130. GENERALIZACIÓN CLASE ENTRENAMIENTO	120
TABLA 131. GENERALIZACIÓN CLASE VALIDACION	121
TABLA 132. GENERALIZACIÓN CLASE PRODUCCION	121



TABLA 133. MATRIZ DE TRAZABILIDAD REQUISITOS DE USUARIO - CASOS DE USO.....	129
TABLA 134. MATRIZ DE TRAZABILIDAD REQUISITOS SOFTWARE - CASOS DE USO	130
TABLA 135. COMPONENTE C-001: SISTEMA EXPERTO	140
TABLA 136. COMPONENTE C-002: MÓDULO DE COMUNICACIÓN	141
TABLA 137. COMPONENTE C-003: REDES DE NEURONAS ARTIFICIALES.....	142
TABLA 138. CLASES DE DISEÑO ASOCIADAS A CASOS DE USO (I)	143
TABLA 139. CLASES DE DISEÑO ASOCIADAS A CASOS DE USO (II)	144
TABLA 140. CLASES DE DISEÑO ASOCIADAS A CASOS DE USO (III)	144
TABLA 141. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE INTERFAZPRINCIPAL	149
TABLA 142. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE INTERFAZIMPORTACION	150
TABLA 143. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE INTERFAZESPERA	150
TABLA 144. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE GESTORCONFIGURACION	151
TABLA 145. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE GESTOREXPERIMENTO.....	152
TABLA 146. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE DIVISIONKFCV	153
TABLA 147. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE DIVISIONLOOCV	154
TABLA 148. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE DIVISIONRBP	155
TABLA 149. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE DIVISIONRBPFIJO	156
TABLA 150. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE DIVISIONVKFCV	157
TABLA 151. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE INTERFAZPRINCIPAL	158
TABLA 152. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE PROCESADOR.....	159
TABLA 153. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE COMUNICACION	160
TABLA 154. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE EVALUACION.....	161
TABLA 155. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE PROCESO	162
TABLA 156. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE ENTRENAMIENTO.....	163
TABLA 157. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE VALIDACION	163
TABLA 158. ATRIBUTOS, FUNCIONES Y PROCEDIMIENTOS CLASE PRODUCCION.....	164
TABLA 159. MATRIZ DE TRAZABILIDAD CLASES DE DISEÑO - CASOS DE USO	167
TABLA 160. REQUISITO SOFTWARE RS-IMP-001	169
TABLA 161. PRUEBA PR-001.....	176
TABLA 162. PRUEBA PR-002.....	177
TABLA 163. PRUEBA PR-003.....	177
TABLA 164. PRUEBA PR-004.....	178
TABLA 165. PRUEBA PR-005.....	178
TABLA 166. PRUEBA PR-006.....	179
TABLA 167. PRUEBA PR-007.....	179
TABLA 168. PRUEBA PR-008.....	179
TABLA 169. PRUEBA PR-009.....	180
TABLA 170. PRUEBA PR-010.....	180
TABLA 171. PRUEBA PR-011.....	180
TABLA 172. PRUEBA PR-012.....	181
TABLA 173. PRUEBA PR-013.....	181
TABLA 174. PRUEBA PR-014.....	181
TABLA 175. PRUEBA PR-015.....	182
TABLA 176. PRUEBA PR-016.....	182
TABLA 177. PRUEBA PR-017.....	183



TABLA 178. PRUEBA PR-018.....	183
TABLA 179. PRUEBA PR-019.....	184
TABLA 180. PRUEBA PR-020.....	184
TABLA 181. PRUEBA PR-021.....	185
TABLA 182. PRUEBA PR-022.....	185
TABLA 183. PRUEBA PR-023.....	186
TABLA 184. PRUEBA PR-024.....	186
TABLA 185. PRUEBA PR-025.....	187
TABLA 186. PRUEBA PR-026.....	187
TABLA 187. PRUEBA PR-027.....	188
TABLA 188. PRUEBA PR-028.....	188
TABLA 189. PRUEBA PR-029.....	189
TABLA 190. PRUEBA PR-030.....	189
TABLA 191. PRUEBA PR-031.....	189
TABLA 192. PRUEBA PR-032.....	190
TABLA 193. PRUEBA PR-033.....	190
TABLA 194. PRUEBA PR-034.....	191
TABLA 195. PRUEBA PR-035.....	191
TABLA 196. PRUEBA PR-036.....	191
TABLA 197. PRUEBA PR-037.....	192
TABLA 198. PRUEBA PR-038.....	192
TABLA 199. PRUEBA PR-039.....	192
TABLA 200. PRUEBA PR-040.....	193
TABLA 201. PRUEBA PR-041.....	193
TABLA 202. PRUEBA PR-042.....	193
TABLA 203. PRUEBA PR-043.....	194
TABLA 204. PRUEBA PR-044.....	194
TABLA 205. PRUEBA PR-045.....	195
TABLA 206. PRUEBA PR-046.....	196
TABLA 207. PRUEBA PR-047.....	197
TABLA 208. PRUEBA PR-048.....	197
TABLA 209. PRUEBA PR-049.....	198
TABLA 210. PRUEBA PR-050.....	199
TABLA 211. PRUEBA PR-051.....	200
TABLA 212. PRUEBA PR-052.....	201
TABLA 213. PRUEBA PR-053.....	201
TABLA 214. PRUEBA PR-054.....	202
TABLA 215. PRUEBA PR-055.....	202
TABLA 216. PRUEBA PR-056.....	203
TABLA 217. PRUEBA PR-057.....	203
TABLA 218. PLANTILLA DE INFORME DE PRUEBAS.....	204
TABLA 219. MATRIZ DE TRAZABILIDAD PRUEBAS/REQUISITOS DE CAPACIDAD.....	207
TABLA 220. INSTANCIA DE PRUEBA DE CLASIFICACIÓN.....	216
TABLA 221. INSTANCIA DE PRUEBA DE REGRESIÓN.....	221
TABLA 222. PARÁMETROS DE ENTRADA VÁLIDOS.....	265



Índice de Figuras

FIGURA 1. DIAGRAMA DE CASOS DE USO MÓDULO DE FICHEROS	24
FIGURA 2. DIAGRAMA DE CASOS DE USO SIMULADOR	25
FIGURA 3. DIAGRAMA DE CASOS DE USO MÓDULO DE COMUNICACIÓN	25
FIGURA 4. CICLO DE VIDA	57
FIGURA 5. RBS	60
FIGURA 6. PROCESO WBS - ESTUDIO DE VIABILIDAD	61
FIGURA 7. PROCESO WBS – GESTIÓN DEL PROYECTO.....	61
FIGURA 8. PROCESO WBS – PRIMER PROTOTIPO	62
FIGURA 9. PROCESO WBS – SEGUNDO PROTOTIPO	63
FIGURA 10. PROCESO WBS – TERCER PROTOTIPO.....	64
FIGURA 11. PROCESO WBS – MANUAL DE USUARIO	65
FIGURA 12. DIAGRAMA DE GANTT (GENERAL)	66
FIGURA 13. DIAGRAMA DE GANTT (ESTUDIO DE VIABILIDAD).....	66
FIGURA 14. DIAGRAMA DE GANTT (GESTIÓN DEL PROYECTO)	67
FIGURA 15. DIAGRAMA DE GANTT (PRIMER PROTOTIPO).....	68
FIGURA 16. DIAGRAMA DE GANTT (SEGUNDO PROTOTIPO).....	69
FIGURA 17. DIAGRAMA DE GANTT (TERCER PROTOTIPO)	69
FIGURA 18. DIAGRAMA DE GANTT (MANUAL DE USUARIO).....	70
FIGURA 19. DIAGRAMA DE GANTT (ENTREGA)	70
FIGURA 20. DIAGRAMA DE SECUENCIA CASO DE USO CU-006.....	105
FIGURA 21. DIAGRAMA DE SECUENCIA CASO DE USO CU-012.....	106
FIGURA 22. DIAGRAMA DE SECUENCIA CASO DE USO CU-014.....	106
FIGURA 23. DIAGRAMA CONCEPTUAL DE CLASES DEL MÓDULO DE FICHEROS	107
FIGURA 24. DIAGRAMA CONCEPTUAL DE CLASES DEL SIMULADOR	108
FIGURA 25. DIAGRAMA CONCEPTUAL DE CLASES DEL MÓDULO DE COMUNICACIÓN	108
FIGURA 26. INTERFAZ PRINCIPAL APLICACIÓN TRATAMIENTO DE FICHEROS	123
FIGURA 27. INTERFAZ DE IMPORTACIÓN - APLICACIÓN TRATAMIENTO DE FICHEROS	125
FIGURA 28. INTERFAZ PRINCIPAL DEL SIMULADOR DEL SISTEMA EXPERTO	126
FIGURA 29. COMPONENTES DE LA ARQUITECTURA DEL SISTEMA	139
FIGURA 30. DIAGRAMA DE SECUENCIA EXTENDIDO DEL CASO DE USO CU-006	145
FIGURA 31. DIAGRAMA DE SECUENCIA EXTENDIDO DEL CASO DE USO CU-012	145
FIGURA 32. DIAGRAMA DE SECUENCIA EXTENDIDO DEL CASO DE USO CU-014	146
FIGURA 33. DIAGRAMA DE CLASES DE DISEÑO (I)	147
FIGURA 34. DIAGRAMA DE CLASES DE DISEÑO (II)	148
FIGURA 35. DIAGRAMA DE CLASES DE DISEÑO (III)	148
FIGURA 36. EJEMPLO DE FICHERO DE CONFIGURACIÓN	165
FIGURA 37. INTERFAZ APLICACIÓN TRATAMIENTO DE FICHEROS	209
FIGURA 38. INTERFAZ SIMULADOR.....	211
FIGURA 39. INTERACCIÓN ENTRE MÓDULOS DEL SISTEMA	213
FIGURA 40. PARÁMETROS DE ENTRADA DEL SISTEMA EXPERTO.....	217
FIGURA 41. EJECUCIÓN EN PROCESO	218



FIGURA 42. RESULTADO DE LA EJECUCIÓN	219
FIGURA 43. PARÁMETROS DE ENTRADA DEL SISTEMA EXPERTO	222
FIGURA 44. EJECUCIÓN EN PROCESO	223
FIGURA 45. RESULTADO DE LA EJECUCIÓN	224
FIGURA 46. ARRANQUE DE LA APLICACIÓN.....	228
FIGURA 47. INTERFAZ INICIAL DE LA APLICACIÓN	229
FIGURA 48. SELECCIONANDO UN FICHERO DE ENTRADA.....	230
FIGURA 49. INTRODUCIENDO MANUALMENTE UNA RUTA DE SALIDA	231
FIGURA 50. SELECCIONANDO UNA RUTA DE SALIDA	232
FIGURA 51. IMPORTAR DATOS.....	233
FIGURA 52. SELECCIÓN DE ATRIBUTOS	234
FIGURA 53. ATRIBUTOS SELECCIONADOS	235
FIGURA 54. ATRIBUTOS DE SALIDA	236
FIGURA 55. SELECCIONAR TÉCNICA DE DIVISIÓN	238
FIGURA 56. NÚMERO DE PRUEBAS	239
FIGURA 57. CONJUNTOS DE INSTANCIAS.....	240
FIGURA 58. REPARTO DE INSTANCIAS TOTALMENTE ALEATORIO	241
FIGURA 59. REPARTO DE INSTANCIAS “DIRIGIDO”	242
FIGURA 60. AÑADIR EXPERIMENTO A PREPARAR	243
FIGURA 61. AÑADIR EXPERIMENTO DE FORMA AUTOMATIZADA	244
FIGURA 62. EJEMPLO DE EXPERIMENTOS AÑADIDOS DE FORMA AUTOMATIZADA	245
FIGURA 63. ELIMINAR VARIOS EXPERIMENTOS.....	246
FIGURA 64. EJEMPLO DE EXPERIMENTOS ELIMINADOS.....	247
FIGURA 65. PROCESO DE GENERACIÓN DE FICHEROS	248
FIGURA 66. FICHEROS GENERADOS CORRECTAMENTE.....	249
FIGURA 67. EJEMPLO DE FICHEROS GENERADOS.....	250
FIGURA 68. GUARDAR CONFIGURACIÓN	251
FIGURA 69. CARGAR CONFIGURACIÓN	252
FIGURA 70. BARRA DE HERRAMIENTAS DE NEURO SOLUTIONS	254
FIGURA 71. MODELO NEURONAL	254
FIGURA 72. FICHERO DE ENTRENAMIENTO	255
FIGURA 73. FICHERO DE DATOS EN FORMATO DE TEXTO PLANO.....	256
FIGURA 74. VARIABLES DE ENTRENAMIENTO	257
FIGURA 75. FICHERO DE TEST	258
FIGURA 76. NÚMERO DE CAPAS OCULTAS.....	259
FIGURA 77. NÚMERO DE NEURONAS OCULTAS	260
FIGURA 78. CAPA DE SALIDA DE LA RED	261
FIGURA 79. CRITERIO DE PARADA DEL ENTRENAMIENTO	262
FIGURA 80. OPCIONES DE SALIDA	263
FIGURA 81. RED NEURONAL CREADA POR NEURO SOLUTIONS	263
FIGURA 82. ARRANQUE DEL SIMULADOR.....	266
FIGURA 83. INTERFAZ DEL SIMULADOR	267
FIGURA 84. INTRODUCIR PARÁMETROS DE ENTRADA.....	268
FIGURA 85. COMIENZO EJECUCIÓN SIMULADOR.....	269
FIGURA 86. FIN EJECUCIÓN SIMULADOR	270



1 Introducción

El presente proyecto surge de la necesidad de crear un framework de comunicación para sistemas híbridos basados en redes de neuronas artificiales y sistemas expertos.

El objetivo es desarrollar un sistema híbrido inteligente genérico combinando un sistema experto con redes de neuronas artificiales. Para completar la arquitectura de dicho sistema híbrido, junto al sistema experto y las redes de neuronas artificiales, es necesario desarrollar un módulo de comunicación entre ambos. La tarea principal de este proyecto es desarrollar dicho módulo de comunicación, el cual permita la interacción entre ambos modelos de inteligencia artificial. De este modo se podrá proporcionar al usuario una arquitectura que permita la aplicación de un sistema híbrido a diversos dominios de investigación.

Además, hay que destacar que el framework a desarrollar proporcionará al usuario información acerca de los resultados obtenidos con el sistema para los experimentos que realice.

Por otro lado hay que mencionar que el uso del sistema híbrido está orientando tanto a uso en modo local como a entornos Web, de manera que se deberá compatibilizar el sistema al acceso de los usuarios a través de varios medios.

De forma complementaria al desarrollo del módulo de comunicación del framework para sistemas híbridos, el proyecto contempla también el desarrollo de una aplicación para el procesamiento previo de los datos de entrada para el sistema, permitiendo la preparación de diversos experimentos para un dominio dado.

Por último, para facilitar la interacción del usuario con los distintos módulos del sistema a desarrollar, se pretende dotar a éstos de interfaces intuitivas y fáciles de manejar, así como de manuales de usuario para su correcta utilización.

Los objetivos que se pretenden alcanzar mediante la realización de este proyecto se exponen a continuación:

- Crear un sistema capaz de cubrir todas las necesidades planteadas por el cliente.
- Seguir la metodología de desarrollo Métrica Versión 3 con el fin de generar software de calidad. En este caso, debido a las características particulares del proyecto se seguirá una adaptación de dicha metodología.



- Respetar la planificación establecida al principio del proyecto, salvo ligeras modificaciones, para poder cumplir los plazos de entrega establecidos.
- Poner en práctica los conocimientos adquiridos durante la realización de la carrera, ampliando los conocimientos acerca de los diferentes roles que intervienen en el desarrollo de un proyecto.
- Proporcionar toda la información necesaria para la utilización, mantenimiento y ampliación de la aplicación.

El presente documento recopila toda la información empleada durante el desarrollo del proyecto. Contiene todos los documentos generados en las distintas etapas del mismo, siguiendo la siguiente estructura:

- **Introducción:** Explica cómo surge el proyecto, cuáles son los objetivos a alcanzar con su realización, y la estructura de la documentación presentada.
- **Estudio de la Viabilidad del Sistema:** Realiza un estudio de la solicitud del cliente y de la situación actual, proponiendo una solución inicial al problema planteado por el cliente en base a las restricciones económicas, técnicas, legales y operativas.
- **Gestión del Proyecto:** Detalla todos los aspectos relacionados con la gestión del proyecto, tales como los recursos necesarios, una planificación detallada, y una estimación de los costes que supone su realización.
- **Análisis del Sistema:** Recoge el conjunto de requisitos que ha de cumplir el sistema a construir.
- **Diseño del Sistema:** Define el diseño del sistema de forma exhaustiva y con un nivel de detalle profundo. También realiza un estudio de toda la tecnología útil para la realización del sistema.
- **Plan de Validación y Verificación del Software:** Ofrece un catálogo de pruebas de aceptación que servirán para comprobar el correcto funcionamiento del sistema y para verificar el cumplimiento de las funcionalidades requeridas por el cliente.
- **Implementación:** Detalla la fase de codificación de los distintos módulos del sistema desarrollado.



- **Pruebas de dominio:** Presenta varios ejemplos del funcionamiento del sistema sobre varios dominios de aplicación.
- **Conclusiones:** Resume las conclusiones alcanzadas tras la realización del presente proyecto.
- **Anexos:** Incluye un documento con el formato recomendado para los ficheros de datos de entrada, así como los manuales de usuario para los módulos del sistema desarrollados.



2 Definiciones y acrónimos

2.1 Definiciones

- **Backup:** Copia de seguridad.
- **Clasificación:** Predicción de la clase a la que pertenece un determinado dato a partir de unos datos de entrada.
- **Componente:** Elemento software de un sistema informático que internamente se estructura en clases.
- **Entorno de desarrollo:** Aplicación compuesta por un conjunto de herramientas para la programación de aplicaciones.
- **Entrenamiento:** Proceso al cual se somete a una red de neuronas artificial para que adquiriera conocimiento a partir de unos datos.
- **Excel:** Aplicación de tipo hoja de cálculo incluida en la suite ofimática Microsoft Office.
- **Experimento:** Conjunto de datos preparados para el procesamiento de una red de neuronas artificial.
- **Extensión (de fichero):** Identificador de 3 caracteres situado a continuación del nombre de un fichero que permite definir el tipo del propio fichero
- **Framework:** Arquitectura de software que provee de una estructura y una metodología de funcionamiento.
- **Hoja de datos:** Elemento de un documento de la aplicación Microsoft Excel que contiene la información estructurada en forma de hoja de cálculo.
- **Instancia:** Dato acerca de un dominio compuesto por una serie de atributos con sus correspondientes valores.
- **Lenguaje de programación:** Lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.
- **Librería:** Conjunto de clases, funciones y atributos que proporcionan servicio a otra aplicación.



- **Localización:** Ruta de acceso a un fichero o directorio del ordenador.
- **Navegador:** Aplicación para visualizar el contenido de las páginas Web.
- **Poner en producción:** Proceso por el cual la red de neuronas artificiales ya entrenada y validada se encuentra en disposición de realizar predicciones a partir de unos datos de entrada.
- **Prototipo:** Elemento tomado como modelo sobre el que trabajar posteriormente.
- **Red de Neuronas Artificial:** Simulación informática que trata de imitar el comportamiento de las redes de neuronas biológicas para adquirir conocimiento.
- **Regresión:** Predicción del valor numérico de una variable.
- **Sistema Experto:** Sistema informático que emula el comportamiento de un experto en un dominio concreto.
- **Sistema Híbrido:** Sistema informático inteligente que combina el funcionamiento de varias técnicas de inteligencia artificial.
- **Stakeholders:** Aquellas personas que afectan o son afectados por el desarrollo del proyecto.
- **Test:** Ver *Validación*.
- **Texto plano:** Se refiere a ficheros compuestos únicamente de texto sin formato alguno.
- **Train:** Ver *Entrenamiento*.
- **Validación:** Proceso al que se somete a una red de neuronas para comprobar si el entrenamiento ha sido correcto y la red ha adquirido el conocimiento esperado.
- **Validación cruzada:** Mecanismo utilizado durante el proceso de entrenamiento de una red de neuronas para evitar el sobreaprendizaje de la misma.



2.2 Acrónimos

- **CU:** Caso de uso
- **RS:** Requisito Software
- **RU-C:** Requisito de Usuario de Capacidad
- **RU-R:** Requisito de Usuario de Restricción
- **XML:** Lenguaje de Marcas Extensible (eXtensible Markup Language).
- **XSL:** Lenguaje de hojas de estilo extensible (eXtensible Stylesheet Language).



3 Estudio de Viabilidad del Sistema

3.1 Introducción

3.1.1 Propósito del plan

El objetivo del Estudio de Viabilidad del Sistema es realizar un análisis detallado de las necesidades del cliente, con el fin de proponer una solución en un plazo establecido. Este estudio debe tener en cuenta restricciones de diversa índole: económicas, técnicas, legales y operativas.

Basándose en las necesidades del cliente, se detallará el alcance del sistema, es decir, se estudiará el alcance de la necesidad planteada por el cliente identificando los primeros requisitos, estructuras implicadas, suposiciones y restricciones así como los *stakeholders*, el equipo de trabajo y la planificación a seguir por éste.

Una vez definido el alcance del sistema se realizará un estudio de la situación actual de los sistemas similares existentes en la actualidad. El objetivo de esta tarea consiste en identificar los recursos de información existentes, posibles problemas y mejoras.

A continuación se realizará una definición de los requisitos que debe cumplir el sistema, de manera que se obtengan un conjunto de necesidades detalladas, no ambiguas, y completas, que sirva de base para las siguientes etapas del ciclo de vida del proyecto. Dichos requisitos se describirán en un lenguaje sencillo con el propósito de que el cliente los comprenda y pueda validarlos sin problemas.

Finalmente se expondrá la solución elegida para ser desarrollada por el equipo de trabajo.



3.2 Establecimiento del alcance del sistema

El objetivo de esta actividad es realizar una primera aproximación al estudio de los requisitos, identificar las unidades organizativas afectadas y analizar las suposiciones y restricciones del sistema. También se identificarán a las personas que deben participar en el estudio de viabilidad, así como sus perfiles, especificando sus tareas y responsabilidades dentro del proceso.

3.2.1 Estudio de la solicitud

En la solicitud del cliente se especifica la necesidad de desarrollar un framework de comunicación entre sistemas expertos y redes de neuronas artificiales, de manera que estos puedan funcionar como un sistema híbrido.

Con el desarrollo de este proyecto se pretende desarrollar un módulo de comunicación que cumpla las siguientes características:

- Interaccionar con una aplicación de redes de neuronas artificiales.
- Interaccionar con un sistema experto, respetando el interfaz de comunicación desarrollado previamente al proyecto actual.
- Generar informes de evaluación, para tratamiento en local y orientado a entornos Web.
- Controlar errores en la introducción de parámetros de entrada por parte del sistema experto.

Asimismo, de forma complementaria a dicho módulo de comunicación se pretende desarrollar también una aplicación para el tratamiento de ficheros de entrada, preparando todos los ficheros de datos para los experimentos que se deseen realizar con el sistema.

Por último, sobre todo de cara al entorno de pruebas, se considera recomendable el desarrollo de un simulador de un sistema experto para facilitar el proceso de pruebas del framework de comunicación.

Tras el estudio de la solicitud del cliente, el equipo de desarrollo del proyecto ha determinado que el cliente propone un proyecto perfectamente viable.

Las personas encargadas de llevar a cabo este proyecto son Israel González Carrasco y José Luis López Cuadrado, como directores del mismo, y David Martín Muñoz, como desarrollador.



3.2.2 Identificación del alcance del sistema

El presente proyecto comprende tres módulos a desarrollar que formarán parte del sistema: una aplicación de tratamiento de ficheros, un simulador de un sistema experto, y el módulo principal, el módulo de comunicación.

En este apartado se pretenden identificar los requisitos de cada uno de estos módulos, identificando las principales funcionalidades que se deberán desarrollar.

En primer lugar, la aplicación de tratamiento de ficheros de entrada debe proporcionar al usuario la funcionalidad principal de generar los ficheros necesarios para la posterior experimentación con el sistema híbrido desarrollado. Estos ficheros se deben generar atendiendo a una serie de parámetros a decidir por el usuario, tales como:

- Técnica de división de instancias.
- Tipo de reparto entre conjuntos de entrenamiento, validación y validación cruzada.
- Selección de atributos a importar del fichero de datos original.
- Número de pruebas del experimento.

Merece la pena destacar que la aplicación debe ser capaz de tratar los ficheros de datos originales en formato de Microsoft Excel, mientras que los generados por la propia aplicación deben ser en formato de texto plano.

Por otro lado, el simulador de un sistema experto pretende facilitar la labor de realización de pruebas al equipo de trabajo, reduciendo el tiempo de proceso al no tener que utilizar un sistema experto real. Para lograr este objetivo, el simulador desarrollado debe utilizar exactamente el mismo interfaz de interacción con el módulo de comunicación.

Por último, el módulo de comunicación es la parte central de este proyecto, ya que se encargará de permitir la comunicación entre un sistema experto y las redes de neuronas artificiales. Esta comunicación entre las dos técnicas de inteligencia artificial dará lugar a un sistema híbrido que no será dependiente del dominio, es decir, permitirá experimentar con datos de cualquier dominio gracias a este framework de comunicación.

Además, a parte de la labor de comunicación entre sistema experto y redes de neuronas artificiales, el módulo a desarrollar debe proporcionar la funcionalidad de generar informes de evaluación que permitan al usuario analizar en profundidad los resultados obtenidos en sus experimentos. Dichos informes deberán ser realizados tanto en formato de Microsoft Excel, como en XML, facilitando una futura orientación del sistema a entornos Web.



3.2.3 Identificación de los interesados en el sistema (stakeholders)

En este punto se lleva a cabo una identificación de todas aquellas personas o entidades interesadas de alguna forma en el proyecto. Se describen a continuación:

- **Cliente:** Es la persona que realiza la solicitud de desarrollo del sistema, en este caso, los clientes son Israel González Carrasco y José Luis López Cuadrado, profesores de la Universidad Carlos III de Madrid.
- **Usuarios gestores:** Son aquellas personas que harán uso del sistema. Harán uso tanto del sistema híbrido como de la aplicación de tratamiento de ficheros para generar los datos de entrada al sistema.
- **Tutores y coordinadores del proyecto:** Israel González Carrasco y José Luis López Cuadrado.
- **Autor del Proyecto:** David Martín Muñoz. Es la persona encargada de la realización del proyecto, siendo además el máximo responsable del mismo y de todos los productos generados durante su desarrollo.



3.3 Estudio de la situación actual

En este apartado se analiza la situación actual de los sistemas híbridos existentes, estableciendo una valoración acerca de los mismos.

3.3.1 Valoración del estudio de la situación actual

En la actualidad, las estrategias y modelos empleados para el diseño de sistemas híbridos basados en redes de neuronas y sistemas expertos se pueden agrupar en varios grupos, que se describen a continuación:

- **Modelos 'Stand-Alone':** Este tipo de modelos consiste en utilizar componentes software independientes que no interactúan de ningún modo entre sí. Proponen resolver un mismo problema de forma independiente con ambas técnicas de inteligencia artificial, para posteriormente comparar resultados o bien ofrecer redundancia de los mismos que sirva como validación.
- **Modelos Transformacionales:** Son similares a los modelos *Stand-Alone* en cuanto a que el resultado de desarrollo es un modelo independiente que no interactúa con ningún otro. La diferencia con ese otro tipo de modelos reside en que los Modelos Transformacionales comienzan siendo un tipo de sistema, como redes de neuronas, y terminan siendo de otro tipo, como puede ser un sistema experto.
- **Modelos 'Loosely-Coupled':** Este tipo de modelos se consideran el primer método verdadero de integración entre sistemas expertos y redes de neuronas, ya que la aplicación se descompone en dos componentes separados que se comunican a través de ficheros de datos. Este tipo de modelos se suelen desarrollar con aplicaciones software comerciales de sistemas expertos y redes de neuronas artificiales. Algunos ejemplos de este tipo de sistemas se han utilizado para el procesamiento de ADN[1] , análisis de habilidades de empleados [6] , sistemas de soporte a la decisión [10] , o para diagnóstico médico [12] .
- **Modelos 'Tightly-Coupled':** A diferencia del modelo anterior, en este caso la información entre los componentes se realiza a través de estructuras residentes en memoria en lugar de emplear ficheros externos de datos. De esta forma se mejoran las capacidades de interacción entre los componentes del sistema híbrido, así como su rendimiento. Destacan entre sus aplicaciones algunas como el mapeado de señales acústicas a símbolos [4] , o el control de temperatura de un robot soldador subacuático.



Otras aplicaciones conocidas de este tipo de sistemas híbridos incluyen el sistema *LAM* utilizado en el diseño de vidrieras [3] , la monitorización de centrales nucleares [9] , el control de un tanque químico [13] , o la interpretación de imágenes a través de la fusión de sensores [9] .

3.3.2 Realización del diagnóstico actual

Una vez realizado el estudio de las técnicas y herramientas existentes en la actualidad, se puede concluir que no existe un sistema que proporcione la misma funcionalidad que se pretende alcanzar con este proyecto.

Mientras que la mayoría de sistemas y aplicaciones existentes se han desarrollado para la resolución de un problema concreto, con el presente proyecto se pretende desarrollar la arquitectura para un sistema híbrido genérico, es decir, que permita la resolución de problemas de cualquier dominio sobre el que se tengan datos suficientes.

3.4 Definición de los requisitos del sistema

Una vez realizado el estudio de la solicitud del cliente se realiza la extracción de requisitos con el fin de cumplir con las funcionalidades debe proporcionar la aplicación.

Esta extracción de requisitos se realiza también con el objetivo de orientar al cliente, permitiendo al equipo de desarrollo obtener una lista de requisitos detallada, completa y sin ambigüedades. Estos requisitos proporcionarán una visión general de la aplicación, sin ahondar en aspectos técnicos, estableciendo las principales funcionalidades y restricciones, sirviendo de base a posteriores procesos del ciclo de vida.

Previamente a la redacción de los requisitos han sido identificados los casos de uso con el fin de lograr una mejor definición.

3.4.1 Especificación de los casos de uso

A continuación se muestran los diagramas de casos de uso para cada uno de los módulos del sistema que se van a desarrollar en este proyecto:

- **Aplicación de tratamiento de ficheros:**

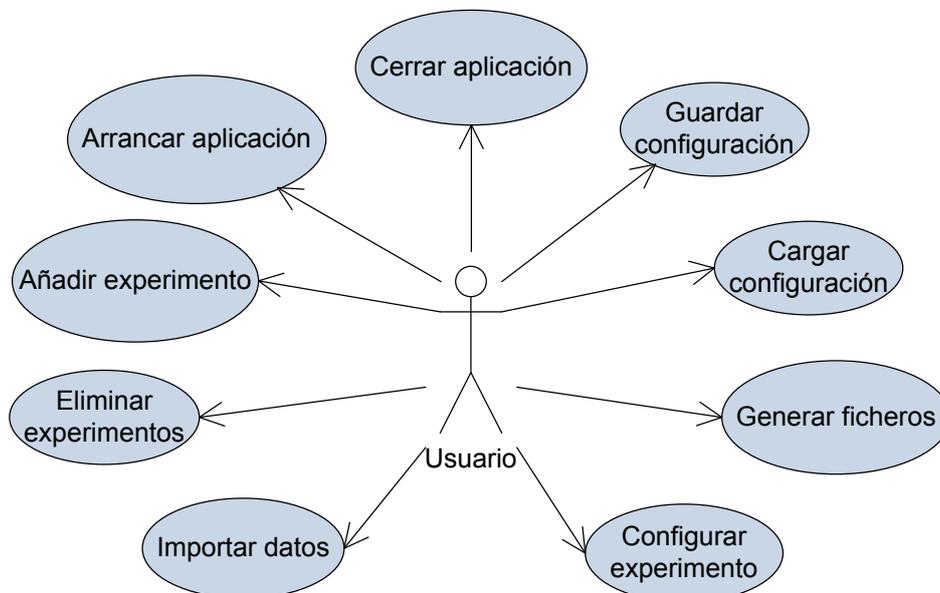


Figura 1. Diagrama de casos de uso Módulo de ficheros

- **Simulador del sistema experto:**

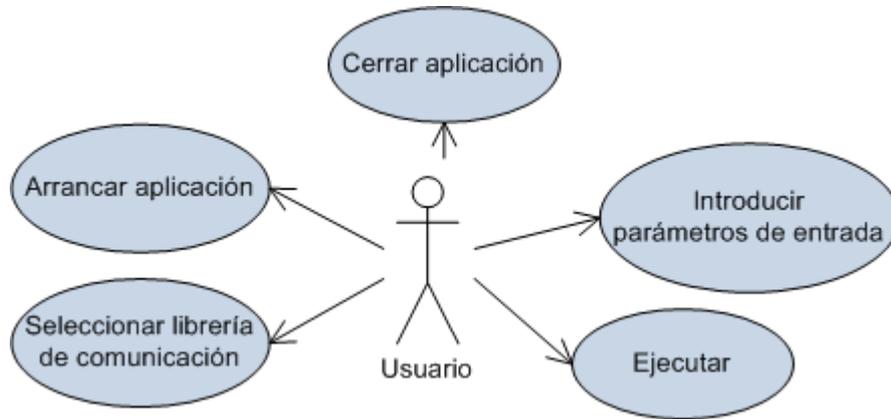


Figura 2. Diagrama de casos de uso Simulador

- **Módulo de comunicación del sistema híbrido:**

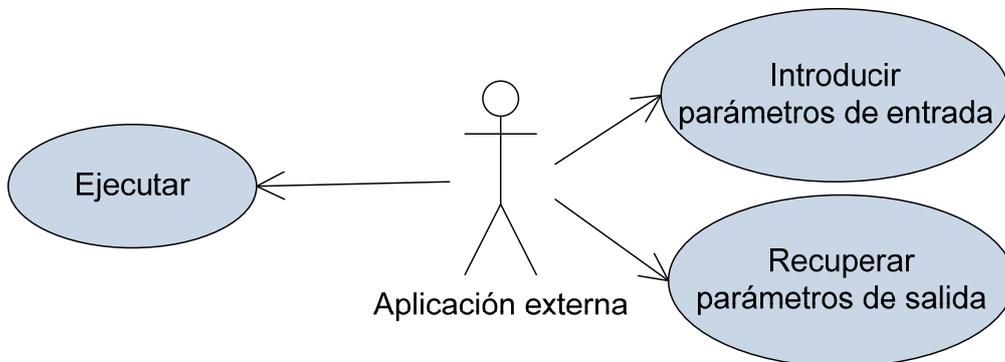


Figura 3. Diagrama de casos de uso Módulo de comunicación



3.4.1.1 Especificación detallada de los casos de uso

En este apartado se realiza la especificación de los casos de uso. Cada caso de uso estará especificado por los siguientes atributos:

- **Identificador:** Identifica al caso de uso de forma única. Debe seguir el formato: *CU-XXX*, siendo *XXX* un valor numérico único para cada caso de uso.
- **Nombre:** Breve especificación textual del caso de uso.
- **Actores:** Tipo de usuario del sistema que inicia el caso de uso.
- **Objetivo:** Finalidad del caso de uso.
- **Precondiciones:** Estado previo que se debe cumplir para poder realizar una operación.
- **Postcondiciones:** Estado en el que queda el sistema tras realizar una operación.
- **Escenario básico:** Especifica la manera en la que interactúa un actor con el sistema y cuál es la respuesta que el sistema le ofrece.
- **Escenarios alternativos:** Condiciones excepcionales que afectan al escenario y respuestas del sistema ante esas situaciones.

Aplicación de tratamiento de ficheros

Identificador	CU-001
Nombre	Arrancar aplicación.
Actores	Usuario.
Objetivo	Iniciar la aplicación de forma correcta.
Precondiciones	Ninguna.
Postcondiciones	Aplicación arrancada.
Escenario básico	<ol style="list-style-type: none">1. El usuario inicia la ejecución de la aplicación.2. La aplicación muestra la interfaz principal de usuario.
Escenarios alternativos	

Tabla 1. Caso de uso CU-001



Identificador	CU-002
Nombre	Importar datos.
Actores	Usuario.
Objetivo	Importar datos desde el fichero de entrada en formato Microsoft Excel a la aplicación para generar los ficheros de los experimentos.
Precondiciones	Aplicación arrancada.
Postcondiciones	Datos importados.
Escenario básico	<ol style="list-style-type: none">1. El usuario selecciona un fichero de entrada.2. La aplicación muestra las hojas de datos disponibles.3. El usuario selecciona una hoja de datos y hace clic sobre "Importar datos".4. La aplicación muestra los atributos disponibles.5. El usuario selecciona los atributos que desee y hace clic sobre "Importar".6. La aplicación muestra los atributos importados en la lista "Atributos del fichero de datos".
Escenarios alternativos	

Tabla 2. Caso de uso CU-002

Identificador	CU-003
Nombre	Configurar experimento.
Actores	Usuario.
Objetivo	Configurar los parámetros de un experimento antes de añadirlo a la lista de "Experimentos a preparar".
Precondiciones	Aplicación arrancada.
Postcondiciones	Experimento configurado.
Escenario básico	<ol style="list-style-type: none">1. El usuario configura las opciones del experimento a través de la interfaz de usuario.
Escenarios alternativos	

Tabla 3. Caso de uso CU-003



Identificador	CU-004
Nombre	Añadir experimento.
Actores	Usuario.
Objetivo	Añadir un experimento configurado a la lista de “Experimentos a preparar”.
Precondiciones	<ul style="list-style-type: none">▪ Aplicación arrancada.▪ Experimento configurado.
Postcondiciones	Experimento añadido a la lista de “Experimentos a preparar”.
Escenario básico	<ol style="list-style-type: none">1. El usuario hace clic sobre el botón “Añadir experimento(s)”.2. La aplicación muestra el experimento añadido en la lista “Experimentos a preparar” y limpia los campos de configuración.
Escenarios alternativos	

Tabla 4. Caso de uso CU-004

Identificador	CU-005
Nombre	Eliminar experimentos.
Actores	Usuario
Objetivo	Eliminar experimentos de la lista de “Experimentos a generar”.
Precondiciones	<ul style="list-style-type: none">▪ Aplicación arrancada.▪ Al menos un experimento añadido en la lista “Experimentos a preparar”.
Postcondiciones	Experimentos eliminados.
Escenario básico	<ol style="list-style-type: none">1. El usuario selecciona los experimentos de la lista “Experimentos a preparar” que desee eliminar.2. El usuario hace clic sobre el botón “Eliminar experimento(s)”.3. La aplicación elimina los experimentos seleccionados de la lista “Experimentos a eliminar” y muestra un mensaje informativo al usuario.
Escenarios alternativos	

Tabla 5. Caso de uso CU-005



Identificador	CU-006
Nombre	Generar ficheros.
Actores	Usuario.
Objetivo	Generar los ficheros correspondientes a los experimentos añadidos a la lista “Experimentos a preparar”.
Precondiciones	<ul style="list-style-type: none">▪ Aplicación arrancada.▪ Al menos un experimento añadido a la lista “Experimentos a preparar”.
Postcondiciones	Ficheros de experimentación generados.
Escenario básico	<ol style="list-style-type: none">1. El usuario hace clic sobre el botón “Generar Todos”.2. La aplicación genera los ficheros correspondientes a cada experimento de la lista “Experimentos a preparar”.3. La aplicación muestra un mensaje informativo al usuario.
Escenarios alternativos	

Tabla 6. Caso de uso CU-006

Identificador	CU-007
Nombre	Guardar configuración.
Actores	Usuario.
Objetivo	Guardar la configuración de la aplicación.
Precondiciones	<ul style="list-style-type: none">▪ Aplicación arrancada.▪ Al menos un experimento configurado y añadido a la lista “Experimentos a preparar”.
Postcondiciones	Configuración guardada.
Escenario básico	<ol style="list-style-type: none">1. El usuario hace clic sobre el botón “Guardar configuración”.2. La aplicación almacena la configuración existente en ese momento en un fichero en formato Excel.
Escenarios alternativos	

Tabla 7. Caso de uso CU-007



Identificador	CU-008
Nombre	Cargar configuración.
Actores	Usuario.
Objetivo	Cargar una configuración de la aplicación almacenada previamente.
Precondiciones	<ul style="list-style-type: none">▪ Aplicación arrancada.▪ Contar con un fichero de configuración almacenado previamente.
Postcondiciones	Configuración cargada.
Escenario básico	<ol style="list-style-type: none">1. El usuario hace clic sobre el botón “Cargar configuración” y selecciona el fichero de configuración almacenado previamente.2. La aplicación restaura la configuración atendiendo a la información almacenada en el fichero.
Escenarios alternativos	

Tabla 8. Caso de uso CU-008

Identificador	CU-009
Nombre	Cerrar aplicación.
Actores	Usuario.
Objetivo	Cerrar la aplicación de forma correcta.
Precondiciones	Ninguna.
Postcondiciones	Aplicación arrancada.
Escenario básico	<ol style="list-style-type: none">1. El usuario hace clic sobre el botón “Salir”.2. La aplicación se cierra completamente.
Escenarios alternativos	

Tabla 9. Caso de uso CU-009



Simulador del sistema experto

Identificador	CU-010
Nombre	Arrancar aplicación.
Actores	Usuario.
Objetivo	Iniciar la aplicación de forma correcta.
Precondiciones	Ninguna.
Postcondiciones	Aplicación arrancada.
Escenario básico	<ol style="list-style-type: none">3. El usuario inicia la ejecución de la aplicación.4. La aplicación muestra la interfaz de usuario.
Escenarios alternativos	

Tabla 10. Caso de uso CU-010

Identificador	CU-011
Nombre	Seleccionar librería de comunicación.
Actores	Usuario.
Objetivo	Seleccionar la librería de comunicación que se desea utilizar.
Precondiciones	Aplicación arrancada.
Postcondiciones	Librería de comunicación seleccionada.
Escenario básico	<ol style="list-style-type: none">1. El usuario introduce el nombre de la librería de comunicación deseada.2. La aplicación muestra el nombre en el cuadro de texto "Librería".
Escenarios alternativos	

Tabla 11. Caso de uso CU-011



Identificador	CU-012
Nombre	Introducir parámetros de entrada.
Actores	Usuario.
Objetivo	Introducir los parámetros de entrada deseados en la aplicación.
Precondiciones	Aplicación arrancada.
Postcondiciones	Parámetros de entrada introducidos.
Escenario básico	<ol style="list-style-type: none">1. El usuario introduce los parámetros de entrada deseados en la tabla "Parámetros de entrada" de la aplicación.2. La aplicación muestra el listado de parámetros introducidos.
Escenarios alternativos	

Tabla 12. Caso de uso CU-012



Identificador	CU-013
Nombre	Ejecutar.
Actores	Usuario.
Objetivo	Realizar una ejecución de un proceso sobre el módulo de comunicación elegido.
Precondiciones	<ul style="list-style-type: none">▪ Aplicación arrancada.▪ Módulo de comunicación seleccionado.▪ Parámetros de entrada introducidos.
Postcondiciones	Llamada al módulo de comunicación realizada para la ejecución del proceso.
Escenario básico	<ol style="list-style-type: none">1. La aplicación instancia el módulo de comunicación seleccionado.2. La aplicación envía los parámetros de entrada al módulo de comunicación y llama a su método principal.3. El módulo de comunicación realiza la ejecución del proceso.4. La aplicación recupera los parámetros de salida del módulo de comunicación y los muestra.
Escenarios alternativos	

Tabla 13. Caso de uso CU-013



Módulo de comunicación del sistema híbrido

Identificador	CU-014
Nombre	Introducir parámetros de entrada.
Actores	Aplicación externa.
Objetivo	Introducir los parámetros de entrada en el módulo de comunicación para poder realizar la ejecución de un proceso determinado.
Precondiciones	Ninguna.
Postcondiciones	Parámetros de entrada introducidos.
Escenario básico	1. La aplicación externa introduce los parámetros de entrada deseados en el objeto <i>ParamIn</i> del módulo de comunicación.
Escenarios alternativos	

Tabla 14. Caso de uso CU-014

Identificador	CU-015
Nombre	Ejecutar.
Actores	Aplicación externa.
Objetivo	Ejecutar el proceso solicitado por el sistema experto a través de los parámetros de entrada.
Precondiciones	Parámetros de entrada introducidos.
Postcondiciones	Ejecución realizada y parámetros de salida generados.
Escenario básico	1. La aplicación externa llama al procedimiento principal del módulo de comunicación <i>Execute()</i> . 2. El módulo de comunicación realiza el proceso deseado interaccionando con la aplicación de redes de neuronas artificiales.
Escenarios alternativos	

Tabla 15. Caso de uso CU-015



Identificador	CU-016
Nombre	Recuperar parámetros de salida.
Actores	Aplicación externa.
Objetivo	Recuperar los parámetros de salida existentes en el módulo de comunicación.
Precondiciones	Ejecución terminada.
Postcondiciones	Parámetros de salida recuperados.
Escenario básico	<ol style="list-style-type: none">1. La aplicación externa solicita al módulo de comunicación los parámetros de salida.2. El módulo de comunicación devuelve los parámetros de salida contenidos en su objeto ParamOut.
Escenarios alternativos	

Tabla 16. Caso de uso CU-016

Identificador	CU-017
Nombre	Cerrar aplicación.
Actores	Usuario.
Objetivo	Cerrar la aplicación de forma correcta.
Precondiciones	Ninguna.
Postcondiciones	Aplicación arrancada.
Escenario básico	<ol style="list-style-type: none">1. El usuario hace clic sobre el botón "Salir".2. La aplicación se cierra completamente.
Escenarios alternativos	

Tabla 17. Caso de uso CU-017



3.4.2 Definición de los requisitos del sistema

En este apartado se realiza una extracción de requisitos del sistema con el fin de presentar las principales funcionalidades deseadas por el cliente, sirviendo de base a posteriores fases del ciclo de vida del proyecto. Los requisitos identificados proporcionarán al cliente una visión general de la aplicación, de forma completa y sin ambigüedades.

3.4.2.1 Identificación de los requisitos

En esta tarea se realiza la obtención detallada de requisitos de usuario mediante sesiones de trabajo con el cliente. La lista de requisitos obtenida no es definitiva y podrá ser modificada durante el proyecto, añadiendo requisitos no contemplados, o bien eliminando o modificando requisitos incorrectos, de forma que se terminen cubriendo todas las funcionalidades expuestas por el cliente.

La definición de requisitos recoge lo que quiere el cliente y lo que necesita, englobando los requisitos obtenidos del usuario en dos grandes categorías:

- **Requisitos de capacidad:** Representan lo que necesitan los usuarios para resolver un problema o lograr un objetivo.
- **Requisitos de restricción:** Son las restricciones impuestas por los usuarios sobre cómo se debe resolver el problema o cómo se debe alcanzar el objetivo.

Cada requisito de usuario debe incluir en su definición una serie de atributos, los cuales proporcionen toda la información necesaria para su seguimiento posterior y su clasificación. Estos atributos se describen a continuación:

- **Identificador:** Cada requisito de usuario debe estar identificado de forma única. Este identificador tendrá el siguiente formato: *RU-C-*nnn** o *RU-R-*nnn**, donde:
 - ❖ RU: Indica que se trata de un requisito de usuario.
 - ❖ C: Indica que se trata de un requisito de usuario de capacidad.
 - ❖ R: Indica que se trata de un requisito de usuario de restricción.
 - ❖ nnn: Tomará valores numéricos dentro del rango 000-999.



- **Prioridad:** Se asignará una prioridad a cada requisito con el fin de poder realizar una planificación correcta durante fases posteriores. Su clasificación puede tomar los valores: alta, media y baja.
- **Necesidad:** Los requisitos clasificados como esenciales para el usuario no pueden ser eliminados, mientras que los demás requisitos estarán siempre sujetos a modificación en el caso de que exista una causa que lo justifique. Por lo tanto, se clasificarán descendientemente de acuerdo a su necesidad de la siguiente forma: esencial, deseable y opcional.
- **Claridad:** Identifica la falta o existencia de ambigüedad de un requisito, esto es, si puede ser interpretado de varias formas dependiendo del contexto. Los requisitos serán clasificados de acuerdo a su claridad de forma descendente de la siguiente forma: alta, media y baja.
- **Fuente:** Identifica el origen del requisito, que puede estar en el usuario, una fuente externa como un documento, o el propio equipo de desarrollo del proyecto.
- **Estabilidad:** Algunos requisitos pueden no estar sujetos a cambios durante el proyecto debido a su naturaleza, mientras que otros pueden estar sujetos a determinados cambios por el desarrollo de la etapa de diseño o los requisitos software. Los requisitos poco estables deberán ser observados con mayor cuidado durante el desarrollo del proyecto al ser susceptibles a cambios. Cada requisito se clasificará en: estable o inestable.
- **Verificabilidad:** Indica si el cumplimiento de un requisito en el sistema puede ser susceptible de comprobación, esto es, si se puede verificar que el requisito se ha incorporado en el diseño y que en el sistema se puede verificar su cumplimiento. La verificabilidad de cada requisito se clasificará en: alta, media y baja.



3.4.2.2 Requisitos de capacidad

A continuación se presentan los requisitos de capacidad, los cuales representan lo que necesita el usuario para resolver un problema o lograr un objetivo.

Aplicación de tratamiento de ficheros

Identificador	RU-C-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Iniciar la aplicación de forma que el usuario no deba borrar información introducida previamente.		

Tabla 18. Requisito de usuario RU-C-001



Identificador	RU-C-002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Introducir una localización para los ficheros de entrada y de salida.		

Tabla 19. Requisito de usuario RU-C-002

Identificador	RU-C-003		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Importar datos desde un fichero de Microsoft Excel a la aplicación.		

Tabla 20. Requisito de usuario RU-C-003



Identificador	RU-C-004		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Repartir las instancias disponibles entre los conjuntos de entrenamiento, validación y validación cruzada.		

Tabla 21. Requisito de usuario RU-C-004

Identificador	RU-C-005		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Utilizar una entre las siguientes técnicas de división: <ul style="list-style-type: none">Validación cruzada k-fold (KFCV).Validación cruzada leave-one-out (LOOCV).Remuestreo bootstrap (RBP).Variante de KFCV (VKFCV).		

Tabla 22. Requisito de usuario RU-C-005



Identificador	RU-C-006		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Añadir experimentos a la lista de “Experimentos a preparar” con los parámetros de configuración seleccionados.		

Tabla 23. Requisito de usuario RU-C-006

Identificador	RU-C-007		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Añadir experimentos a la lista de “Experimentos a preparar” con los parámetros de configuración seleccionados, incrementando o decrementando de forma automática los valores deseados por el usuario.		

Tabla 24. Requisito de usuario RU-C-007



Identificador	RU-C-008		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Eliminar experimentos añadidos previamente a la lista de "Experimentos a preparar".		

Tabla 25. Requisito de usuario RU-C-008

Identificador	RU-C-009		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Gestionar la configuración de la aplicación, permitiendo al usuario guardar y cargar el estado de la aplicación en un momento dado.		

Tabla 26. Requisito de usuario RU-C-009



Identificador	RU-C-010		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Generar los archivos de entrenamiento, validación cruzada y validación para los experimentos añadidos a la lista "Experimentos a preparar".		

Tabla 27. Requisito de usuario RU-C-010

Identificador	RU-C-011		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Cerrar la aplicación mediante el botón "Salir" situado en la interfaz.		

Tabla 28. Requisito de usuario RU-C-011



Simulador del sistema experto

Identificador	RU-C-012		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Iniciar la aplicación de forma que se encuentre en disposición de comenzar una nueva ejecución.		

Tabla 29. Requisito de usuario RU-C-012

Identificador	RU-C-013		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Permitir la utilización de múltiples librerías de comunicación.		

Tabla 30. Requisito de usuario RU-C-013



Identificador	RU-C-014		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo de desarrollo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Mostrar el estado en el que se encuentra en todo momento el proceso en ejecución.		

Tabla 31. Requisito de usuario RU-C-014

Identificador	RU-C-015		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Introducir parámetros de entrada definidos por las siguientes características: <ul style="list-style-type: none">Nombre del parámetro.Valor del parámetro.		

Tabla 32. Requisito de usuario RU-C-015



Identificador	RU-C-016		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Mostrar los parámetros de salida devueltos por el módulo de comunicación, definidos por las siguientes características: <ul style="list-style-type: none">Nombre del parámetro.Valor del parámetro.		

Tabla 33. Requisito de usuario RU-C-016

Identificador	RU-C-017		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Ejecutar una petición de comunicación con una aplicación de redes de neuronas artificiales a través de un módulo de comunicación elegido.		

Tabla 34. Requisito de usuario RU-C-017



Identificador	RU-C-018		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Cerrar la aplicación mediante un botón en la interfaz de la misma.		

Tabla 35. Requisito de usuario RU-C-018

Módulo de comunicación del sistema híbrido

Identificador	RU-C-019		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Interaccionar con una aplicación de redes de neuronas artificiales para realizar las siguientes funciones: <ul style="list-style-type: none">▪ Entrenar una red.▪ Validar una red.▪ Poner una red en producción.		

Tabla 36. Requisito de usuario RU-C-019



Identificador	RU-C-020		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Interaccionar con un sistema experto o simulador del mismo a través de parámetros de entrada y parámetros de salida definidos con las siguientes características: <ul style="list-style-type: none">Nombre del parámetro.Valor del parámetro.		

Tabla 37. Requisito de usuario RU-C-020

Identificador	RU-C-021		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Generar informes de evaluación en formato Excel y XML, con la información necesaria para analizar los resultados obtenidos en la validación de las redes de neuronas artificiales.		

Tabla 38. Requisito de usuario RU-C-021



Identificador	RU-C-022		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Controlar la introducción de parámetros de entrada erróneos o valores no válidos para los parámetros, informando de dichos errores a través de los parámetros de salida.		

Tabla 39. Requisito de usuario RU-C-022

3.4.2.3 Requisitos de restricción

Son las restricciones impuestas acerca de cómo se debe resolver el problema o como se debe alcanzar el objetivo.

Aplicación de tratamiento de ficheros

Identificador	RU-R-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Generar archivos en formato de texto plano, con extensión ".txt".		

Tabla 40. Requisito de usuario RU-R-001



Identificador	RU-R-002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Disponer de una interfaz sencilla e intuitiva que facilite el uso de la aplicación.		

Tabla 41. Requisito de usuario RU-R-002

Identificador	RU-R-003		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Mostrar el progreso de la generación de archivos mediante una barra de progreso en la interfaz.		

Tabla 42. Requisito de usuario RU-R-003



Identificador	RU-R-004		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Almacenar los archivos de configuración en formato de Microsoft Excel.		

Tabla 43. Requisito de usuario RU-R-004

Identificador	RU-R-005		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Desarrollar la aplicación en Microsoft Visual Studio 2005.		

Tabla 44. Requisito de usuario RU-R-005



Simulador del sistema experto

Identificador	RU-R-006		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Disponer de una interfaz sencilla e intuitiva, compuesta por una única pantalla en la que se muestren los parámetros de entrada y de salida en forma de tabla.		

Tabla 45. Requisito de usuario RU-R-006

Identificador	RU-R-007		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Desarrollar la aplicación en Microsoft Visual Studio 2005.		

Tabla 46. Requisito de usuario RU-R-007



Módulo de comunicación del sistema híbrido

Identificador	RU-R-008		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Generar un fichero XSL que sirva para dar formato a los informes generados en lenguaje XML, para que puedan ser visualizados con los navegadores Microsoft Internet Explorer® y Mozilla Firefox®.		

Tabla 47. Requisito de usuario RU-R-008

Identificador	RU-R-009		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Utilizar la clase <i>CParametro</i> de la librería <i>AviUtiles.dll</i> desarrollada previamente para instanciar los objetos <i>ParamIn</i> y <i>ParamOut</i> que contendrán los parámetros de entrada y salida, respectivamente.		

Tabla 48. Requisito de usuario RU-R-009



Identificador	RU-R-010		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Implementar un procedimiento principal denominado <i>Execute()</i> al que pueda llamar el sistema experto o el simulador del mismo.		

Tabla 49. Requisito de usuario RU-R-010

Identificador	RU-R-011		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Desarrollar el módulo de comunicación en lenguaje Visual Basic.NET en el entorno de desarrollo Microsoft Visual Studio 2005.		

Tabla 50. Requisito de usuario RU-R-011



3.5 Estudio de alternativas de solución

Con el desarrollo de este proyecto se pretende desarrollar un framework de comunicación entre Redes de Neuronas y Sistemas Expertos para el funcionamiento como sistema híbrido multidominio. La solución ha sido impuesta por el cliente y no aplica realizar un estudio de diferentes alternativas.

Dada la necesidad de utilizar una librería desarrollada previamente a este proyecto, desarrollada en Visual Basic, se empleará como lenguaje de programación Visual Basic.NET.

En cuanto al entorno de desarrollo se ha optado por Microsoft Visual Studio 2005, dada su amplia capacidad y las funcionalidades ofrecidas para el lenguaje de programación elegido.



3.6 Valoración de las alternativas

No aplica porque la solución viene dada.

3.7 Selección de la solución

No aplica porque la solución viene dada.

4 Gestión del Proyecto

4.1 Ciclo de vida

En primer lugar se definirá este concepto para seguidamente explicar que actividades se realizan en cada una de las fases que establece el modelo elegido para finalmente justificar la elección del ciclo de vida.

Se entiende por ciclo de vida software el conjunto de fases, procesos y actividades requeridas para ofertar, desarrollar, probar, integrar, explotar y mantener un producto software indicando como funciones principales la de por un lado determinar el orden de las fases y procesos y por otro el de establecer los criterios de transición para pasar de una fase a otra. (Cuevas, 2002)

Para el presente proyecto el ciclo de vida elegido ha sido el modelo “ciclo de vida con prototipado”, que es el que más se adapta a las necesidades de desarrollo en este caso, ya que el propio producto a lo largo de su desarrollo puede considerarse una sucesión de prototipos que progresan hasta llegar al estado deseado. En cada ciclo (prototipo), las especificaciones se van resolviendo paulatinamente, completando las del ciclo anterior y ganando “madurez” hasta que se alcanza el punto en el cuál se completa y puede dejarse de entrar en un nuevo ciclo.

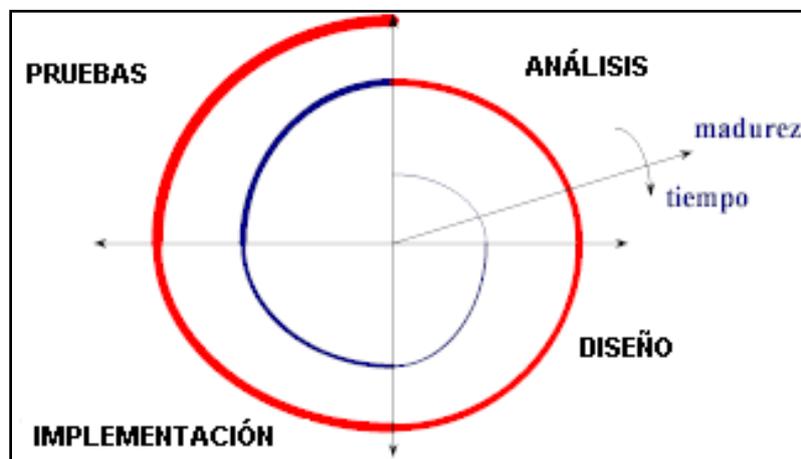


Figura 4. Ciclo de vida



En cada ciclo se completan varias fases, que se explican brevemente a continuación:

- **Análisis:** En esta fase se determinan (o amplían) los requisitos software del sistema, especificando a un alto nivel la arquitectura de la solución que se propone para dichos requisitos.
- **Diseño:** En esta segunda fase, se diseñan tanto los interfaces de usuario de la aplicación, como la arquitectura a un nivel más bajo de especificación, detallando los procesos del sistema.
- **Implementación:** En esta etapa se codifican los interfaces de usuario, se codifican los procesos, y se documenta el manual de usuario.
- **Pruebas:** En esta última fase, se definen las pruebas a realizar por el prototipo en cuestión y se llevan a cabo.

Este ciclo se ha considerado el más adecuado debido a los siguientes motivos:

- Permite evaluar distintas posibilidades de desarrollo, ya que en los prototipos se puede fácilmente añadir o eliminar funcionalidades.
- Permite al tutor o cliente realizar un seguimiento periódico del proyecto para verificar si se han comprendido bien los requisitos y si se está desarrollando la aplicación de acuerdo con éstos.
- El prototipo es un documento vivo de buen funcionamiento del producto final.
- Experimenta sobre los aspectos del sistema que representan mayor complejidad.
- El usuario reacciona mucho mejor ante el prototipo, sobre el que puede experimentar, que no sobre una especificación escrita.
- Se incrementa la calidad del producto final, ya que el prototipo permite trabajar.



4.2 Organización del proyecto

A la hora de llevar a cabo cualquier proyecto, se hace necesaria la tarea de realizar una planificación, detallando el tiempo estimado necesario para cumplir las distintas actividades y tareas que forman parte del proyecto, así como los recursos asignados a cada una de ellas. Por ello, se ha desarrollado una planificación de trabajo en función del ciclo de vida seleccionado anteriormente, ciclo de vida por prototipos.

Se ha decidido realizar tres iteraciones completas, tras cada una de las cuales el desarrollador se reúne con el tutor para corregir o matizar distintos aspectos de la aplicación.

En lo que se refiere a los recursos humanos que intervienen en el desarrollo del proyecto, a continuación se muestran las personas y roles que participan en el proyecto. Las funciones de las que se encarga cada uno de ellos son:

- **Jefe de Proyecto:** Se encarga de la gestión del proyecto, su organización, planificación y supervisión a lo largo de todo el desarrollo del mismo.
- **Analista:** Se encarga de obtener y redactar los requisitos de usuario, además de modelar los procesos y tareas a codificar.
- **Diseñador:** Su tarea es el diseño de las interfaces, la arquitectura del sistema y el plan de verificación y validación.
- **Programador:** Se encarga de la codificación del sistema, así como de llevar a cabo las pruebas necesarias sobre el mismo.

El RBS (Resource Breakdown Structure) es una representación jerárquica de los recursos, tanto humanos como materiales, necesarios para la planificación de un proyecto. Esta técnica de organización de proyectos software tiene por objeto representar la organización humana del proyecto, así como la estructura de recursos tecnológicos y materiales.

El diagrama mostrado a continuación muestra el RBS del presente proyecto:

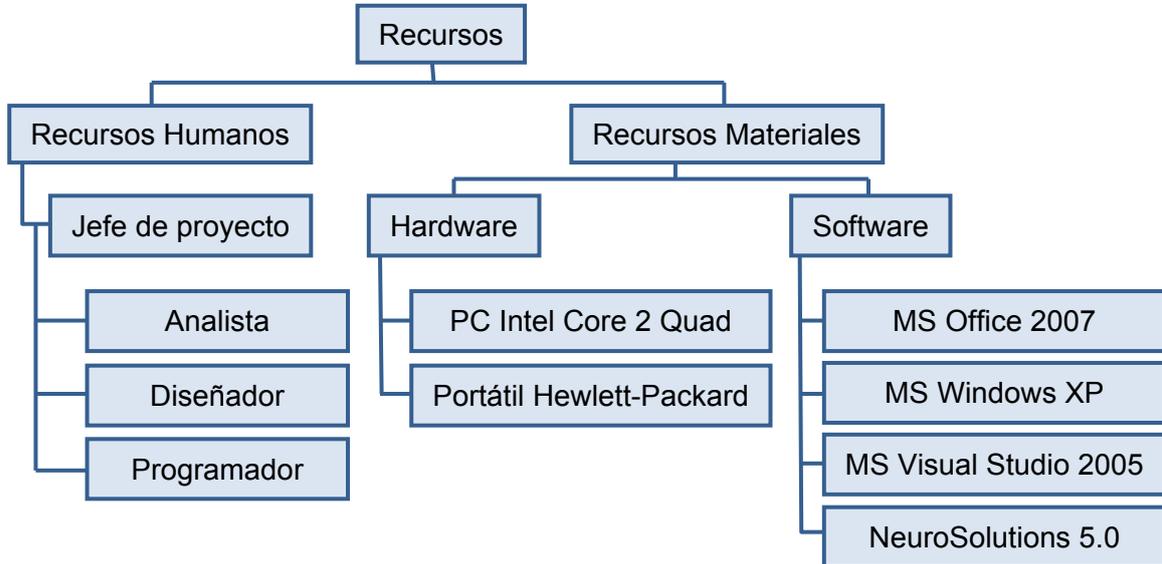


Figura 5. RBS

El WBS (World Breakdown Structure) es una herramienta simple y práctica que permite definir el alcance de un proyecto. Es la base para definir que se requiere hacer, y realizar la planificación del proyecto. Trata de identificar y definir las tareas que se van a realizar en el proyecto.

El WBS permite observar la descomposición de los distintos procesos en subprocesos, y éstos a su vez en tareas, agrupadas según su funcionalidad. La base en la que está basada el WBS es el modelo de proceso software IEEE 1074.

A continuación se muestra la división de las distintas tareas del proyecto mediante el WBS:

El primer proceso que se debe llevar a cabo es el estudio de viabilidad del proyecto, mediante el cual se analiza la solicitud del cliente, la situación actual y si el desarrollo del proyecto es viable. Toda la información se recoge en el documento de Estudio de Viabilidad.

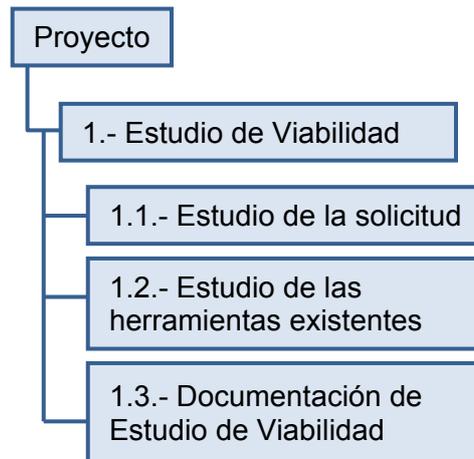


Figura 6. Proceso WBS - Estudio de Viabilidad

Una vez que el desarrollo del proyecto se ha considerado viable, el siguiente proceso a realizar es la gestión del proyecto. En este proceso se debe seleccionar el modelo de ciclo de vida más adecuado para el proyecto, y se realiza tanto la planificación a seguir durante el desarrollo del proyecto como una estimación del coste de desarrollo.

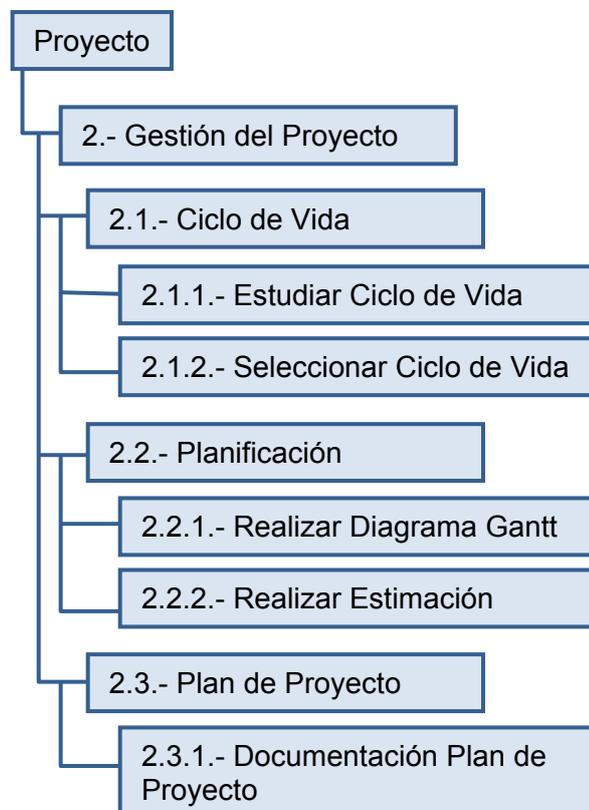


Figura 7. Proceso WBS – Gestión del Proyecto

A continuación se presentan las tres iteraciones realizadas en el modelo de ciclo de vida elegido para el desarrollo del sistema. En cada iteración se deben llevar a cabo las tareas propias del análisis, tales como la recopilación de especificaciones y requisitos, o la selección de las herramientas de desarrollo más apropiadas. Es imprescindible la realización de estas tareas antes del diseño, para asegurar que en la fase de diseño se logre un diseño del software eficiente. Además, como parte de la fase de diseño se van a desarrollar las tareas relacionadas con la arquitectura del sistema y la plataforma del proyecto. La fase de implementación consiste principalmente en realizar las tareas de codificación de las interfaces y procesos diseñados previamente. Por último, este proceso consta de un subproceso imprescindible como es el de Pruebas, en el que las tareas a realizar son la definición y ejecución de pruebas para la verificación y validación del software desarrollado.



Figura 8. Proceso WBS – Primer Prototipo



Figura 9. Proceso WBS – Segundo Prototipo



Figura 10. Proceso WBS – Tercer Prototipo



Cuando el último prototipo desarrollado es aprobado por el cliente o los tutores se pasa al siguiente proceso. En este proceso se realiza el manual de usuario para que éste pueda utilizar todas las funcionalidades disponibles en el sistema con la mayor facilidad.

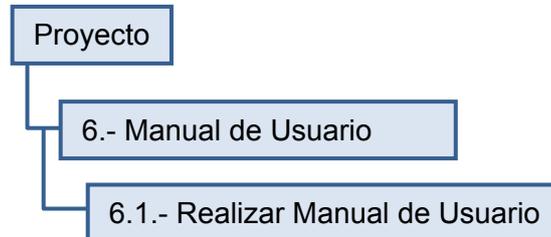


Figura 11. Proceso WBS – Manual de Usuario

4.3 Planificación

La duración total del proyecto es de 209 días laborables, comenzando su desarrollo el día 26 de Febrero de 2009 y finalizando el día 2 de Octubre de 2009, cuando el sistema se encuentre finalizado y documentado para su entrega al cliente.

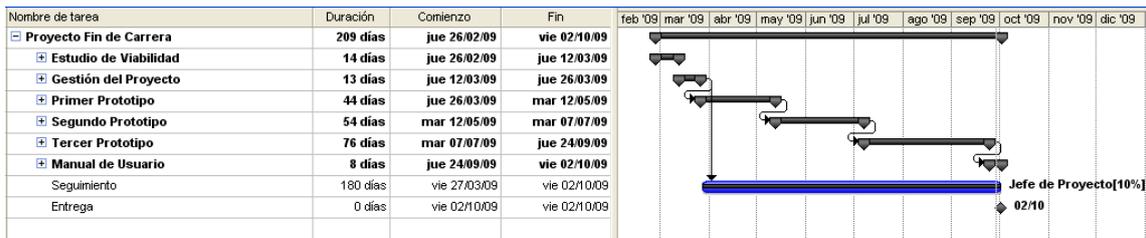


Figura 12. Diagrama de Gantt (General)

El punto de partida del proyecto es el estudio de viabilidad del mismo. Para ello, el jefe de proyecto debe realizar un estudio de la solicitud realizada por el cliente, de la situación actual y elaborar un documento que contenga dicha información. Asimismo, este documento debe indicar si la solicitud realizada es factible de ser realizada o no.

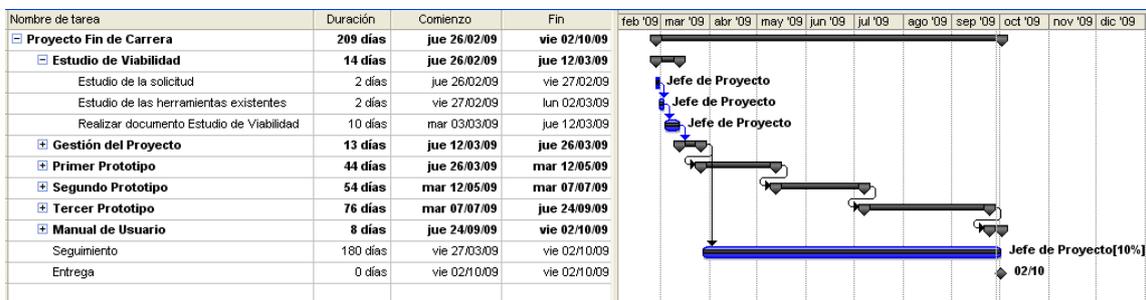


Figura 13. Diagrama de Gantt (Estudio de Viabilidad)

Una vez que el proyecto se ha considerado viable, comienza la fase de gestión del proyecto. De esta fase también se encarga el jefe de proyecto, y en ella debe realizar un estudio sobre los ciclos de vida existentes, seleccionar el más apropiado para el desarrollo del presente proyecto, documentando claramente las razones y motivos de su elección.

A continuación, se realiza el diagrama Gantt en el cual figurarán todas las tareas y actividades necesarias para realizar el desarrollo de la aplicación, así como la duración estimada de cada una de ellas y los recursos necesarios que se utilizarán en cada una de las tareas. En este diagrama las tareas ya aparecen como finalizadas, puesto que ya se ha concluido el proyecto.

Por último, el jefe de proyecto ha de realizar el correspondiente documento de Gestión de Proyecto, en el que se incluyan toda la planificación y estimaciones realizadas y comienza el seguimiento de éste, el seguimiento se llevará a cabo en paralelo a todas las actividades a realizar hasta la entrega final de la aplicación.

En la siguiente figura se muestran las tareas que conforman la Gestión de Proyecto:

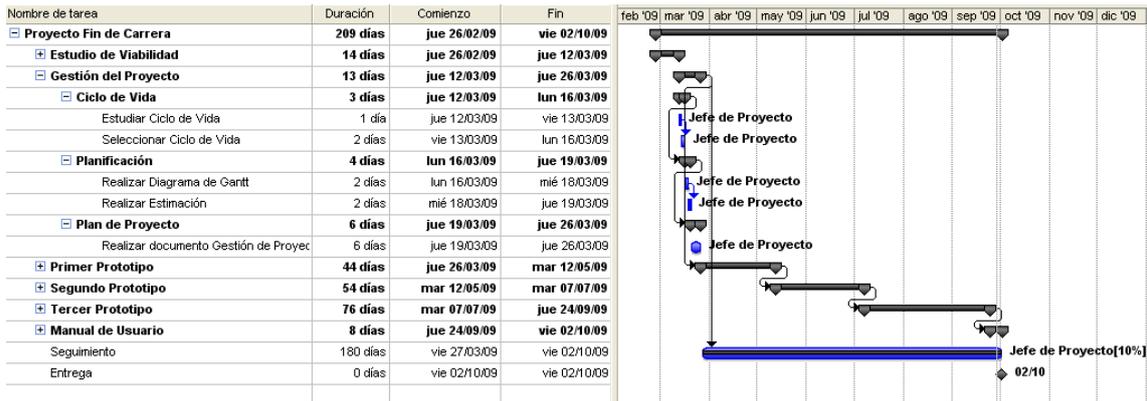


Figura 14. Diagrama de Gantt (Gestión del Proyecto)

Cuando todas las tareas de gestión del proyecto han sido realizadas comienza el desarrollo del primer prototipo del sistema. En primer lugar el analista debe recopilar los requisitos del usuario para obtener los requisitos software y seleccionar las herramientas necesarias para satisfacerlos. Es necesario que antes de crear un prototipo, el analista trabaje en conjunto con el cliente o tutor para lograr una correcta identificación de los requerimientos a satisfacer.

El diseñador lee la documentación generada por el analista y diseña la arquitectura del sistema, definiendo para ello tanto las interfaces como los procesos a implementar.

A continuación el programador debe implementar el sistema de acuerdo al diseño realizado para satisfacer los requisitos del usuario lo más fielmente posible al diseño realizado.

El primer prototipo se concluye con una batería de pruebas, para verificar su correcto funcionamiento y la evaluación por parte del cliente o tutor. Es responsabilidad del cliente trabajar con el prototipo y evaluar sus características y operación. La experiencia del sistema, bajo condiciones reales, permite obtener la familiaridad indispensable para determinar los cambios o mejoras que sean necesarios, así como las características inadecuadas.

Las pruebas serán redactadas por el diseñador y ejecutadas por el programador para comprobar el correcto funcionamiento de la aplicación.

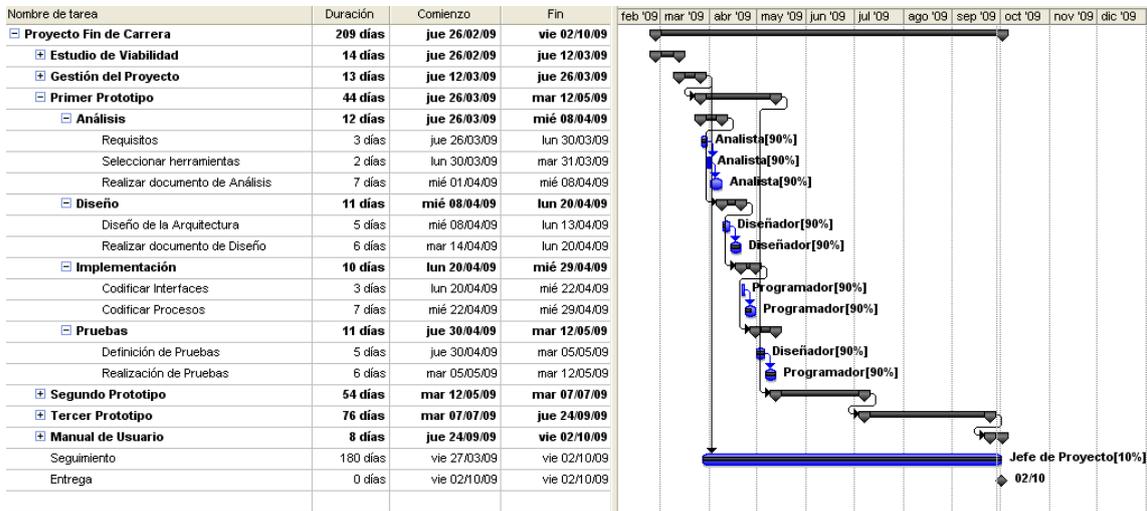


Figura 15. Diagrama de Gantt (Primer Prototipo)

El analista comienza la segunda iteración partiendo de la información aportada por el usuario en la evaluación del primer prototipo. Los cambios al prototipo son planificados con el cliente antes de llevarlos a cabo, aunque es el analista el responsable de tales modificaciones.

Como en el primer prototipo, el analista documenta los nuevos requisitos y selecciona las herramientas necesarias. A continuación el diseñador lee la documentación aportada por el analista y diseña las nuevas interfaces y funcionalidades, las cuales el programador codifica de acuerdo a las especificaciones.

Se realizan las pruebas para verificar que todo funciona correctamente y se le muestra al cliente para que evalúe el nuevo prototipo.

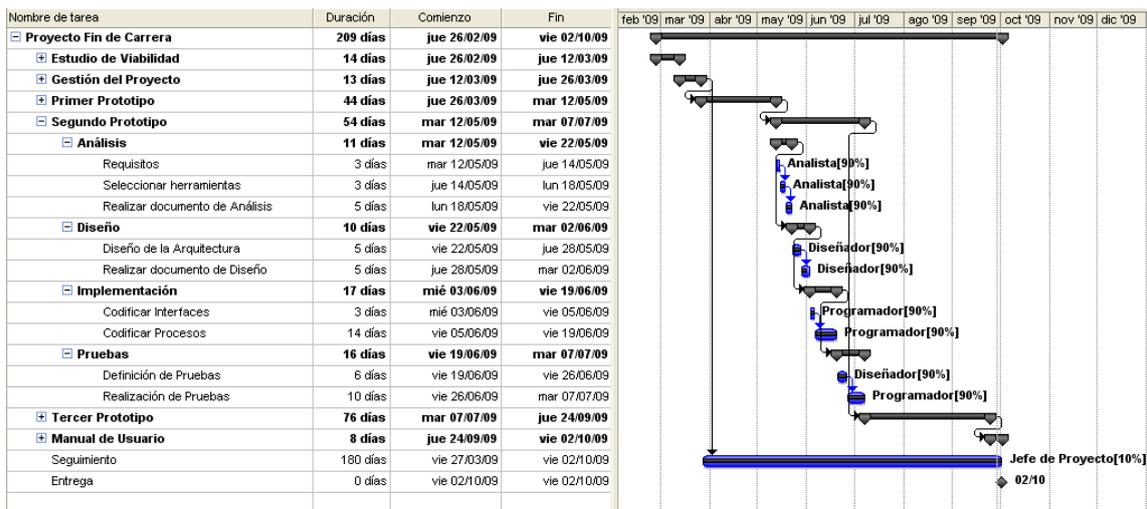


Figura 16. Diagrama de Gantt (Segundo Prototipo)

Finalmente se comienza con el tercer y último prototipo, en el que se intentarán satisfacer todos los requisitos expuestos por el cliente a lo largo del proyecto. El analista recoge todos los requisitos y comentarios del usuario y elabora la documentación definitiva, si es necesario utilizar una nueva herramienta de desarrollo la buscará y seleccionará.

El diseñador plantea las interfaces y funcionalidades que sean necesarias y el programador implementa la aplicación definitiva.

La fase de pruebas es crucial para determinar que los requerimientos han sido satisfechos y que el sistema se comporta como se esperaba. El prototipo debe pasar las pruebas de funcionalidad para validar su correcto funcionamiento y la evaluación por parte del cliente, siendo éste el que certifique la aceptación del sistema que utiliza.

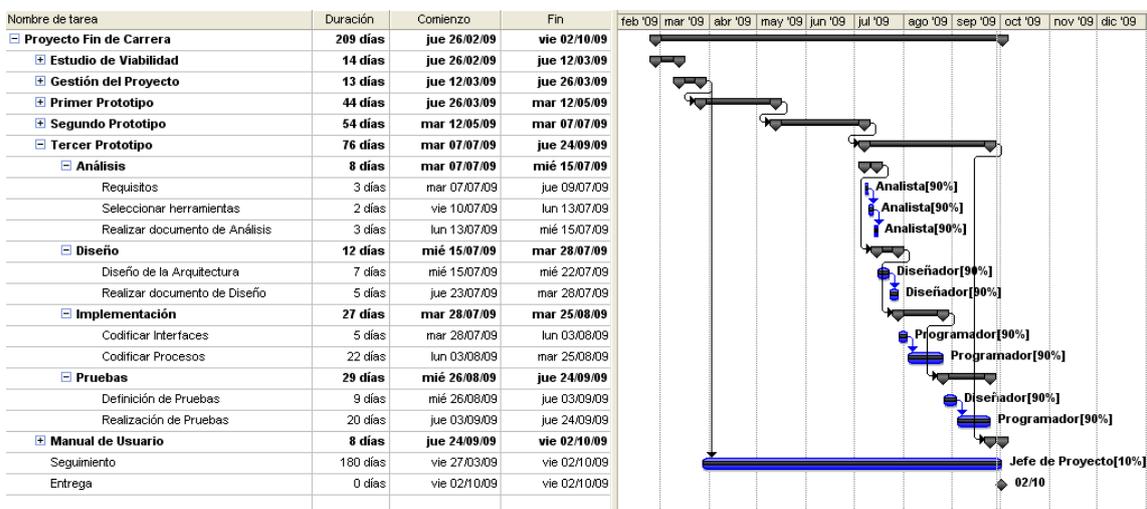


Figura 17. Diagrama de Gantt (Tercer Prototipo)

Finalmente se genera un manual de usuario indicando todos los aspectos de la aplicación, tanto sus funciones como la manera de ejecutarlas o seleccionarlás, para que el usuario final u otra persona sea capaz de utilizar la nueva aplicación sin problemas, y no tener que recurrir a los desarrolladores.

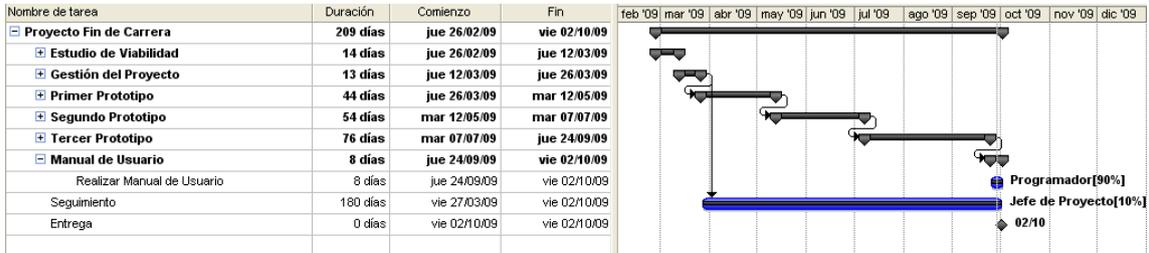


Figura 18. Diagrama de Gantt (Manual de Usuario)

El último hito del proyecto es la entrega de la documentación y el software desarrollado al cliente. Tras este paso se da por concluido el proyecto el día 2 de Octubre de 2009.



Figura 19. Diagrama de Gantt (Entrega)



4.4 Estimación de costes

Para calcular el coste estimado del proyecto se han tenido en cuenta tanto los costes materiales como los costes humanos del mismo.

En primer lugar se calcula el coste asociado a las herramientas utilizadas para el desarrollo del proyecto:

HERRAMIENTA	COSTE LICENCIA
Microsoft Office 2007	470,94 €
Microsoft Windows XP	114,84 €
Microsoft Visual Studio 2005	846,95 €
NeuroSolutions 5.0	136,77 €
TOTAL (IVA INCLUIDO)	1.569,50 €

Tabla 51. Costes de licencias

Para el cálculo de los costes asociados a los recursos humanos se ha tomado una jornada laboral de 8 horas, distribuidas en horario de 9:00 a 14:00 y 16:00 a 19:00. El proyecto ha sido desarrollado por un solo ingeniero, el cual ha adoptado los distintos roles anteriormente descritos. La siguiente tabla muestra una relación entre las funciones que se han adoptado, el tiempo invertido en cada una de ellas y su coste:

RECURSO	HORAS	PRECIO/HORA	TOTAL
Jefe de proyecto	270	45 €	12.150 €
Analista	167	35 €	5.845 €
Diseñador	286	35 €	10.010 €
Programador	529	25 €	13.225 €
TOTAL			41.230 €

Tabla 52. Costes de recursos humanos



Analizando la tabla anterior se puede observar que el coste total resultante de las horas dedicadas por el desarrollador asciende a **41.230 €**

Por lo tanto, el coste total de desarrollo del proyecto es la suma de los costes materiales más los costes humanos:

Recursos materiales	1.569,50 €
Recursos humanos	41.230 €
TOTAL	42.799,50 €

Tabla 53. Coste total del proyecto

El coste total de desarrollo del proyecto asciende a la cantidad de **42.799,50 € (cuarenta y dos mil setecientos noventa y nueve euros con cincuenta céntimos de euro)**.



5 Análisis del Sistema

5.1 Introducción

5.1.1 Objetivo del Análisis del Sistema

Al realizar el Análisis del Sistema se pretende obtener una colección completa y detallada de los requisitos del sistema, tomando como punto de partida los requisitos identificados en el Estudio de Viabilidad del Sistema.

El documento generado en esta fase será la base para el desarrollo de la fase de Diseño del Sistema, en el que se especificará el diseño completo del sistema de información.

5.1.2 Alcance

Esta fase tiene como objetivo obtener una especificación detallada del sistema que se va a diseñar. Mediante su producto, el Documento de Análisis del Sistema, se pretende captar cuales son las necesidades que tiene el cliente.

En primer lugar, se definirá cual es el alcance del sistema que se desea desarrollar, así como el entorno tecnológico asociado al proyecto. Además, se identificarán los diferentes participantes que aparecen a lo largo de la vida del proyecto así como los usuarios finales.

A continuación, se definirán cuales son los requisitos software que debe cumplir el sistema a desarrollar, tomando como punto de partida los casos de uso y los requisitos de usuario especificados en la anterior fase, el Estudio de Viabilidad del Sistema.

Por último, se definirán las interfaces de usuario que se utilizarán, dejando como última tarea las comprobaciones de calidad sobre los distintos modelos y requisitos software que se han generado durante la fase de análisis.



5.2 Definición del Sistema

5.2.1 Determinación del alcance del sistema

En este apartado se determina el alcance del sistema a desarrollar, para que permita satisfacer las necesidades planteadas por el cliente y recogidas en el documento de Estudio de la Viabilidad.

El sistema a desarrollar consiste en un framework que proporcione la vía de comunicación entre un sistema experto y una aplicación de redes de neuronas artificiales, de manera que estos puedan funcionar como un sistema híbrido.

Como pieza principal de este framework, en el desarrollo de este proyecto se pretende desarrollar un módulo de comunicación que cumpla las siguientes características:

- Interaccionar con una aplicación de redes de neuronas artificiales.
- Interaccionar con un sistema experto, respetando el interfaz de comunicación desarrollado previamente al proyecto actual.
- Generar informes de evaluación, para tratamiento en local y orientado a entornos Web.
- Controlar errores en la introducción de parámetros de entrada por parte del sistema experto.

Asimismo, de forma complementaria a dicho módulo de comunicación se pretende desarrollar también una aplicación para el tratamiento de ficheros de entrada, preparando todos los ficheros de datos para los experimentos que se deseen realizar con el sistema.

Por último, sobre todo el entorno de pruebas se considera recomendable el desarrollo de un simulador de un sistema experto para facilitar el proceso de pruebas del framework de comunicación.

5.2.2 Identificación del entorno tecnológico

Véase el apartado 5.2.3.3. Entorno operacional.



5.2.3 Especificación de estándares y normas

Los estándares y normas que ha de respetar el proceso de desarrollo del presente proyecto son los siguientes:

- **Métrica Versión 3:** Los documentos Estudio de Viabilidad del Sistema, Análisis del Sistema y Diseño del Sistema generados durante el proyecto seguirán una adaptación de la metodología de desarrollo Métrica Versión 3. La metodología será adaptada a las necesidades del proyecto, de manera que determinadas tareas no se realizarán o no se profundizará demasiado en ellas.
- **UML:** Es el lenguaje de modelado de sistemas que se utilizará para realizar el diseño del sistema.
- **IEEE 1074:** Es la norma utilizada en la Gestión del Proyecto para la definición del modelo de procesos planificado para el desarrollo del proyecto.
- **ESA Lite:** Es la metodología a seguir para el Plan de Validación y Verificación del Software. Ha sido elegida esta metodología ya que debido a las características del proyecto facilita la definición de pruebas y cubre los objetivos establecidos.

5.2.3.1 Restricciones generales

A continuación se detallan las restricciones que deberá cumplir el sistema a diseñar:

En primer lugar, el sistema funcionará bajo sistemas operativos Microsoft Windows únicamente, por lo que la máquina para la ejecución del sistema deberá cumplir esta restricción.

La interfaz, en los módulos del sistema que la incluyan, con la que interaccionará el usuario final, debe proporcionar todas las funcionalidades descritas en los requisitos software del sistema, de una forma amigable e intuitiva, evitando posibles ambigüedades que puedan ocasionar confusión al usuario.

El idioma usado en el desarrollo del sistema será el castellano. Los usuarios tienen libertad de expresarse en el idioma que consideren oportuno, pero la interfaz de la aplicación estará en castellano.



5.2.3.2 Supuestos y dependencias

Todos los usuarios participantes en la presente fase de análisis conocen el estándar empleado *Métrica Versión 3*. Por este motivo, todos los productos generados durante el proceso deben respetar lo establecido en dicho estándar.

5.2.3.3 Entorno operacional

El sistema se desarrollará mediante el entorno de desarrollo Microsoft Visual Studio 2005, utilizando como lenguaje de programación Visual Basic .NET. Los usuarios del sistema únicamente tendrán que disponer de un ordenador con sistema operativo Microsoft Windows XP.

Para el desarrollo del proyecto se han utilizado dos ordenadores con las siguientes características:

- **Ordenador Personal:**
 - ❖ Microprocesador: Intel Core 2 Quad.
 - ❖ Memoria RAM: 2048 mb.
 - ❖ Sistema Operativo: Microsoft Windows XP.

- **Ordenador portátil Hewlett-Packard:**
 - ❖ Microprocesador: Intel Core 2 Duo P8600.
 - ❖ Memoria RAM: 4096 mb.
 - ❖ Sistema Operativo: Microsoft Windows Vista.

En ambos ordenadores se contaba con la suite ofimática Microsoft Office 2007 para la gestión de documentos.

5.2.3.4 Identificación de los usuarios

En este apartado se van a detallar los usuarios que participan en el proceso de análisis del sistema, así como los usuarios que lo validarán y aceptarán finalmente.

En primer lugar, en el proceso de análisis van a participar:

- **Equipo de desarrollo:** Grupo de personas encargado de llevar a cabo el desarrollo del presente proyecto.



- **Cliente o tutores:** El cliente o en este caso tutores del proyecto, que esperan un correcto desarrollo del producto final en términos de calidad, tiempo y coste. Además, son parte importante a la hora de la obtención de los requisitos del sistema a diseñar, con el objetivo de obtener un producto que cumpla con las necesidades que le han llevado a solicitar el sistema.

En cuanto a los usuarios finales del sistema, éstos serán los que se describen a continuación:

- **Usuarios gestores:** Son aquellas personas que harán uso del sistema desarrollado. Se encargarán tanto de la configuración del sistema como de su puesta en producción.

5.2.3.5 Estudio de la seguridad requerida en el proceso de análisis

El acceso a los productos generados en la presente fase de análisis estará limitado a los usuarios participantes que se describen en el apartado 5.2.3.4. Identificación de los usuarios.



5.3 Establecimiento de Requisitos Software

5.3.1 Obtención de requisitos

A continuación se presentan los requisitos de software identificados en el análisis. El origen de estos requisitos se encuentra en los requisitos de usuario definidos en el Estudio de Viabilidad del Sistema, junto con decisiones tomadas por el equipo de desarrollo.

Los tipos de Requisitos de Software identificados en el proceso han sido los siguientes:

- **Requisitos funcionales (F):** Son aquellos requisitos que especifican el propósito del sistema, los cuales derivan directamente de los casos de uso.
- **Requisitos de interfaz (I):** Especifican hardware y/o software con los que debe interactuar el sistema.
- **Requisitos de operación (O):** Son todos aquellos requisitos que especifican como se va a resolver el problema.
- **Requisitos de documentación (D):** Requisitos que especifican criterios de realización de la documentación asociada al proyecto.
- **Requisitos de calidad (C):** Estos requisitos especifican los atributos del software que aseguran que serán adecuados para su propósito.

Cada requisito software debe incluir en su definición, además de la descripción del mismo, una serie de atributos, los cuales proporcionen toda la información necesaria para su seguimiento posterior y su clasificación. Estos atributos se describen a continuación:

- **Identificador:** Cada requisito debe estar identificado de forma única. Este identificador tendrá el siguiente formato: *RS-XX-*nnn** o *RS-XX-*nnn**, donde:
 - ❖ RS: Indica que se trata de un requisito software.
 - ❖ XX: Indicará el tipo de requisitos software al que corresponde (ver listado anterior).
 - ❖ nnn: Tomará valores numéricos dentro del rango 000-999.
- **Prioridad:** Se asignará una prioridad a cada requisito con el fin de poder realizar una planificación correcta durante fases posteriores. Su clasificación puede tomar los valores: alta, media y baja.



- **Necesidad:** Los requisitos clasificados como esenciales no pueden ser eliminados, mientras que los demás requisitos estarán siempre sujetos a modificación en el caso de que exista una causa que lo justifique. Por lo tanto, se clasificarán descendientemente de acuerdo a su necesidad de la siguiente forma: esencial, deseable y opcional.
- **Claridad:** Identifica la falta o existencia de ambigüedad de un requisito, esto es, si puede ser interpretado de varias formas dependiendo del contexto. Los requisitos serán clasificados de acuerdo a su claridad de forma descendente de la siguiente forma: alta, media y baja.
- **Fuente:** Identifica el origen del requisito, que puede estar en un requisito de usuario, o en el propio equipo de desarrollo del proyecto.
- **Estabilidad:** Algunos requisitos pueden no estar sujetos a cambios durante el proyecto debido a su naturaleza, mientras que otros pueden estar sujetos a determinados cambios por el desarrollo de la etapa de diseño. Los requisitos poco estables deberán ser observados con mayor cuidado durante el desarrollo del proyecto al ser susceptibles a cambios. Cada requisito se clasificará en: estable o inestable.
- **Verificabilidad:** Indica si el cumplimiento de un requisito en el sistema puede ser susceptible de comprobación, esto es, si se puede verificar que el requisito se ha incorporado en el diseño y que en el sistema se puede verificar su cumplimiento. La verificabilidad de cada requisito se clasificará en: alta, media y baja.



5.3.1.1 Requisitos funcionales

Aplicación de tratamiento de ficheros

Identificador	RS-F-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-001
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Iniciar la aplicación de forma que el usuario no deba borrar información introducida previamente, esto es, con todos sus elementos de información en su estado inicial.		

Tabla 54. Requisito software RS-F-001

Identificador	RS-F-002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-002
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Seleccionar una ruta del fichero de entrada mediante un cuadro de diálogo de selección de archivos.		

Tabla 55. Requisito software RS-F-002



Identificador	RS-F-003		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-002
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Seleccionar una ruta de salida mediante dos posibilidades: <ul style="list-style-type: none">▪ Cuadro de diálogo de selección de carpetas.▪ Introducción manual de la ruta completa.		

Tabla 56. Requisito software RS-F-003

Identificador	RS-F-004		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-003
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Importar datos desde un fichero en formato Microsoft Excel, pudiendo seleccionar la hoja en la que se encuentran los datos, y los atributos que se desean importar.		

Tabla 57. Requisito software RS-F-004



Identificador	RS-F-005		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-004
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Mostrar las instancias disponibles en el fichero de entrada y las instancias pendientes de repartir.		

Tabla 58. Requisito software RS-F-005

Identificador	RS-F-006		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-004
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Permitir el reparto de instancias asignando porcentajes a cada una de las clases existentes para un "Atributo de reparto" seleccionado entre los "Atributos del fichero de datos". Dichos porcentajes los puede asignar el usuario de forma manual o bien asignarse de forma aleatoria.		

Tabla 59. Requisito software RS-F-006



Identificador	RS-F-007		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-004
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Repartir las instancias disponibles en tres conjuntos: entrenamiento, test y validación, siendo opcional la inclusión de este último en los ficheros generados. Se deberá tener en cuenta la existencia o no de un “Atributo de reparto”, y de los porcentajes de reparto asignados a cada clase, en caso de existir.</p>		

Tabla 60. Requisito software RS-F-007

Identificador	RS-F-008		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-004
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Permitir la opción de “Forzar reparto”, para que la aplicación alcance las cantidades indicadas para los conjuntos de entrenamiento, test y validación, a pesar de no cumplir los porcentajes de reparto de clases establecidos.</p>		

Tabla 61. Requisito software RS-F-008



Identificador	RS-F-009		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-005
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Disponer de las siguientes técnicas de división de instancias: <ul style="list-style-type: none">Validación cruzada k-fold (KFCV).Validación cruzada leave-one-out (LOOCV).Remuestreo bootstrap (RBP).Variante de KFCV (VKFCV).		

Tabla 62. Requisito software RS-F-009

Identificador	RS-F-010		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-006
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Añadir experimentos a la lista de “Experimentos a preparar” con los parámetros de configuración seleccionados. Dichos parámetros de configuración son: <ul style="list-style-type: none">Técnica de división.Número de pruebas.Atributo de reparto.Atributos de salida.Porcentajes de reparto de instancias según las clases existentes.		

Tabla 63. Requisito software RS-F-010



Identificador	RS-F-011		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-007
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Añadir experimentos a la lista de “Experimentos a preparar” con los parámetros de configuración seleccionados, incrementando o decrementando de forma automática los valores deseados por el usuario. Dichos parámetros de configuración son:</p> <ul style="list-style-type: none"> ▪ Técnica de división. ▪ Número de pruebas. ▪ Atributo de reparto. ▪ Atributos de salida. ▪ Porcentajes de reparto de instancias según las clases existentes. 		

Tabla 64. Requisito software RS-F-011

Identificador	RS-F-012		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-008
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Eliminar experimentos añadidos previamente a la lista de “Experimentos a preparar”, pudiendo eliminarse múltiples experimentos de una vez.</p>		

Tabla 65. Requisito software RS-F-012



Identificador	RS-F-013		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-009
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Guardar la configuración de la aplicación en un momento dado, almacenando las siguientes características:</p> <ul style="list-style-type: none"> ▪ Ruta del fichero de entrada. ▪ Ruta de salida. ▪ Lista de “Experimentos a preparar”. ▪ Lista de “Atributos del fichero de datos”. ▪ Porcentajes de reparto de las clases, o marcador de reparto aleatorio en su defecto. ▪ Marcador de “Forzar reparto”. 		

Tabla 66. Requisito software RS-F-013

Identificador	RS-F-014		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-009
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Cargar la configuración de la aplicación almacenada previamente, recuperando las siguientes características:</p> <ul style="list-style-type: none"> ▪ Ruta del fichero de entrada. ▪ Ruta de salida. ▪ Lista de “Experimentos a preparar”. ▪ Lista de “Atributos del fichero de datos”. ▪ Porcentajes de reparto de las clases, o marcador de reparto aleatorio en su defecto. ▪ Marcador de “Forzar reparto”. 		

Tabla 67. Requisito software RS-F-014



Identificador	RS-F-015		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-010
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Generar los archivos de entrenamiento, validación y test correspondientes a los experimentos situados en la lista “Experimentos a preparar”. La denominación de estos archivos seguirá el formato: “TrainXX.txt”, “ValXX.txt”, y “TestXX.txt”, donde XX representa un valor numérico secuencial.		

Tabla 68. Requisito software RS-F-015

Identificador	RS-F-016		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-011
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Cerrar la aplicación mediante el botón “Salir” situado en la interfaz.		

Tabla 69. Requisito software RS-F-016



Simulador del sistema experto

Identificador	RS-F-017		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-012
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Iniciar la aplicación con los cuadros de texto y las tablas de parámetros de entrada y de salida vacíos.		

Tabla 70. Requisito software RS-F-017

Identificador	RS-F-018		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-013
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Permitir al usuario elegir la librería de comunicación a utilizar introduciendo su nombre en el cuadro de texto "Librería", especificando además su clase principal en el cuadro de texto "Clase a instanciar".		

Tabla 71. Requisito software RS-F-018



Identificador	RS-F-019		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-014
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Mostrar el estado en el que se encuentra en todo momento el proceso en ejecución, a través de un cuadro de texto multilínea, en el cual se muestren mensajes a medida que la aplicación realice tareas.		

Tabla 72. Requisito software RS-F-019

Identificador	RS-F-020		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-015
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Introducir parámetros de entrada definidos por las siguientes características: <ul style="list-style-type: none">▪ Nombre del parámetro.▪ Valor del parámetro.		

Tabla 73. Requisito software RS-F-020



Identificador	RS-F-021		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-016
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Mostrar los parámetros de salida devueltos por el módulo de comunicación, definidos por las siguientes características:</p> <ul style="list-style-type: none"> Nombre del parámetro. Valor del parámetro. 		

Tabla 74. Requisito software RS-F-021

Identificador	RS-F-022		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-017
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Ejecutar una petición de comunicación con una aplicación de redes de neuronas artificiales a través del módulo de comunicación elegido mediante el cuadro de texto "Librería".</p>		

Tabla 75. Requisito software RS-F-022



Identificador	RS-F-023		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-018
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Cerrar la aplicación mediante el botón “Salir” situado en la interfaz de la misma.		

Tabla 76. Requisito software RS-F-023

Módulo de comunicación del sistema híbrido

Identificador	RS-F-024		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-019
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Interaccionar con NeuroSolutions para realizar las siguientes funciones: <ul style="list-style-type: none">▪ Entrenar una red ya creada.▪ Validar una red entrenada.▪ Poner una red en producción. En cualquier caso la predicción a realizar podrá ser tanto Clasificación como Regresión.		

Tabla 77. Requisito software RS-F-024



Identificador	RS-F-025		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-020
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Interaccionar con un sistema experto o simulador del mismo a través de parámetros de entrada y parámetros de salida definidos con las siguientes características:</p> <ul style="list-style-type: none"> ▪ Nombre del parámetro. ▪ Valor del parámetro. 		

Tabla 78. Requisito software RS-F-025

Identificador	RS-F-026		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-021
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Generar informes de evaluación en formato de Microsoft Excel, incluyendo:</p> <ul style="list-style-type: none"> ▪ <u>Clasificación</u>: <ul style="list-style-type: none"> ❖ Rangos de valores de entrenamiento. ❖ Tasa de aciertos y fallos. ❖ Influencia de las variables. ❖ Variables críticas. ▪ <u>Regresión</u>: <ul style="list-style-type: none"> ❖ Coeficientes de correlación de las variables de salida. 		

Tabla 79. Requisito software RS-F-026



Identificador	RS-F-027		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-021
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Generar informes en formato XML con hoja de estilo XSL para su correcta visualización en navegadores Web. Dichos informes deben incluir:</p> <ul style="list-style-type: none"> ▪ <u>Clasificación:</u> <ul style="list-style-type: none"> ❖ Rangos de valores de entrenamiento. ❖ Tasa de aciertos y fallos. ❖ Influencia de las variables. ❖ Variables críticas. ▪ <u>Regresión:</u> <ul style="list-style-type: none"> ❖ Coeficientes de correlación de las variables de salida. 		

Tabla 80. Requisito software RS-F-027



Identificador	RS-F-028		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-022
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Controlar la introducción de parámetros de entrada erróneos o valores no válidos para los parámetros, informando de dichos errores a través de los parámetros de salida.		

Tabla 81. Requisito software RS-F-028

5.3.1.2 Requisitos de interfaz

Aplicación de tratamiento de ficheros

Identificador	RS-I-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-001
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Generar archivos en formato de texto plano y con extensión “.txt”.		

Tabla 82. Requisito software RS-I-001



Módulo de comunicación del sistema híbrido

Identificador	RS-I-002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-008
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Asegurar la correcta visualización de los informes en formato XML en los navegadores Microsoft Internet Explorer® y Mozilla Firefox®, mediante la implementación de un fichero XSL de formato.		

Tabla 83. Requisito software RS-I-002

5.3.1.3 Requisitos de operación

Aplicación de tratamiento de ficheros

Identificador	RS-O-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-002
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Disponer de una interfaz sencilla e intuitiva que permita configurar todos los parámetros de experimentación de forma rápida para el usuario.		

Tabla 84. Requisito software RS-O-001



Identificador	RS-O-002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-003
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Mostrar el progreso de la generación de archivos mediante una barra de progreso situada en la interfaz de usuario de la aplicación.		

Tabla 85. Requisito software RS-O-002

Identificador	RS-O-003		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-004
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Almacenar los archivos de configuración de la aplicación en formato de Microsoft Excel. Este proceso debe ser transparente al usuario, es decir, el usuario simplemente pulsará un botón “Guardar configuración” y la aplicación se encargará de todo el proceso.		

Tabla 86. Requisito software RS-O-003



Identificador	RS-O-004		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-005
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Desarrollar la aplicación en lenguaje Visual Basic.NET mediante el entorno de desarrollo Microsoft Visual Studio 2005.		

Tabla 87. Requisito software RS-O-004

Simulador del sistema experto

Identificador	RS-O-005		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-006
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Disponer de una interfaz de usuario sencilla e intuitiva, compuesta por una única pantalla en la que se muestren los parámetros de entrada y de salida en forma de tabla.		

Tabla 88. Requisito software RS-O-005



Identificador	RS-O-006		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-007
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Desarrollar la aplicación del simulador en lenguaje Visual Basic.NET mediante el entorno de desarrollo Microsoft Visual Studio 2005.		

Tabla 89. Requisito software RS-O-006

Módulo de comunicación del sistema híbrido

Identificador	RS-O-007		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-009
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Utilizar la clase <i>CParametro</i> de la librería <i>AviUtiles.dll</i> desarrollada previamente, instanciando los objetos <i>ParamIn</i> y <i>ParamOut</i> , que contendrán los parámetros de entrada y salida, respectivamente.		

Tabla 90. Requisito software RS-O-007



Identificador	RS-O-008		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-010
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Implementar un procedimiento principal denominado Execute() al que pueda llamar el sistema experto o el simulador del mismo. Este procedimiento se debe encargar de utilizar los parámetros de entrada, realizar la ejecución del proceso deseado, y devolver los parámetros de salida correspondientes.</p>		

Tabla 91. Requisito software RS-O-008

Identificador	RS-O-009		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-011
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Desarrollar la aplicación del simulador en lenguaje Visual Basic.NET mediante el entorno de desarrollo Microsoft Visual Studio 2005.</p>		

Tabla 92. Requisito software RS-O-009



5.3.1.4 Requisitos de documentación

Identificador	RS-D-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo de desarrollo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Proporcionar al cliente un manual de usuario impreso, en el que se incluya una descripción detallada de las principales funcionalidades del sistema. El idioma del manual es castellano.		

Tabla 93. Requisito software RS-D-001

5.3.1.5 Requisitos de calidad

Identificador	RS-C-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo de desarrollo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Comprobar la adecuación de todos los documentos entregados a lo largo del proyecto al estándar establecido para este desarrollo software.		

Tabla 94. Requisito software RS-C-001



Identificador	RS-C-002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo de desarrollo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Realizar copias de seguridad o backups semanales. Estas copias semanales serán totales, es decir, se realizarán sobre todo el trabajo realizado en esa semana.		

Tabla 95. Requisito software RS-C-002



5.4 Identificación de subsistemas de análisis

No existen subsistemas ya que los módulos del sistema a construir se consideran como un único bloque que realiza toda la funcionalidad de dicho módulo.



5.5 Análisis de los casos de uso

En este apartado se realiza un estudio de los casos de uso identificados en el Estudio de Viabilidad del Sistema, con el fin de identificar las clases necesarias para el sistema, así como las relaciones entre ellas y su asociación con los casos de uso.

El objetivo es identificar el mínimo número de clases que serán necesarias para el sistema, pero suficientes para cubrir todos los requisitos especificados.

5.5.1 Identificación de clases asociadas a un caso de uso

En los siguientes apartados se relacionan los casos de uso de cada uno de los módulos del sistema con las clases asociadas a cada uno de ellos.

5.5.1.1 Aplicación de tratamiento de ficheros

Identificador	Clases asociadas
CU-001	<ul style="list-style-type: none">▪ InterfazPrincipal.
CU-002	<ul style="list-style-type: none">▪ InterfazPrincipal.▪ InterfazImportacion.
CU-003	<ul style="list-style-type: none">▪ InterfazPrincipal.
CU-004	<ul style="list-style-type: none">▪ InterfazPrincipal.
CU-005	<ul style="list-style-type: none">▪ InterfazPrincipal.
CU-006	<ul style="list-style-type: none">▪ InterfazPrincipal.▪ GestorExperimento.▪ DivisionKFCV.▪ DivisionLOOCV.▪ DivisionRBP.▪ DivisionVKFCV.
CU-007	<ul style="list-style-type: none">▪ InterfazPrincipal.▪ GestorConfiguracion.
CU-008	<ul style="list-style-type: none">▪ InterfazPrincipal.▪ GestorConfiguracion.

Tabla 96. Clases asociadas a casos de uso (I)

5.5.1.2 Simulador del sistema experto

Identificador	Clases asociadas
CU-009	<ul style="list-style-type: none">▪ InterfazPrincipal.
CU-010	<ul style="list-style-type: none">▪ InterfazPrincipal.
CU-011	<ul style="list-style-type: none">▪ InterfazPrincipal.
CU-012	<ul style="list-style-type: none">▪ InterfazPrincipal.▪ Procesador.

Tabla 97. Clases asociadas a casos de uso (II)

5.5.1.3 Módulo de comunicación del sistema híbrido

Identificador	Clases asociadas
CU-013	<ul style="list-style-type: none">▪ Comunicacion.
CU-014	<ul style="list-style-type: none">▪ Comunicacion.▪ Proceso.▪ Entrenamiento.▪ Validacion.▪ Produccion.▪ Evaluacion.
CU-015	<ul style="list-style-type: none">▪ Comunicacion.

Tabla 98. Clases asociadas a casos de uso (III)

5.5.2 Descripción de la interacción de objetos

A continuación se representa la interacción descrita en el apartado anterior entre los casos de uso y las clases.

Se han utilizado diagramas de secuencia de manera que se pueda observar con una mayor claridad la interacción existente entre las clases que formarán el sistema.

Cabe destacar que en este apartado se muestran los diagramas de secuencia para los casos de uso más relevantes para cada uno de los módulos que componen el sistema.

5.5.2.1 Aplicación de tratamiento de ficheros

A continuación se muestra el diagrama de secuencia para el caso de uso CU-006: Generar ficheros.

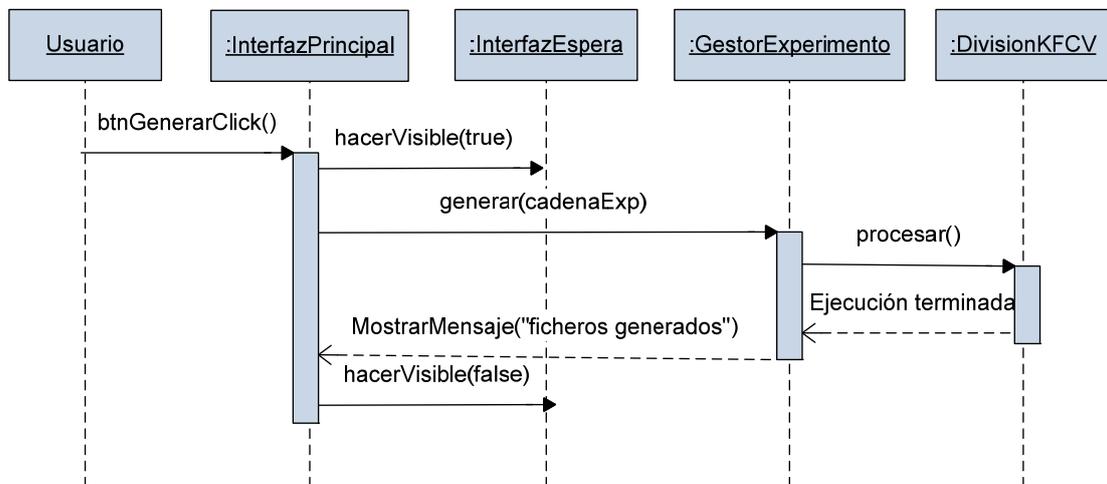


Figura 20. Diagrama de secuencia caso de uso CU-006

5.5.2.2 Simulador del sistema experto

El siguiente diagrama de secuencia representa el caso de uso CU-012: Ejecutar.

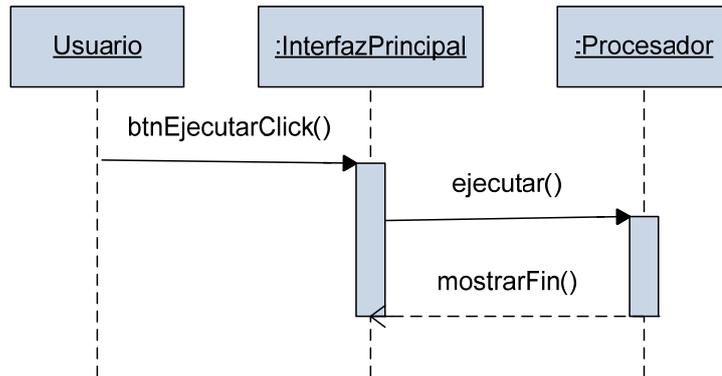


Figura 21. Diagrama de secuencia caso de uso CU-012

5.5.2.3 Módulo de comunicación del sistema híbrido

Para el módulo de comunicación del sistema híbrido se va a representar mediante el siguiente diagrama de secuencia el caso de uso CU-014: Ejecutar.

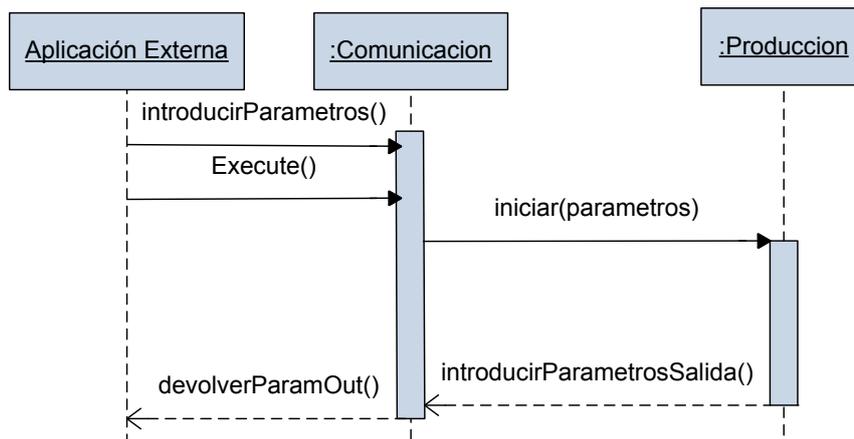


Figura 22. Diagrama de secuencia caso de uso CU-014

5.6 Análisis de clases

El objetivo de esta tarea es la descripción de cada una de las clases identificadas en el proceso de análisis. Para ello, se detallarán sus atributos y las relaciones existentes entre ellas.

Es importante destacar, que en esta fase de desarrollo del proyecto, el proceso de análisis, únicamente se realizará un estudio general de los requisitos sin hacer un especial énfasis en cuanto a la toma de decisiones relativas a la implementación. Por lo tanto, el modelo obtenido en este apartado será un simple análisis conceptual de lo que el cliente requiere, una representación a alto nivel del sistema sin ahondar en ningún tipo de detalle relacionado con la implementación. En la siguiente etapa del proyecto, la fase de diseño, se realizará un estudio más exhaustivo.

Además, con el fin de simplificar el diagrama se han incluido las funciones y procedimientos necesarios para la comprensión del sistema, obviando algunas funciones de obtención o modificación de propiedades, que aunque si serán implementadas no se muestran en este modelo.

A continuación se muestran los diagramas de clases para cada uno de los módulos del sistema:

Aplicación de tratamiento de ficheros:

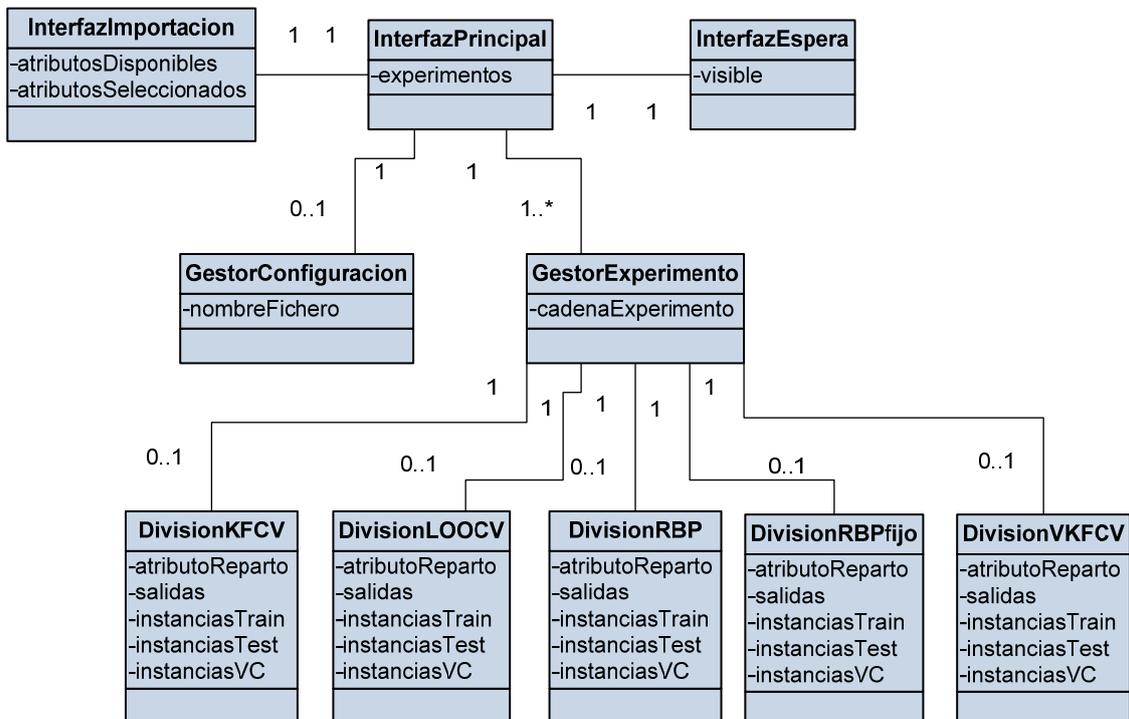


Figura 23. Diagrama conceptual de clases del Módulo de ficheros

Simulador del sistema experto:

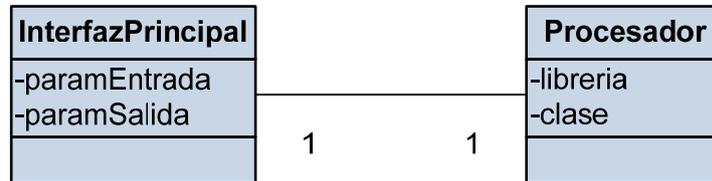


Figura 24. Diagrama conceptual de clases del Simulador

Módulo de comunicación del sistema híbrido:

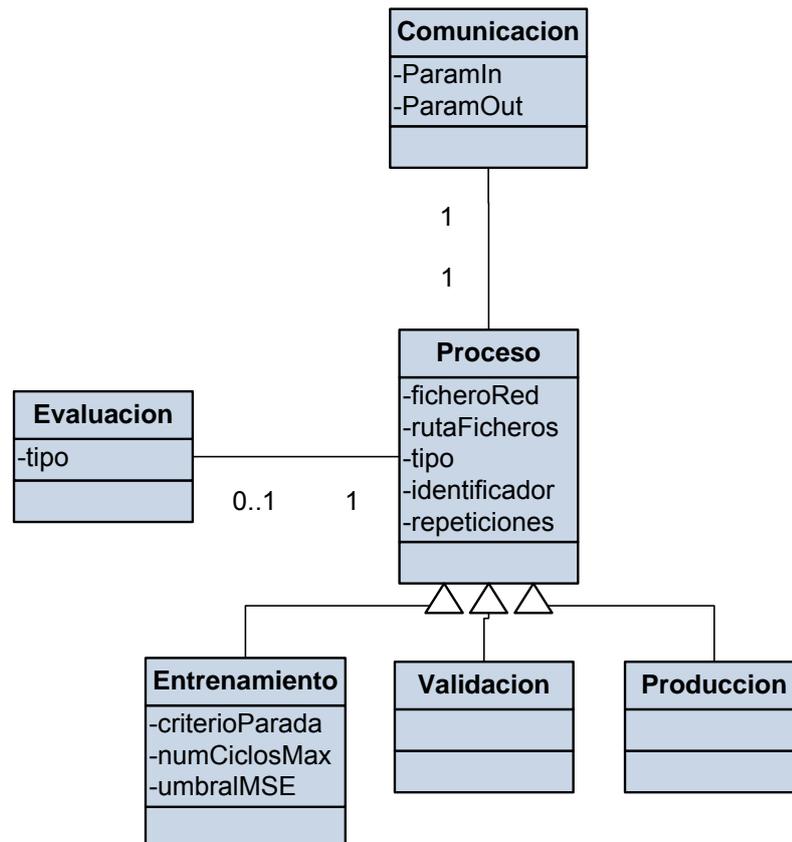


Figura 25. Diagrama conceptual de clases del Módulo de Comunicación



5.6.1 Identificación de responsabilidades y atributos

En este apartado se especifican tanto las responsabilidades asociadas a cada una de las clases mostradas en la ilustración anterior. Con ello, se pretende describir el papel que desempeñan los objetos pertenecientes a las clases en los diferentes casos de uso del sistema.

Además, se describen los atributos correspondientes a las clases, los cuales especifican propiedades de las mismas y están involucrados, a su vez, en las diferentes responsabilidades de la clase.

A continuación se describen las responsabilidades y atributos más relevantes de cada una de las clases del diagrama:

5.6.1.1 Aplicación de tratamiento de ficheros

Clase	InterfazPrincipal
Responsabilidades	Clase que gestiona el funcionamiento de la interfaz principal de la aplicación.
Atributos	<ul style="list-style-type: none">▪ experimentos: Listado de los experimentos a preparar.

Tabla 99. Clase InterfazPrincipal

Clase	InterfazImportacion
Responsabilidades	Clase que gestiona el funcionamiento de la interfaz de usuario que permite al usuario importar datos desde el fichero de origen.
Atributos	<ul style="list-style-type: none">▪ atributosDisponibles: Listado de atributos existentes en el fichero de entrada.▪ atributosSeleccionados: Listado de atributos seleccionados por el usuario.

Tabla 100. Clase InterfazImportacion



Clase	InterfazEspera
Responsabilidades	Clase que gestiona el funcionamiento de la interfaz mostrada durante la generación de archivos.
Atributos	<ul style="list-style-type: none">▪ visible: Estado de la visibilidad de la interfaz.

Tabla 101. Clase InterfazEspera

Clase	GestorConfiguracion
Responsabilidades	Clase encargada de gestionar los archivos de configuración de la aplicación.
Atributos	<ul style="list-style-type: none">▪ nombreFichero: Nombre del fichero de configuración a guardar o cargar.

Tabla 102. Clase GestorConfiguracion

Clase	GestorExperimento
Responsabilidades	Clase encargada de la preparación de un experimento añadido por el usuario.
Atributos	<ul style="list-style-type: none">▪ cadenaExperimento: Cadena textual identificadora del experimento.

Tabla 103. Clase GestorExperimento



Clase	DivisionKFCV
Responsabilidades	Clase encargada de procesar los datos utilizando la técnica de división KFCV.
Atributos	<ul style="list-style-type: none">▪ atributoReparto: Atributo en función del cual se deben repartir las instancias.▪ salidas: Atributos seleccionados por el usuario para su predicción.▪ instanciasTrain: Número de instancias para entrenamiento.▪ instanciasTest: Número de instancias para validación.▪ instanciasVC: Número de instancias para validación cruzada.

Tabla 104. Clase DivisionKFCV

Clase	DivisionLOOCV
Responsabilidades	Clase encargada de procesar los datos utilizando la técnica de división KFCV.
Atributos	<ul style="list-style-type: none">▪ atributoReparto: Atributo en función del cual se deben repartir las instancias.▪ salidas: Atributos seleccionados por el usuario para su predicción.▪ instanciasTrain: Número de instancias para entrenamiento.▪ instanciasTest: Número de instancias para validación.▪ instanciasVC: Número de instancias para validación cruzada.

Tabla 105. Clase DivisionLOOCV



Clase	DivisionRBP
Responsabilidades	Clase encargada de procesar los datos utilizando la técnica de división RBP.
Atributos	<ul style="list-style-type: none">▪ atributoReparto: Atributo en función del cual se deben repartir las instancias.▪ salidas: Atributos seleccionados por el usuario para su predicción.▪ instanciasTrain: Número de instancias para entrenamiento.▪ instanciasTest: Número de instancias para validación.▪ instanciasVC: Número de instancias para validación cruzada.

Tabla 106. Clase DivisionRBP

Clase	DivisionRBPfijo
Responsabilidades	Clase encargada de procesar los datos utilizando la técnica de división RBPfijo.
Atributos	<ul style="list-style-type: none">▪ atributoReparto: Atributo en función del cual se deben repartir las instancias.▪ salidas: Atributos seleccionados por el usuario para su predicción.▪ instanciasTrain: Número de instancias para entrenamiento.▪ instanciasTest: Número de instancias para validación.▪ instanciasVC: Número de instancias para validación cruzada.

Tabla 107. Clase DivisionRBPfijo



Clase	DivisionVKFCV
Responsabilidades	Clase encargada de procesar los datos utilizando la técnica de división VKFCV.
Atributos	<ul style="list-style-type: none">▪ atributoReparto: Atributo en función del cual se deben repartir las instancias.▪ salidas: Atributos seleccionados por el usuario para su predicción.▪ instanciasTrain: Número de instancias para entrenamiento.▪ instanciasTest: Número de instancias para validación.▪ instanciasVC: Número de instancias para validación cruzada.

Tabla 108. Clase DivisionVKFCV

5.6.1.2 Simulador del sistema experto

Clase	InterfazPrincipal
Responsabilidades	Clase que gestiona el funcionamiento de la interfaz principal de la aplicación.
Atributos	<ul style="list-style-type: none">▪ paramEntrada: Parámetros de entrada para la ejecución de un proceso.▪ paramSalida: Parámetros de salida devueltos por el módulo de comunicación tras la ejecución.

Tabla 109. Clase InterfazPrincipal



Clase	Procesador
Responsabilidades	Clase que se encarga de procesar la petición de ejecución a través del módulo de comunicación elegido.
Atributos	<ul style="list-style-type: none">▪ libreria: Nombre de la librería de comunicación.▪ clase: Clase a instanciar dentro de la librería de comunicación.

Tabla 110. Clase Procesador

5.6.1.3 Módulo de comunicación del sistema híbrido

Clase	Comunicacion
Responsabilidades	Clase encargada de interactuar con el sistema experto.
Atributos	<ul style="list-style-type: none">▪ ParamIn: Listado de parámetros de entrada.▪ ParamOut: Listado de parámetros de salida.

Tabla 111. Clase Comunicacion

Clase	Evaluacion
Responsabilidades	Clase encargada de la generación de informes de evaluación con los resultados obtenidos en la ejecución de un proceso.
Atributos	<ul style="list-style-type: none">▪ tipo: Tipo de informe a realizar, de clasificación o de regresión.

Tabla 112. Clase Evaluacion



Clase	Proceso
Responsabilidades	Clase encargada de la ejecución de un proceso determinado en la aplicación de redes de neuronas.
Atributos	<ul style="list-style-type: none">▪ ficheroRed: Fichero de la red de neuronas.▪ rutaFicheros: Ruta en la que se encuentran los ficheros necesarios para el proceso.▪ tipo: Tipo de proceso, clasificación o regresión.▪ identificador: Identificador para el proceso.▪ repeticiones: Número de repeticiones a realizar del mismo proceso.

Tabla 113. Clase Proceso

Clase	Entrenamiento
Responsabilidades	Clase encargada del proceso de entrenamiento.
Atributos	<ul style="list-style-type: none">▪ criterioParada: Criterio de parada a utilizar durante el entrenamiento de la red.▪ numCiclosMax: Número de ciclos de entrenamiento máximos.▪ umbralMSE: Valor del umbral a utilizar con el criterio de parada por error MSE.

Tabla 114. Clase Entrenamiento



Clase	Validacion
Responsabilidades	Clase encargada del proceso de validación.
Atributos	-

Tabla 115. Clase Validacion

Clase	Produccion
Responsabilidades	Clase encargada del proceso de puesta en producción.
Atributos	-

Tabla 116. Clase Produccion

5.6.2 Identificación de asociaciones

En esta tarea se estudian las relaciones de asociación entre las distintas clases identificadas en el diagrama. Estas relaciones han sido obtenidas partir de la especificación de casos de uso.

Cada una de estas asociaciones será especificada con su cardinalidad, número de instancias que pueden estar relacionadas con una instancia del otro lado de la asociación, y con una breve descripción de su funcionalidad.

A continuación se describen las asociaciones identificadas en los diagramas de clases anteriores:

5.6.2.1 Aplicación de tratamiento de ficheros

Clases asociadas	InterfazPrincipal - InterfazImportacion
Cardinalidades	<ul style="list-style-type: none">InterfazPrincipal (1)InterfazImportacion(1)
Descripción	Una interfaz principal de la aplicación utiliza una interfaz de importación para extraer los datos del fichero de origen.

Tabla 117. Asociación clases InterfazPrincipal e InterfazImportacion

Clases asociadas	InterfazPrincipal - InterfazEspera
Cardinalidades	<ul style="list-style-type: none"> ▪ InterfazPrincipal (1) ▪ InterfazEspera (1)
Descripción	Una interfaz principal de la aplicación utiliza una interfaz de espera para mostrarla al generar los archivos.

Tabla 118. Asociación clases InterfazPrincipal e InterfazEspera

Clases asociadas	InterfazPrincipal – GestorConfiguracion
Cardinalidades	<ul style="list-style-type: none"> ▪ InterfazPrincipal (1) ▪ GestorConfiguracion (0..1)
Descripción	Una interfaz principal de la aplicación puede utilizar un gestor de configuración para guardar y/o cargar ficheros de configuración.

Tabla 119. Asociación clases InterfazPrincipal y GestorConfiguracion

Clases asociadas	InterfazPrincipal – GestorExperimento
Cardinalidades	<ul style="list-style-type: none"> ▪ InterfazPrincipal (1) ▪ GestorExperimento (1..*)
Descripción	Una interfaz principal utiliza un gestor de experimento por cada experimento que tenga que preparar la aplicación al generar ficheros.

Tabla 120. Asociación clases InterfazPrincipal y GestorExperimento

Clases asociadas	GestorExperimento – DivisionKFCV
Cardinalidades	<ul style="list-style-type: none"> ▪ GestorExperimento (1) ▪ DivisionKFCV (0..1)
Descripción	Un gestor de experimento puede utilizar o no un objeto de división KFCV.

Tabla 121. Asociación clases GestorExperimento y DivisionKFCV



Clases asociadas	GestorExperimento – DivisionLOOCV
Cardinalidades	<ul style="list-style-type: none">GestorExperimento (1)DivisionLOOCV (0..1)
Descripción	Un gestor de experimento puede utilizar o no un objeto de división LOOCV.

Tabla 122. Asociación clases GestorExperimento y DivisionLOOCV

Clases asociadas	GestorExperimento – DivisionRBP
Cardinalidades	<ul style="list-style-type: none">GestorExperimento (1)DivisionRBP (0..1)
Descripción	Un gestor de experimento puede utilizar o no un objeto de división RBP.

Tabla 123. Asociación clases GestorExperimento y DivisionRBP

Clases asociadas	GestorExperimento – DivisionRBPfijo
Cardinalidades	<ul style="list-style-type: none">GestorExperimento (1)DivisionRBPfijo (0..1)
Descripción	Un gestor de experimento puede utilizar o no un objeto de división RBP de tipo fijo.

Tabla 124. Asociación clases GestorExperimento y DivisionRBPfijo

Clases asociadas	GestorExperimento – DivisionVKFCV
Cardinalidades	<ul style="list-style-type: none">GestorExperimento (1)DivisionVKFCV (0..1)
Descripción	Un gestor de experimento puede utilizar o no un objeto de división VKFCV.

Tabla 125. Asociación clases GestorExperimento y DivisionVKFCV



5.6.2.2 Simulador del sistema experto

Clases asociadas	InterfazPrincipal - Procesador
Cardinalidades	<ul style="list-style-type: none">▪ InterfazPrincipal (1)▪ Procesador (1)
Descripción	Una interfaz principal de la aplicación utiliza un procesador para ejecutar la función seleccionada.

Tabla 126. Asociación clases InterfazPrincipal y Procesador

5.6.2.3 Módulo de comunicación del sistema híbrido

Clases asociadas	Comunicacion – Proceso
Cardinalidades	<ul style="list-style-type: none">▪ Comunicación (1)▪ Proceso (1)
Descripción	Un objeto comunicación utiliza un objeto de tipo proceso para realizar el proceso elegido por el usuario.

Tabla 127. Asociación clases Comunicacion y Proceso

Clases asociadas	Proceso – Evaluacion
Cardinalidades	<ul style="list-style-type: none">▪ Proceso (1)▪ Evaluacion (0..1)
Descripción	Un objeto de tipo Proceso puede utilizar o no un objeto de tipo Evaluacion para generar un informe de los resultados de la ejecución.

Tabla 128. Asociación clases Proceso y Evaluacion

5.6.3 Identificación de generalizaciones

En este apartado se estudia otro tipo de relación entre clases, la generalización, donde una clase comparte estructura y/o comportamiento con una o más clases. El término superclase se refiere a la clase que guarda la información común, mientras que el término subclase se refiere a cada uno de los descendientes de la superclase.

Cada una de las clases involucradas en la generalización será especificada con su nivel (superclase o subclase), el nombre de su superclase, el nombre de sus subclases y una breve descripción de su funcionalidad.

A continuación se describen cada una de las clases involucradas en generalizaciones del diagrama del módulo de comunicación del sistema híbrido, ya que en el resto de módulos no aparecen generalizaciones:

Clase	Funcion
Nivel	Superclase.
Superclase	-
Subclases	<ul style="list-style-type: none"> ▪ Entrenamiento. ▪ Validacion. ▪ Produccion.
Descripción	Representa al conjunto de funciones que permite ejecutar el módulo de comunicación.

Tabla 129. Generalización clase Funcion

Clase	Entrenamiento
Nivel	Subclase.
Superclase	Funcion.
Subclases	-
Descripción	Especifica la clase Funcion para el procesamiento de un entrenamiento de la red de neuronas.

Tabla 130. Generalización clase Entrenamiento



Clase	Validacion
Nivel	Subclase.
Superclase	Funcion.
Subclases	-
Descripción	Especifica la clase Funcion para el procesamiento de una validación de la red de neuronas entrenada.

Tabla 131. Generalización clase Validacion

Clase	Produccion
Nivel	Subclase.
Superclase	Funcion.
Subclases	-
Descripción	Especifica la clase Funcion para el procesamiento de la puesta en producción de la red de neuronas entrenada y validada.

Tabla 132. Generalización clase Produccion



5.7 Definición de las interfaces de usuario

5.7.1 Especificación de principios generales de la interfaz

Uno de los aspectos más importantes de una aplicación con la que interactúan los usuarios es la interfaz, ya que es el medio de comunicación entre el usuario y el sistema. En este proyecto, tanto la aplicación de tratamiento de ficheros como el simulador del sistema experto contarán con una interfaz de usuario.

A pesar de que el perfil de usuario que utilice ambas aplicaciones será un usuario con ciertos conocimientos informáticos, es recomendable que la interfaz de usuario sea lo más clara e intuitiva posible. De esta forma se agilizará el proceso de adaptación de los usuarios al manejo de las aplicaciones.

Es importante además hacer las interfaces lo más modulables y reutilizables posible de cara a las posibles actualizaciones que puedan desarrollarse en un futuro.

Con el objetivo de facilitar el proceso de adaptación de los usuarios a las aplicaciones, además de ser interfaces sencillas, se tratará de proporcionar información precisa acerca de posibles errores cometidos en el uso de la interfaz de usuario.

5.7.2 Identificación de perfiles y diálogos

Tanto para la aplicación de tratamiento de ficheros como para el simulador de sistemas expertos existirá un único perfil de usuario, el cual podrá acceder a todas las funcionalidades especificadas por los requisitos.

5.7.3 Especificación de formatos individuales de la interfaz de la pantalla

A continuación se muestran las interfaces de usuario para las aplicaciones desarrolladas como parte del proyecto, describiendo los elementos que las componen.

5.7.3.1 Aplicación de tratamiento de ficheros

La interfaz de esta aplicación cuenta con dos pantallas. En primer lugar, la interfaz principal se muestra a continuación:

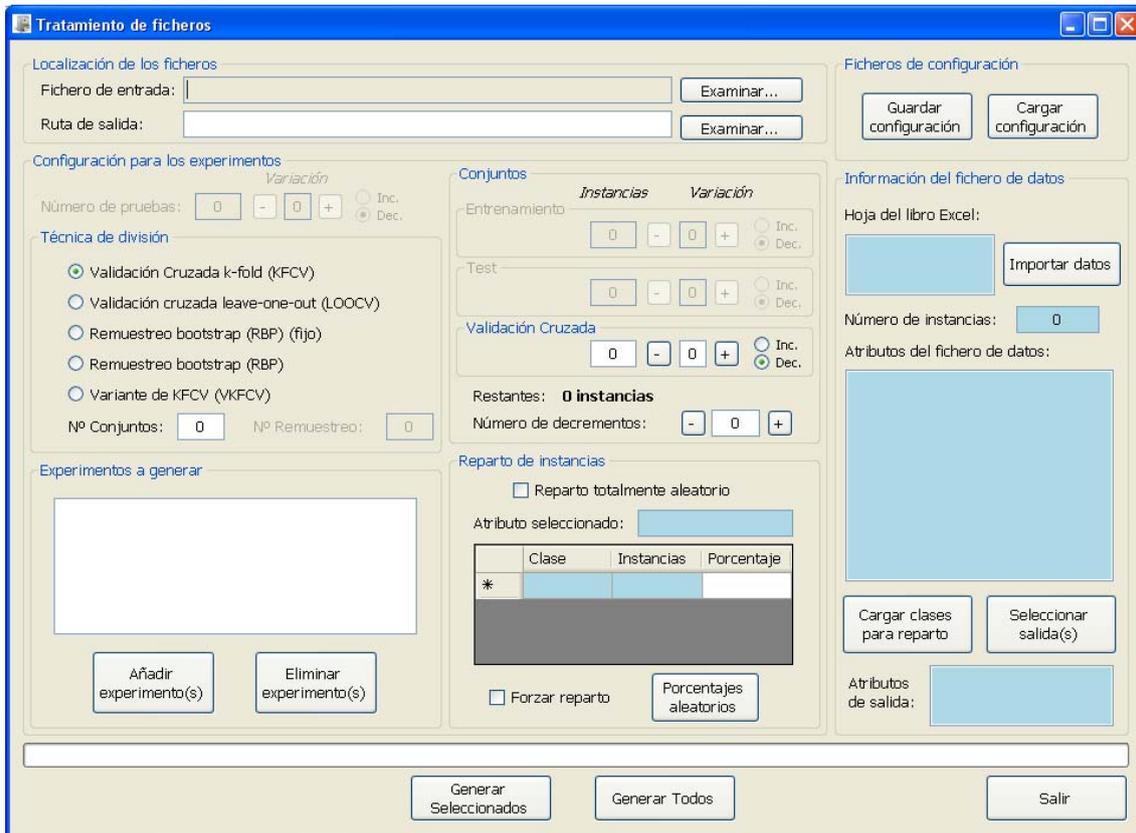


Figura 26. Interfaz principal Aplicación tratamiento de ficheros

Tal y como se puede observar en la imagen anterior, la interfaz de la aplicación se compone principalmente de los siguientes elementos:

- **Cuadro de texto “Fichero de entrada”:** Permite introducir la ruta en la que se encuentra el fichero de datos original.
- **Cuadro de texto “Ruta de salida”:** Permite especificar la ruta en la que la aplicación generará los ficheros.
- **Botón “Guardar configuración”:** Almacena la configuración de la aplicación en un fichero externo.
- **Botón “Cargar configuración”:** Carga la configuración de la aplicación desde un fichero externo almacenado previamente.



- **Lista “Experimentos a preparar”**: Listado de los experimentos para los que la aplicación debe generar ficheros de entrada.
- **Botón “Añadir experimento(s)”**: Añade los experimentos configurados a la lista “Experimentos a preparar”.
- **Botón “Eliminar experimento(s)”**: Elimina los experimentos seleccionados de la lista “Experimentos a preparar”.
- **Botón “Porcentajes aleatorios”**: Asigna de manera aleatoria porcentajes a las clases cargadas para un atributo determinado.
- **Lista “Hoja del libro Excel”**: Listado de las hojas existentes en el fichero de entrada seleccionado.
- **Botón “Importar datos”**: Abre la interfaz de importación de datos.
- **Lista “Atributos del fichero de datos”**: Muestra los atributos seleccionados por el usuario para incluir en sus ficheros de datos.
- **Botón “Cargar clases para reparto”**: Carga las clases existentes en el fichero de datos para un determinado atributo.
- **Botón “Seleccionar salida(s)”**: Añade los atributos seleccionados a la lista “Atributos de salida”.
- **Lista “Atributos de salida”**: Muestra los atributos seleccionados por el usuario sobre los que se efectuarán las predicciones.
- **Botón “Generar Seleccionados”**: Genera los ficheros para los experimentos seleccionados de la lista “Experimentos a preparar”.
- **Botón “Generar Todos”**: Genera los ficheros correspondientes para todos los experimentos de la lista “Experimentos a preparar”.
- **Botón “Salir”**: Cierra la aplicación.

La siguiente imagen muestra la interfaz de importación de datos anteriormente mencionada:

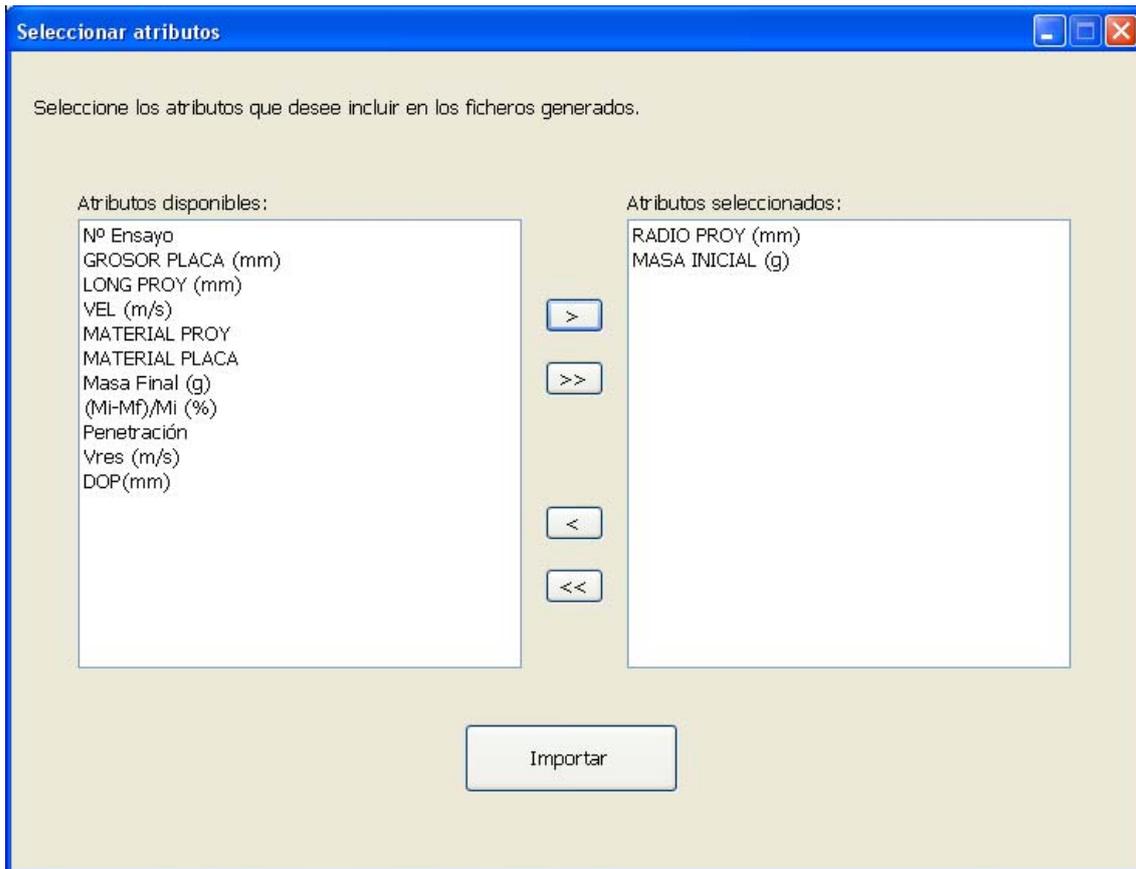


Figura 27. Interfaz de importación - Aplicación tratamiento de ficheros

En esta interfaz destacan los siguientes elementos principalmente:

- **Lista “Atributos disponibles”:** Listado de todos los atributos existentes en el fichero de datos seleccionado.
- **Lista “Atributos seleccionados”:** Listado de los atributos seleccionados por el usuario para su inclusión en los ficheros generados.
- **Botón “Importar”:** Importa la información de los atributos incluidos en la lista “Atributos seleccionados”.

5.7.3.2 Simulador de sistema experto

En este caso la interfaz del simulador también está formada por una única pantalla. La siguiente imagen muestra el contenido de la misma:

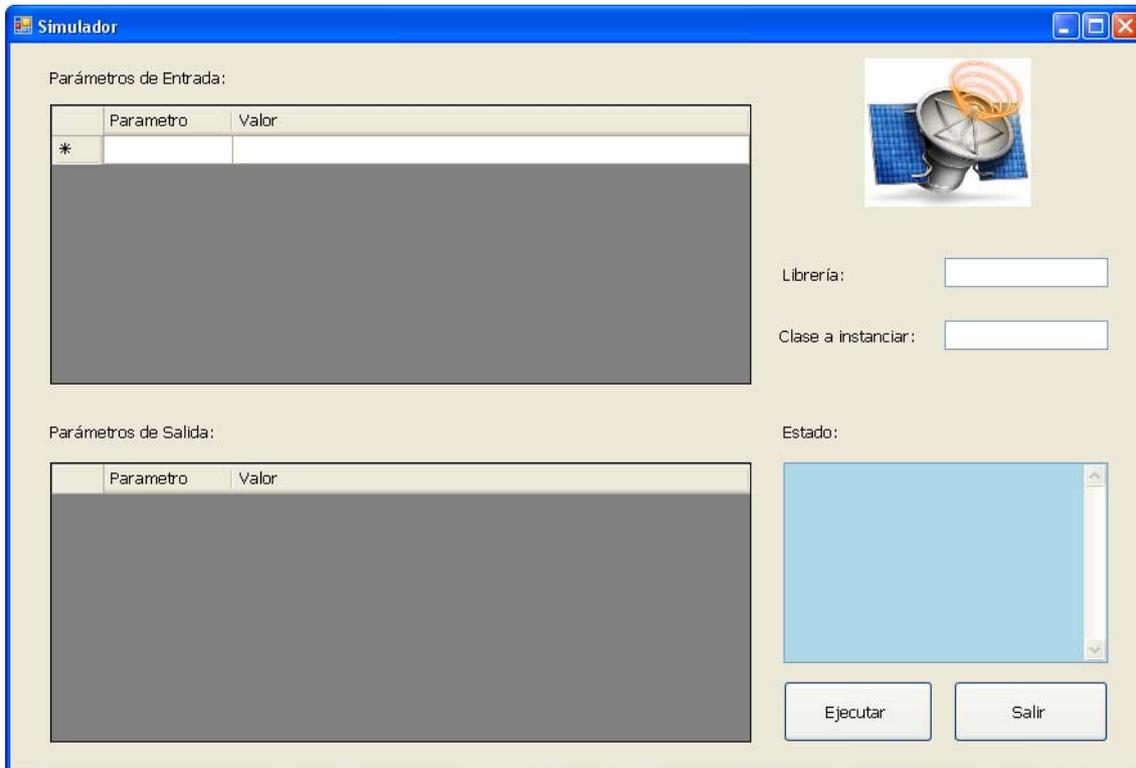


Figura 28. Interfaz principal del Simulador del Sistema Experto

Los controles que conforman la interfaz y su funcionamiento se describen a continuación:

- **Lista “Parámetros de Entrada”:** Es un listado que contiene los parámetros de entrada que se enviarán al módulo de comunicación. Cada parámetro se define mediante su nombre y su valor.
- **Lista “Parámetros de Salida”:** Este listado de parámetros contiene los parámetros de salida recibidos del módulo de comunicación al finalizar una ejecución. Al igual que los parámetros de entrada, cada parámetro se define mediante su nombre y su valor.
- **Cuadro de texto “Librería”:** Permite introducir el nombre de la librería (DLL) a instanciar.



- **Cuadro de texto “Clase a instanciar”:** Permite introducir el nombre de la clase dentro de la librería seleccionada.
- **Cuadro de texto “Estado”:** La aplicación muestra al usuario el estado en el que se encuentra el proceso de ejecución a través de este cuadro de texto.
- **Botón “Ejecutar”:** Inicia la ejecución del proceso definido mediante los parámetros de entrada.
- **Botón “Salir”:** Cierra la aplicación.



5.8 Análisis de consistencia y especificación de requisitos

En esta sección se detalla cómo se asegurará la calidad entre los distintos modelos que se han generado como productos, comprobando la consistencia entre ellos.

5.8.1 Verificación y análisis de consistencia entre modelos

Mediante esta tarea se comprobará que cada modelo generado es correcto, comprobando la coherencia de los modelos y la existencia de ambigüedad o redundancia.

Se ha comprobado que los casos de uso se pueden realizar con las clases definidas en el modelo de clases.

Por último se muestran las matrices de trazabilidad que permiten asegurar la consistencia existente entre el Estudio de Viabilidad del Sistema y el Análisis.

5.8.1.1 Matriz de Requisitos de usuario de capacidad-Casos de uso

	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007	CU-008	CU-009	CU-010	CU-011	CU-012	CU-013	CU-014	CU-015	CU-016	CU-017
RU-C-001	X																
RU-C-002		X															
RU-C-003		X															
RU-C-004			X														
RU-C-005			X														
RU-C-006				X													
RU-C-007				X													
RU-C-008					X												
RU-C-009							X	X									



RU-C-010						X												
RU-C-011								X										
RU-C-012									X									
RU-C-013										X								
RU-C-014											X							
RU-C-015											X							
RU-C-016												X						
RU-C-017												X						
RU-C-018																		X
RU-C-019																	X	
RU-C-020													X				X	
RU-C-021																	X	
RU-C-022														X				

Tabla 133. Matriz de trazabilidad Requisitos de Usuario - Casos de Uso

5.8.1.2 Matrices de Requisitos software-Casos de uso

	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007	CU-008	CU-009	CU-010	CU-011	CU-012	CU-013	CU-014	CU-015	CU-016	CU-017	
RS-F-001	X																	
RS-F-002		X																
RS-F-003			X															
RS-F-004		X																
RS-F-005		X																
RS-F-006			X															



RS-F-007		X																
RS-F-008		X																
RS-F-009		X																
RS-F-010			X															
RS-F-011			X															
RS-F-012				X														
RS-F-013					X													
RS-F-014						X												
RS-F-015					X													
RS-F-016							X											
RS-F-017								X										
RS-F-018									X									
RS-F-019										X								
RS-F-020										X								
RS-F-021											X							
RS-F-022											X							
RS-F-023																		X
RS-F-024																	X	
RS-F-025												X				X		
RS-F-026																X		
RS-F-027																X		
RS-F-028													X					

Tabla 134. Matriz de trazabilidad Requisitos Software - Casos de Uso



5.8.2 Validación de los modelos

Se ha comprobado la integridad de las matrices de trazabilidad asegurando que los modelos planteados son consistentes.



6 Diseño del Sistema

6.1 Introducción

6.1.1 Propósito

El objetivo principal de este documento es presentar el diseño del sistema realizado de forma detallada. También se estudiará la tecnología que será de utilidad para llevar a cabo esta actividad.

Se realizará una especificación detallada de los componentes en los que se dividirá el sistema, con el fin de cubrir todas las decisiones de diseño correspondientes a la fase de construcción. De esta manera, sólo quedarán posibles decisiones a tomar relacionadas con el lenguaje de programación seleccionado para la codificación del sistema.

El Documento de Diseño del Sistema es de vital importancia en el desarrollo de un proyecto software, ya que marca las pautas para las posteriores fases de construcción y de implantación final del sistema, siendo un documento básico para los programadores encargados de la implementación.



6.2 Alcance

El presente documento presenta una especificación detallada de los componentes en los que se dividirá el sistema, permitiendo cubrir todas las decisiones de diseño que se verán reflejadas en la fase construcción.

Por otro lado, se especificará el entorno tecnológico necesario para que el sistema pueda entrar en ejecución. Contendrá además la planificación de capacidades, los requisitos de administración, el control de accesos, la seguridad y la operación.

A continuación, se describen con mayor detalle las principales tareas que se presentan en este documento:

- **Definición de la arquitectura del sistema:** Se realizará una división del sistema con el objetivo de organizar y facilitar el diseño. Estos subsistemas serán partes lógicas y coherentes, que tendrán interfaces de comunicación entre ellas bien definidas. Una vez definidos estos subsistemas, habrá que acoplarlos adecuadamente dentro de la arquitectura del sistema.
- **Diseño de la arquitectura de soporte:** En esta tarea se determinarán los mecanismos genéricos de diseño que servirán posteriormente de ayuda para el diseño detallado del sistema de información. Por tanto, se deberá realizar en paralelo con el diseño detallado, ya que hay mucho en común entre ambas actividades y los cambios de una tarea afectan a la otra.
- **Diseño de casos de uso reales:** Se partirá de los casos de uso identificados en la fase de análisis del sistema y se relacionarán con las clases identificadas en el diseño, en lugar de trabajar con el modelo conceptual del documento de Análisis.
- **Diseño de clases:** El objetivo es transformar el modelo de clases del análisis en un modelo de clases de diseño. Dicho modelo es mucho más extenso que el anterior, ya que recoge información de los atributos, métodos y las relaciones que existen entre las distintas clases. Para ello, es fundamental considerar las especificaciones tecnológicas elegidas para la implementación.
- **Diseño físico de datos:** Indica el formato de los ficheros que la aplicación va a utilizar.
- **Verificación y aceptación de la arquitectura del sistema:** En este apartado se comprobará la trazabilidad entre componentes, clases y casos de uso.



- **Especificación técnica del plan de pruebas:** Se especificará en detalle el entorno de pruebas a realizar para comprobar el correcto funcionamiento del sistema.
- **Establecimiento de requisitos de implantación:** En esta actividad se completará el catálogo de requisitos con aquéllos relacionados con la implantación del sistema en el entorno. Todo esto permite preparar los medios y recursos necesarios para que los usuarios puedan manejar la aplicación sin problemas.

Es importante destacar que el documento que aquí se presenta es una adaptación del diseño planteado por *Métrica Versión 3*, ya que debido a las características del proyecto (se trata de un proyecto pequeño) algunas de las actividades son demasiado extensas para el problema tratado. Muestra de ello es que la actividad definida por métrica versión 3 DSI 8 Generación de Especificaciones de Construcción no se ha desarrollado. Además, por la misma razón, en otras actividades no se ha profundizado demasiado como es el caso de la actividad DSI 3 Diseño de casos de uso reales en el que se han definido los casos de uso más representativos de cada uno de los módulos del sistema.



6.3 Definición de la arquitectura

6.3.1 Definición de los niveles de la arquitectura

La arquitectura del sistema a desarrollar en el presente proyecto no atiende a ninguno de los modelos estandarizados, utiliza una arquitectura específica en la que interactúan los componentes entre sí.

Los componentes del sistema experto y de las redes de neuronas artificiales son componentes que se encuentran intercomunicados mediante el componente del módulo de comunicación, el cual permite sostener la arquitectura completa.

6.3.2 Especificación de estándares y normas de diseño y construcción

Entre las actividades a realizar en este proyecto se encuentra la codificación del sistema, lo cual hace indispensable la tarea de definir un estándar que sirva de guía tanto a diseñadores como a programadores a la hora de generar el código fuente. A continuación se detallan las consideraciones a tener en cuenta:

6.3.2.1 Nombres de Ficheros

Los nombres de los ficheros que conforman el código fuente del sistema deberán ser representativos de la funcionalidad que contiene cada uno.

Además, su longitud no deberá superar los 25 caracteres y la primera letra deberá estar en escrita en mayúscula. En caso de ser un nombre compuesto la primera letra de cada palabra será escrita en mayúscula y el resto en minúsculas.

Ejemplo: *NombreDeFichero*

6.3.2.2 Idioma

El idioma utilizado a lo largo de todo el desarrollo del proyecto será el castellano, por lo que los comentarios introducidos en el código fuente deben realizarse en este idioma.



6.3.2.3 Cabeceras de las clases

Cada una de las clases generadas mostrará al principio de su código un conjunto de líneas de comentario siguiendo el formato que se muestra a continuación:

```
`Nombre de la clase:  
`Descripción:  
`Autor:
```

6.3.2.4 Visibilidad de las clases

Las clases podrán ser públicas (public) o privadas (private).

6.3.2.5 Identificadores de las clases

Los identificadores de las clases estarán formados únicamente por letras, como excepción, la letra “ñ” será representada mediante los caracteres “ny”.

Además, los identificadores siempre comenzarán con una letra mayúscula. En el caso de que sea un nombre compuesto, la primera letra a partir de la segunda palabra será mayúscula también.

Ejemplo: *IdentificadorCompuesto*

6.3.2.6 Constantes

En caso de utilizar constantes, su nombre estará formado únicamente por letras mayúsculas. En el caso de que sea una palabra compuesta, las diferentes palabras se separarán con guiones bajos “_”.

Ejemplo: *CONSTANTE*

6.3.2.7 Variables

Los nombres de variables estarán formados por letras minúsculas. En el caso de ser una palabra compuesta, la primera letra de cada palabra posterior a la primera palabra será mayúscula.

Ejemplo: *nombreVariable*



6.3.2.8 Funciones y procedimientos

En cuanto a las funciones y procedimientos, es muy importante que el nombre describa con la mayor precisión posible la funcionalidad que desempeña. Para ello, los nombres utilizados serán verbos que describan la acción que hace la función o procedimiento siempre que sea posible.

Los nombres deberán ser escritos en letras minúsculas únicamente, utilizando mayúsculas en el caso de tratarse de un nombre compuesto, en cuyo caso la primera letra de cada palabra posterior a la primera será escrita en mayúsculas.

Ejemplo: *nombreFuncion*

Los posibles parámetros de entrada o salida de las funciones y procedimientos seguirán el formato de nomenclatura descrito en el apartado 6.3.2.7. Variables

6.3.3 Identificación de los subsistemas de la aplicación

No aplica, el sistema no contiene subsistemas.

6.3.4 Especificación del entorno tecnológico

En este apartado se describe el conjunto de especificaciones referentes al entorno tecnológico del sistema diseñado. Se definirán el conjunto de elementos que compondrán el eje central de la infraestructura técnica que va a adoptar el sistema, además de abordar aquellos inconvenientes técnicos que puedan afectar el desarrollo del proyecto.

6.3.4.1 Hardware

Para el desarrollo del proyecto se emplearán dos sistemas: un ordenador PC con microprocesador Intel Core 2 Quad y un ordenador portátil Hewlett-Packard.

6.3.4.2 Software

El sistema operativo instalado en la máquina empleada para la ejecución del sistema es Windows XP.



6.3.4.3 Comunicaciones

No aplica, el sistema no realiza comunicaciones.

6.3.5 Especificación de requisitos de seguridad y operación

Para analizar la especificación de los requisitos de operación se remite al lector al apartado 5.3.1.3. Requisitos de operación.

6.3.6 Estudio de la seguridad requerida en el proceso de diseño del sistema

Al tratarse de un sistema que no maneja información sensible para el usuario del propio sistema no es necesario supervisar la seguridad de las actividades del proceso de diseño.

Por otro lado, con el objetivo de evitar posibles pérdidas de información se realizará un backup diariamente.

6.3.7 Análisis de los riesgos del entorno tecnológico

No aplica, los riesgos asociados al entorno tecnológico son mínimos debido al software utilizado.

6.4 Diseño de la arquitectura de soporte

En este apartado se presentan los elementos más significativos del sistema: los componentes de la arquitectura.

Como se introdujo en el apartado 6.3. Definición de la arquitectura, el sistema tiene una arquitectura específica compuesta por tres componentes principales: el sistema experto, las redes de neuronas artificiales, y el módulo de comunicación del sistema híbrido. A continuación se describen estos componentes y la interacción entre ellos, para posteriormente analizar las clases que los componen.

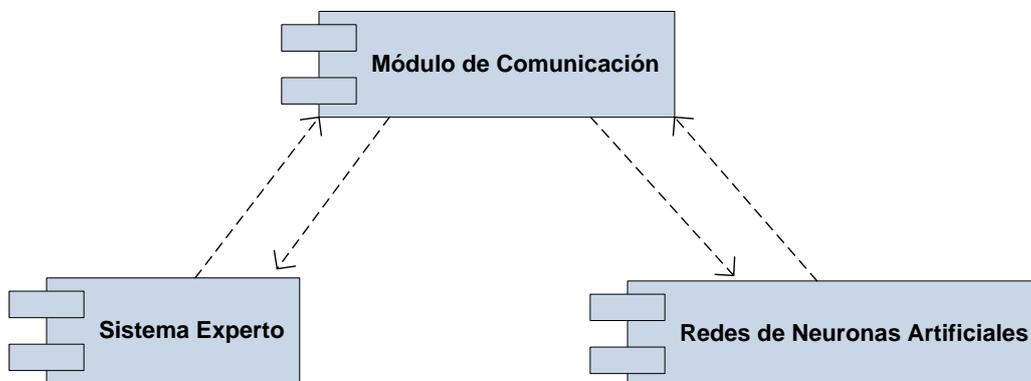


Figura 29. Componentes de la arquitectura del sistema

A continuación se definen las características de los componentes que constituyen el sistema. Para cada uno de ellos se contará con los siguientes atributos:

- **Identificador:** Será único para cada uno de los componentes. Para ello utilizaremos la siguiente nomenclatura: “C-XXX”, donde XXX es el número del componente.
- **Nombre:** Nombre del elemento.
- **Tipo:** Tipo de elemento, en este caso será componente.
- **Propósito:** Fin del componente.
- **Función:** De qué se encarga el componente.
- **Subordinados:** Hace referencia a los elementos que en algún modo heredan algún dato del componente en cuestión.



- **Dependencias:** Componente que utiliza el elemento tratado para realizar sus funciones.
- **Interfaz:** Servicios proporcionados para el resto de elementos.
- **Recursos:** Hace referencia a elementos de tipo hardware que el componente puede necesitar para realizar su función.
- **Referencias:** Documentación necesaria para poder entender el componente.
- **Proceso:** Algoritmos especiales de procesamiento.
- **Datos:** Datos internos del elemento, atributos, relaciones de interés que no se hayan indicado en la descripción, o posibles valores de alguno de los elementos.

Identificador	C-001
Nombre	Sistema Experto.
Tipo	Componente.
Propósito	Realiza una elección de parámetros en función de unos criterios inteligentes.
Función	Recoge los parámetros de entrada y se los comunica al Módulo de Comunicación para la ejecución del proceso sobre las redes de neuronas.
Subordinados	No aplica.
Dependencias	Módulo de Comunicación.
Interfaz	No aplica.
Recursos	No aplica.
Referencias	No aplica.
Proceso	No aplica.
Datos	No aplica.

Tabla 135. Componente C-001: Sistema Experto



Identificador	C-002
Nombre	Módulo de Comunicación.
Tipo	Componente.
Propósito	Intercomunicación del Sistema Experto con la Red de Neuronas.
Función	Recoge los parámetros introducidos por el Sistema Experto y los comunica a la Red de Neuronas. Una vez realizada la ejecución toma los parámetros de salida de la Red de Neuronas y los comunica al Sistema Experto.
Subordinados	No aplica.
Dependencias	Sistema Experto y Redes de Neuronas Artificiales.
Interfaz	No aplica.
Recursos	No aplica.
Referencias	No aplica.
Proceso	No aplica.
Datos	No aplica.

Tabla 136. Componente C-002: Módulo de Comunicación



Identificador	C-003
Nombre	Redes de Neuronas Artificiales.
Tipo	Componente.
Propósito	Realiza la función seleccionada por el Sistema Experto: entrenamiento, validación o puesta en producción.
Función	Recibe los parámetros del módulo de comunicación y ejecuta el proceso atendiendo a dichos parámetros de configuración.
Subordinados	No aplica.
Dependencias	Módulo de Comunicación.
Interfaz	No aplica.
Recursos	No aplica.
Referencias	No aplica.
Proceso	No aplica.
Datos	No aplica.

Tabla 137. Componente C-003: Redes de Neuronas Artificiales

6.5 Diseño de casos de uso reales

En este apartado se realiza un estudio de los casos de uso identificados en la fase de Análisis, con el fin de identificar las clases de diseño necesarias para los distintos módulos que componen el sistema, así como las relaciones entre ellas.

El objetivo es identificar el mínimo número de clases necesarias para el sistema, siempre y cuando permitan cubrir todos los requisitos especificados anteriormente.

6.5.1 Identificación de clases asociadas a un caso de uso

A continuación se muestran las clases identificadas en el diseño y su asociación con los distintos caso de uso del sistema.

6.5.1.1 Aplicación de tratamiento de ficheros

Identificador	Clases asociadas
CU-001	<ul style="list-style-type: none">▪ InterfazPrincipal.
CU-002	<ul style="list-style-type: none">▪ InterfazPrincipal.▪ InterfazImportacion.
CU-003	<ul style="list-style-type: none">▪ InterfazPrincipal.
CU-004	<ul style="list-style-type: none">▪ InterfazPrincipal.
CU-005	<ul style="list-style-type: none">▪ InterfazPrincipal.
CU-006	<ul style="list-style-type: none">▪ InterfazPrincipal.▪ GestorExperimento.▪ DivisionKFCV.▪ DivisionLOOCV.▪ DivisionRBP.▪ DivisionVKFCV.
CU-007	<ul style="list-style-type: none">▪ InterfazPrincipal.▪ GestorConfiguracion.
CU-008	<ul style="list-style-type: none">▪ InterfazPrincipal.▪ GestorConfiguracion.

Tabla 138. Clases de diseño asociadas a casos de uso (I)

6.5.1.2 Simulador del sistema experto

Identificador	Clases asociadas
CU-009	<ul style="list-style-type: none">▪ InterfazPrincipal.
CU-010	<ul style="list-style-type: none">▪ InterfazPrincipal.
CU-011	<ul style="list-style-type: none">▪ InterfazPrincipal.
CU-012	<ul style="list-style-type: none">▪ InterfazPrincipal.▪ Procesador.

Tabla 139. Clases de diseño asociadas a casos de uso (II)

6.5.1.3 Módulo de comunicación del sistema híbrido

Identificador	Clases asociadas
CU-013	<ul style="list-style-type: none">▪ Comunicacion.
CU-014	<ul style="list-style-type: none">▪ Comunicacion.▪ Proceso.▪ Entrenamiento.▪ Validacion.▪ Produccion.▪ Evaluacion.
CU-015	<ul style="list-style-type: none">▪ Comunicacion.

Tabla 140. Clases de diseño asociadas a casos de uso (III)

6.5.2 Diseño de la realización de los casos de uso

En este apartado se especifican de forma extendida los casos de uso detallados en el análisis, haciendo corresponder los distintos escenarios con acciones concretas identificadas en las clases obtenidas en el desarrollo del documento.

6.5.2.1 Aplicación de tratamiento de ficheros

El siguiente diagrama de secuencia muestra con las clases de diseño el caso de uso CU-006: Generar ficheros.

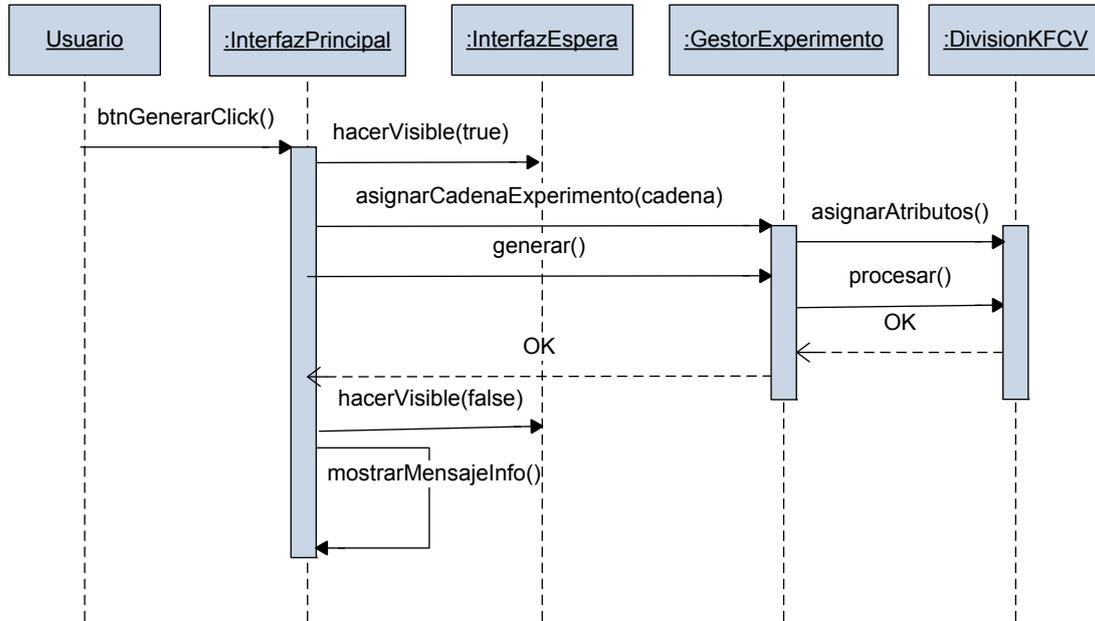


Figura 30. Diagrama de secuencia extendido del caso de uso CU-006

6.5.2.2 Simulador del sistema experto

A continuación se muestra el diagrama de secuencia del caso de uso CU-012: Ejecutar, perteneciente al simulador del sistema experto.

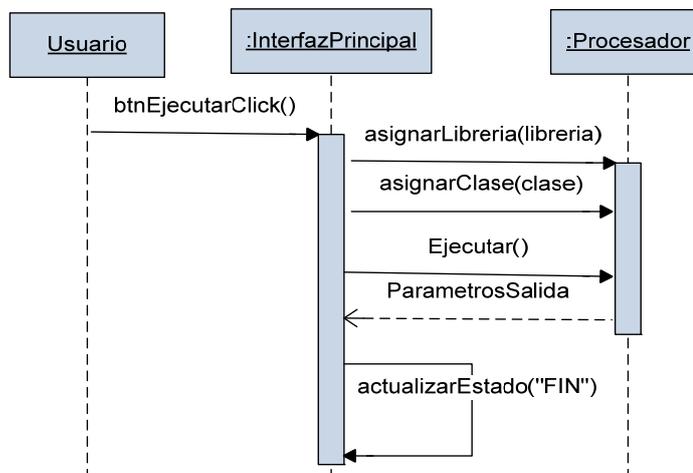


Figura 31. Diagrama de secuencia extendido del caso de uso CU-012

6.5.2.3 Módulo de comunicación del sistema híbrido

La interacción entre las clases de diseño para el caso de uso CU-014: Ejecutar queda representada en el siguiente diagrama de secuencia.

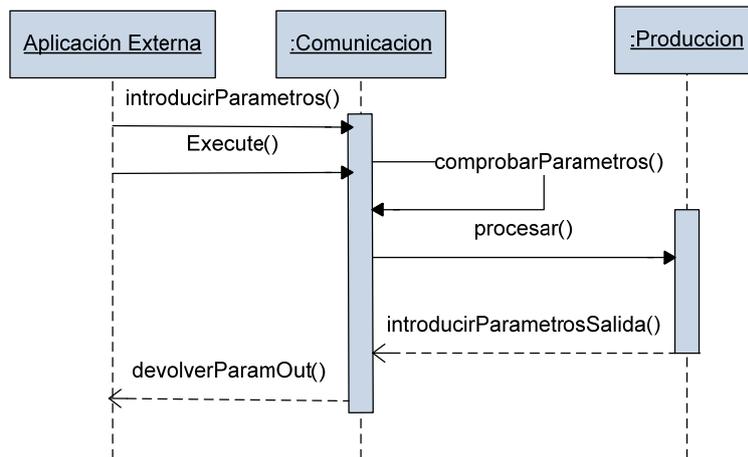


Figura 32. Diagrama de secuencia extendido del caso de uso CU-014

6.6 Diseño de clases

En esta sección se desarrolla el modelo de clases del sistema que se está diseñando, completando el realizado durante la fase de Análisis del Sistema.

Los modelos aquí presentados servirán de guía al programador, por lo que los identificadores, relaciones, atributos y funciones son orientativos, dejando al programador libertad para programar bajo su propio criterio.

6.6.1 Identificación de clases de diseño

A continuación se muestran los modelos de clases de diseño del sistema, divididos en los tres módulos diseñados:

6.6.1.1 Aplicación de tratamiento de ficheros

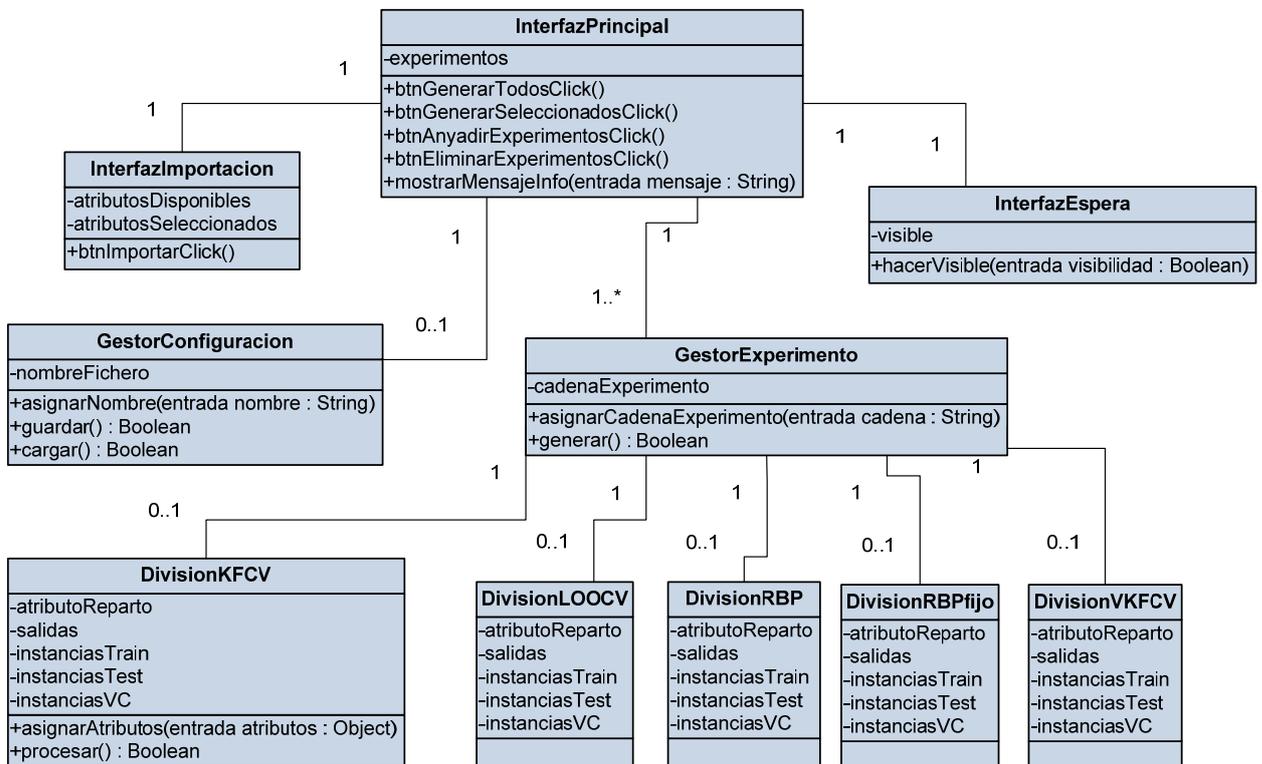


Figura 33. Diagrama de clases de diseño (I)

(NOTA: Para facilitar la visibilidad del diagrama anterior, las funciones y procedimientos de las clases *DivisionLOOCV*, *DivisionRBP*, *DivisionRBPfijo* y *DivisionVKFCV* se han omitido al ser iguales a los de la clase *DivisionKFCV*.)

6.6.1.2 Simulador del sistema experto

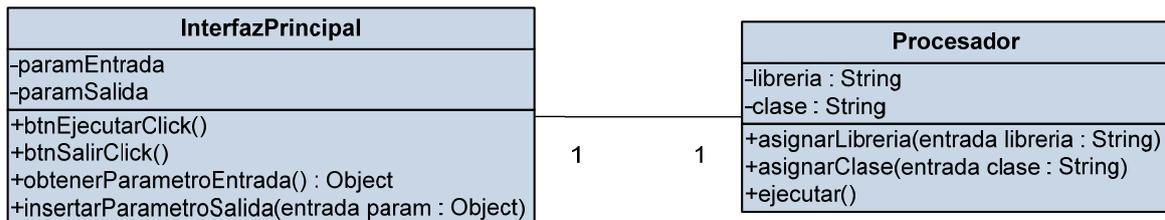


Figura 34. Diagrama de clases de diseño (II)

6.6.1.3 Módulo de comunicación del sistema híbrido

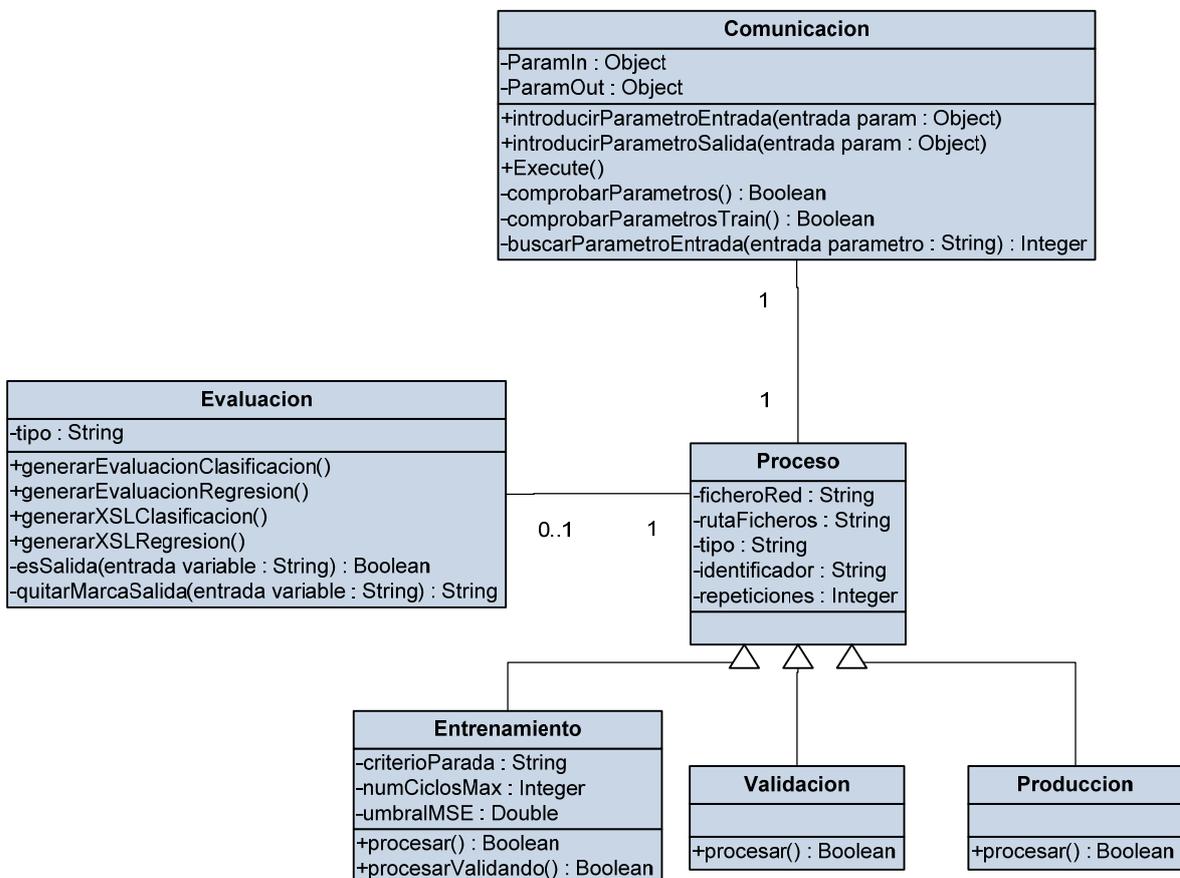


Figura 35. Diagrama de clases de diseño (III)

6.6.2 Identificación de atributos y métodos de clases de diseño

A continuación se describen las clases de diseño identificadas, incluyendo el detalle de sus atributos y métodos.

6.6.2.1 Aplicación de tratamiento de ficheros

Clase	InterfazPrincipal
Descripción	Clase encargada de gestionar la interfaz principal de la aplicación para interactuar con el usuario.
Dependencias	InterfazImportacion, InterfazEspera.
ATRIBUTOS	
Identificador	<i>experimentos</i>
Descripción	Listado de experimentos a preparar.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>btnGenerarTodosClick()</i>
Descripción	Controla la funcionalidad asociada al botón "Generar Todos" de la interfaz.
Signatura	<i>btnGenerarSeleccionadosClick()</i>
Descripción	Controla la funcionalidad asociada al botón "Generar Seleccionados" de la interfaz.
Signatura	<i>btnAnyadirExperimentosClick()</i>
Descripción	Controla la funcionalidad asociada al botón "Añadir Experimento(s)" de la interfaz.
Signatura	<i>btnEliminarExperimentosClick()</i>
Descripción	Controla la funcionalidad asociada al botón "Eliminar Experimento(s)" de la interfaz.
Signatura	<i>mostrarMensajeInfo(mensaje)</i>
Descripción	Muestra por pantalla un cuadro de diálogo informativo con el mensaje recibido como parámetro.

Tabla 141. Atributos, funciones y procedimientos clase InterfazPrincipal



Clase	InterfazImportacion
Descripción	Clase encargada de gestionar las funcionalidades proporcionadas por la interfaz para la importación de datos.
Dependencias	InterfazPrincipal.
ATRIBUTOS	
Identificador	<i>atributosDisponibles</i>
Descripción	Listado de atributos disponibles en el fichero de datos de entrada.
Identificador	<i>atributosSeleccionados</i>
Descripción	Listado de atributos seleccionados por el usuario.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>btnImportarClick()</i>
Descripción	Controla la funcionalidad asociada al botón "Importar".

Tabla 142. Atributos, funciones y procedimientos clase InterfazImportacion

Clase	InterfazEspera
Descripción	Clase encargada de gestionar la interfaz de espera, utilizada cuando la aplicación está generando los archivos para los experimentos.
Dependencias	InterfazPrincipal.
ATRIBUTOS	
Identificador	<i>visible</i>
Descripción	Indica si la interfaz es visible o no.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>hacerVisible(visibilidad)</i>
Descripción	Modifica el estado de la visibilidad de la interfaz.

Tabla 143. Atributos, funciones y procedimientos clase InterfazEspera



Clase	GestorConfiguracion
Descripción	Clase encargada de gestionar el almacenamiento y recuperación de los ficheros de configuración.
Dependencias	InterfazPrincipal.
ATRIBUTOS	
Identificador	<i>nombreFichero</i>
Descripción	Nombre del fichero de configuración a guardar o cargar.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>asignarNombre(nombre)</i>
Descripción	Asigna el valor del parámetro “nombre” al atributo “nombreFichero”.
Signatura	<i>guardar()</i>
Descripción	Almacena la configuración en el archivo con nombre “nombreFichero”.
Signatura	<i>cargar()</i>
Descripción	Recupera la configuración del archivo con nombre “nombreFichero”.

Tabla 144. Atributos, funciones y procedimientos clase GestorConfiguracion



Clase	GestorExperimento
Descripción	Clase encargada de la generación de un experimento.
Dependencias	InterfazPrincipal, DivisionKFCV, DivisionLOOCV, DivisionRBP, DivisionRBPfijo, DivisionVKFCV.
ATRIBUTOS	
Identificador	<i>cadenaExperimento</i>
Descripción	Cadena de caracteres que representa al experimento a preparar.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>asignarCadenaExperimento(cadena)</i>
Descripción	Asigna el valor "cadena" al atributo "cadenaExperimento".
Signatura	<i>generar()</i>
Descripción	Genera los archivos correspondientes al experimento identificado por su "cadenaExperimento".

Tabla 145. Atributos, funciones y procedimientos clase GestorExperimento



Clase	DivisionKFCV
Descripción	Clase encargada de realizar el reparto de instancias usando la división de tipo KFCV.
Dependencias	GestorExperimento.
ATRIBUTOS	
Identificador	<i>atributoReparto</i>
Descripción	Atributo utilizado para repartir las instancias entre los distintos conjuntos.
Identificador	<i>salidas</i>
Descripción	Conjunto de atributos seleccionados por el usuario como salidas para la predicción del sistema.
Identificador	<i>instanciasTrain</i>
Descripción	Cantidad de instancias requeridas para entrenamiento.
Identificador	<i>instanciasTest</i>
Descripción	Cantidad de instancias requeridas para validación.
Identificador	<i>instanciasVC</i>
Descripción	Cantidad de instancias requeridas para validación cruzada.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>asignarAtributos(atributos)</i>
Descripción	Asigna los atributos de reparto y salida a los atributos de la clase.
Signatura	<i>procesar()</i>
Descripción	Realiza el procesamiento del reparto de instancias.

Tabla 146. Atributos, funciones y procedimientos clase DivisionKFCV



Clase	DivisionLOOCV
Descripción	Clase encargada de realizar el reparto de instancias usando la división de tipo LOOCV.
Dependencias	GestorExperimento.
ATRIBUTOS	
Identificador	<i>atributoReparto</i>
Descripción	Atributo utilizado para repartir las instancias entre los distintos conjuntos.
Identificador	<i>salidas</i>
Descripción	Conjunto de atributos seleccionados por el usuario como salidas para la predicción del sistema.
Identificador	<i>instanciasTrain</i>
Descripción	Cantidad de instancias requeridas para entrenamiento.
Identificador	<i>instanciasTest</i>
Descripción	Cantidad de instancias requeridas para validación.
Identificador	<i>instanciasVC</i>
Descripción	Cantidad de instancias requeridas para validación cruzada.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>asignarAtributos(atributos)</i>
Descripción	Asigna los atributos de reparto y salida a los atributos de la clase.
Signatura	<i>procesar()</i>
Descripción	Realiza el procesamiento del reparto de instancias.

Tabla 147. Atributos, funciones y procedimientos clase DivisionLOOCV



Clase	DivisionRBP
Descripción	Clase encargada de realizar el reparto de instancias usando la división de tipo RBP.
Dependencias	GestorExperimento.
ATRIBUTOS	
Identificador	<i>atributoReparto</i>
Descripción	Atributo utilizado para repartir las instancias entre los distintos conjuntos.
Identificador	<i>salidas</i>
Descripción	Conjunto de atributos seleccionados por el usuario como salidas para la predicción del sistema.
Identificador	<i>instanciasTrain</i>
Descripción	Cantidad de instancias requeridas para entrenamiento.
Identificador	<i>instanciasTest</i>
Descripción	Cantidad de instancias requeridas para validación.
Identificador	<i>instanciasVC</i>
Descripción	Cantidad de instancias requeridas para validación cruzada.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>asignarAtributos(atributos)</i>
Descripción	Asigna los atributos de reparto y salida a los atributos de la clase.
Signatura	<i>procesar()</i>
Descripción	Realiza el procesamiento del reparto de instancias.

Tabla 148. Atributos, funciones y procedimientos clase DivisionRBP



Clase	DivisionRBPfijo
Descripción	Clase encargada de realizar el reparto de instancias usando la división de tipo RBP fijo.
Dependencias	GestorExperimento.
ATRIBUTOS	
Identificador	<i>atributoReparto</i>
Descripción	Atributo utilizado para repartir las instancias entre los distintos conjuntos.
Identificador	<i>salidas</i>
Descripción	Conjunto de atributos seleccionados por el usuario como salidas para la predicción del sistema.
Identificador	<i>instanciasTrain</i>
Descripción	Cantidad de instancias requeridas para entrenamiento.
Identificador	<i>instanciasTest</i>
Descripción	Cantidad de instancias requeridas para validación.
Identificador	<i>instanciasVC</i>
Descripción	Cantidad de instancias requeridas para validación cruzada.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>asignarAtributos(atributos)</i>
Descripción	Asigna los atributos de reparto y salida a los atributos de la clase.
Signatura	<i>procesar()</i>
Descripción	Realiza el procesamiento del reparto de instancias.

Tabla 149. Atributos, funciones y procedimientos clase DivisionRBPfijo



Clase	DivisionVKFCV
Descripción	Clase encargada de realizar el reparto de instancias usando la división de tipo VKFCV.
Dependencias	GestorExperimento.
ATRIBUTOS	
Identificador	<i>atributoReparto</i>
Descripción	Atributo utilizado para repartir las instancias entre los distintos conjuntos.
Identificador	<i>salidas</i>
Descripción	Conjunto de atributos seleccionados por el usuario como salidas para la predicción del sistema.
Identificador	<i>instanciasTrain</i>
Descripción	Cantidad de instancias requeridas para entrenamiento.
Identificador	<i>instanciasTest</i>
Descripción	Cantidad de instancias requeridas para validación.
Identificador	<i>instanciasVC</i>
Descripción	Cantidad de instancias requeridas para validación cruzada.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>asignarAtributos(atributos)</i>
Descripción	Asigna los atributos de reparto y salida a los atributos de la clase.
Signatura	<i>procesar()</i>
Descripción	Realiza el procesamiento del reparto de instancias.

Tabla 150. Atributos, funciones y procedimientos clase DivisionVKFCV

6.6.2.2 Simulador del sistema experto

Clase	InterfazPrincipal
Descripción	Clase encargada de interactuar con el usuario ofreciéndole todas las funcionalidades de la aplicación.
Dependencias	Procesador.
ATRIBUTOS	
Identificador	<i>paramEntrada</i>
Descripción	Listado de parámetros de entrada.
Identificador	<i>paramSalida</i>
Descripción	Listado de parámetros de salida.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>btnEjecutarClick()</i>
Descripción	Realiza la funcionalidad asociada al botón “Ejecutar” de la interfaz.
Signatura	<i>btnSalirClick()</i>
Descripción	Realiza la funcionalidad asociada al botón “Salir” de la interfaz.
Signatura	<i>obtenerParametroEntrada()</i>
Descripción	Devuelve el primer parámetro de la lista de parámetros de entrada.
Signatura	<i>insertarParametroSalida(param)</i>
Descripción	Inserta el parámetro “param” en la lista de parámetros de salida de la aplicación.

Tabla 151. Atributos, funciones y procedimientos clase InterfazPrincipal



Clase	Procesador
Descripción	Clase encargada de instanciar la librería de comunicación seleccionada por el usuario y enviarle la petición de ejecución de un proceso.
Dependencias	InterfazPrincipal.
ATRIBUTOS	
Identificador	<i>libreria</i>
Descripción	Librería de comunicación que se quiere usar.
Identificador	<i>clase</i>
Descripción	Clase de la librería de comunicación que se debe instanciar.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>asignarLibreria(libreria)</i>
Descripción	Asigna el valor del parámetro “libreria” al atributo “libreria” de la clase.
Signatura	<i>asignarClase(clase)</i>
Descripción	Asigna el valor del parámetro “clase” al atributo “clase”.
Signatura	<i>ejecutar()</i>
Descripción	Ejecuta la llamada al módulo de comunicación para ejecutar el proceso deseado.

Tabla 152. Atributos, funciones y procedimientos clase Procesador

6.6.2.3 Módulo de comunicación del sistema híbrido

Clase	Comunicacion
Descripción	Clase encargada de la interacción con el sistema experto, atendiendo a sus peticiones para comunicarlás a la red de neuronas.
Dependencias	Proceso.
ATRIBUTOS	
Identificador	<i>ParamIn</i>
Descripción	Estructura que contiene los parámetros de entrada del módulo de comunicación.
Identificador	<i>ParamOut</i>
Descripción	Estructura que contiene los parámetros de salida del módulo de comunicación.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>introducirParametroEntrada(param)</i>
Descripción	Introduce el parámetro “param” en “ParamIn”.
Signatura	<i>introducirParametroSalida(param)</i>
Descripción	Introduce el parámetro “param” en “ParamOut”.
Signatura	<i>Execute()</i>
Descripción	Inicia la ejecución del proceso configurado mediante los parámetros de entrada.
Signatura	<i>comprobarParametros()</i>
Descripción	Comprueba que los parámetros en ParamIn sean correctos.
Signatura	<i>comprobarParametrosTrain()</i>
Descripción	Comprueba que los parámetros específicos para la función ‘Train’ sean correctos.
Signatura	<i>buscarParametroEntrada(parametro)</i>
Descripción	Busca la posición del parámetro “parametro” en “ParamIn”.

Tabla 153. Atributos, funciones y procedimientos clase Comunicacion



Clase	Evaluacion
Descripción	Clase encargada de generar los ficheros de evaluación para un proceso ejecutado sobre la red de neuronas.
Dependencias	Proceso.
ATRIBUTOS	
Identificador	<i>Tipo</i>
Descripción	Tipo de evaluación a realizar: clasificación o regresión.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>generarEvaluacionClasificacion()</i>
Descripción	Genera los ficheros de evaluación para el tipo Clasificación.
Signatura	<i>generarEvaluacionRegresion()</i>
Descripción	Genera los ficheros de evaluación para el tipo Regresión.
Signatura	<i>generarXSLClasificacion()</i>
Descripción	Genera el fichero en formato XSL para el tipo Clasificación.
Signatura	<i>generarXSLRegresion()</i>
Descripción	Genera el fichero en formato XSL para el tipo Regresión.
Signatura	<i>esSalida(variable)</i>
Descripción	Comprueba si “variable” es un atributo marcado como salida.
Signatura	<i>quitarMarcaSalida(variable)</i>
Descripción	Eliminar la marca de salida de “variable”.

Tabla 154. Atributos, funciones y procedimientos clase Evaluacion



Clase	Proceso
Descripción	Superclase que representa los distintos procesos a realizar.
Dependencias	Evaluacion, Entrenamiento, Validacion, Produccion.
ATRIBUTOS	
Identificador	<i>ficheroRed</i>
Descripción	Ruta del fichero de la red de neuronas.
Identificador	<i>rutaFicheros</i>
Descripción	Ruta de los ficheros necesarios para el proceso.
Identificador	<i>tipo</i>
Descripción	Tipo de proceso: clasificación o regresión.
Identificador	<i>identificador</i>
Descripción	Identificador opcional del proceso.
Identificador	<i>repeticiones</i>
Descripción	Número de veces a repetir el mismo proceso.

Tabla 155. Atributos, funciones y procedimientos clase Proceso



Clase	Entrenamiento
Descripción	Clase encargada del proceso de entrenamiento.
Dependencias	Proceso.
ATRIBUTOS	
Identificador	<i>criteroParada</i>
Descripción	Criterio de parada que se desea usar en el entrenamiento.
Identificador	<i>numCiclosMax</i>
Descripción	Número máximo de ciclos a utilizar en el entrenamiento.
Identificador	<i>umbralMSE</i>
Descripción	Valor del umbral de error MSE a utilizar.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>procesar()</i>
Descripción	Realiza el proceso de entrenamiento.
Signatura	<i>procesarValidando()</i>
Descripción	Realiza el proceso de entrenamiento, validando la red además de entrenarla.

Tabla 156. Atributos, funciones y procedimientos clase Entrenamiento

Clase	Validacion
Descripción	Clase encargada del proceso de validación.
Dependencias	Proceso.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>procesar()</i>
Descripción	Realiza el proceso de validación.

Tabla 157. Atributos, funciones y procedimientos clase Validacion



Clase	Produccion
Descripción	Clase encargada del proceso de puesta en producción de la red de neuronas.
Dependencias	Proceso.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	<i>procesar()</i>
Descripción	Realiza el proceso de puesta en producción de la red.

Tabla 158. Atributos, funciones y procedimientos clase Produccion

6.7 Diseño físico de datos

La aplicación de tratamiento de ficheros, para el almacenamiento de los ficheros de configuración utiliza ficheros en formato Microsoft Excel. En ellos almacena tanto los experimentos de la lista “Experimentos a preparar” de la aplicación como el resto de opciones de configuración.

A continuación se muestra un ejemplo de fichero de configuración almacenado:

	A	B	C	D	E
1	C:\DAVID\Mis documentos\UNI\PFC\DATOS.xls			Nº Ensayo	Penetración
2	C:\DAVID\Escritorio			GROSOR PLACA (mm)	
3	Exp. 1 (1 pruebas con VKFCV) --> 10-10-10{Penetración}30%-70%	0		RADIO PROY (mm)	
4	Exp. 2 (1 pruebas con VKFCV) --> 10-10-10{Penetración}30%-70%	0		LONG PROY (mm)	
5	Exp. 3 (10 pruebas con VKFCV) --> 120-10-10{Penetración}30%-70%	1		VEL (m/s)	
6				MASA INICIAL (g)	
7				MATERIAL PROY	
8				MATERIAL PLACA	
9				Masa Final (g)	
10				(Mi-Mf)/Mi (%)	
11				Penetración	
12				Vres (m/s)	
13				DOP(mm)	
14					

Figura 36. Ejemplo de fichero de configuración

Por otro lado, para conocer más detalles acerca del formato necesario que deben utilizar los ficheros de entrada véase el Anexo B: Formato para ficheros de entrada.



6.8 Verificación y aceptación de la arquitectura del sistema

El objetivo de la tarea documentada en este apartado es el de garantizar la calidad y viabilidad de las especificaciones del diseño del sistema de información antes de realizar el diseño detallado.

6.8.1 Análisis de consistencia de las especificaciones de diseño

Esta tarea pretende asegurar que las especificaciones de diseño realizadas hasta el momento son coherentes, comprobando que no son ambiguas, que no hay información duplicada y que todos los casos de uso quedan cubiertos con el diseño.

Estas comprobaciones se fundamentan en técnicas matriciales o de revisión entre los elementos comunes de los distintos modelos.

6.8.1.1 Trazabilidad casos de uso - clases de diseño

	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007	CU-008	CU-009	CU-010	CU-011	CU-012	CU-013	CU-014	CU-015
InterfazPrincipal (Ficheros)	X	X	X	X	X	X	X	X							
InterfazImportacion		X													
InterfazEspera						X									
GestorExperimento						X									
GestorConfiguración							X	X							
DivisionKFCV						X									
DivisionLOOCV						X									
DivisionRBP						X									
DivisionRBPfijo						X									
DivisionVKFCV						X									



InterfazPrincipal (Simulador)									X	X	X	X			
Procesador												X			
Comunicación													X	X	X
Evaluacion														X	X
Proceso														X	X
Entrenamiento														X	X
Validacion														X	X
Produccion														X	X

Tabla 159. Matriz de trazabilidad Clases de diseño - Casos de Uso



6.9 Especificación Técnica del Plan de Pruebas

El objetivo de la tarea descrita en este apartado es el de especificar un plan detallado de las diferentes pruebas a realizar una vez implementado el sistema.

La elaboración de un plan de pruebas detallado tiene como finalidad comprobar que el sistema desarrollado cumple todas las funcionalidades requeridas, no presenta errores y proporciona un nivel de calidad suficiente.

El plan de pruebas se ha realizado tomando como punto de partida el catálogo de requisitos y el diseño detallado del sistema. Se puede analizar el plan de pruebas resultante en el apartado 7. Plan de Validación y Verificación del Software.



6.10 Establecimiento de Requisitos de Implantación

6.10.1 Especificación de Requisitos de Implantación

En este apartado se especifican los requisitos de implantación del proyecto. Estos requisitos están relacionados con la formación, infraestructura e instalación, siendo necesarios para preparar y organizar la implantación del sistema de forma adecuada.

El identificador de cada uno de los requisitos de implantación tendrá el siguiente formato: RS-IMP-XXX, donde XXX representa un valor numérico entre 000 y 999.

A continuación se detallan los requisitos de implantación identificados para el presente proyecto:

Identificador	RS-IMP-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo de desarrollo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Disponer de un ordenador con sistema operativo Microsoft Windows XP.		

Tabla 160. Requisito software RS-IMP-001



7 Plan de Validación y Verificación del Software

7.1 Introducción

7.1.1 Propósito del documento

El objetivo de este documento es presentar la definición de un plan de pruebas que permitan comprobar que el sistema funciona correctamente, cumpliendo con las especificaciones dadas por el cliente en su solicitud.

El plan de pruebas propuesto detalla todas las características a comprobar del sistema, junto con los procedimientos a llevar a cabo para realizar dichas comprobaciones. Cada una de las pruebas a realizar debe llevar asociada una descripción completa, los pasos a seguir para su realización y los criterios para la aceptación de la prueba. Asimismo se especifica también el entorno necesario para la realización de las pruebas, y el proceso de realización de informes de las pruebas realizadas.

Es destacable la importancia de una buena definición de requisitos software y hardware previa, que permita garantizar una correcta evaluación del sistema, haciendo que los resultados de las pruebas realizadas se acerquen en la mayor medida posible a los obtenidos por el sistema en el entorno de explotación.

Teniendo como guía el presente documento, obtendremos un sistema verificado y validado, funcionando correctamente y asegurando la satisfacción del cliente debido el cumplimiento de su solicitud.

Debido a las características de este proyecto, para la realización de este documento se ha utilizado una adaptación de la metodología *ESA Lite*, la cual facilita la definición de pruebas y cubre los objetivos establecidos.



7.1.2 Visión general del Plan de Validación y Verificación del Sistema

El documento está dividido en los siguientes apartados:

- **Plan de prueba:** Esta sección describe todos y cada uno de los elementos del sistema que serán probados, así como las características que serán verificadas y los productos que deberán entregarse antes y después de la realización de las pruebas. El plan de prueba abarca también las tareas necesarias para preparar y llevar a cabo cada una de las pruebas, así como el entorno necesario para la ejecución de las mismas y el criterio de aceptación o rechazo que se va a aplicar a las pruebas.
- **Procedimientos de prueba:** Recoge los procedimientos de prueba que se aplicarán sobre el sistema. Cada uno de ellos vendrá definido por un identificador único, su descripción, las condiciones necesarias en el sistema, y los pasos a seguir para llevar a cabo la prueba.
- **Plantilla de informe de prueba:** Por cada prueba que se realice sobre el sistema, se deberá rellenar un informe, siguiendo el formato y el proceso que se describe en este apartado.
- **Matriz de Trazabilidad:** Aunque la metodología *ESA Lite* no lo incluye para este documento, se realizará una matriz de trazabilidad entre requisitos de capacidad y pruebas con el fin de asegurar que todas las funcionalidades requeridas por el cliente serán probadas.



7.2 Plan de prueba

Para poder desarrollar un sistema informático de calidad que funcione de manera correcta es imprescindible establecer un buen plan de pruebas.

En este apartado se definen los objetivos de las pruebas a realizar y cómo han de llevarse a cabo. Debe quedar recogido qué elementos van a ser probados, las características del sistema que se van a identificar, los elementos entregables, las actividades a realizar para poder construir una prueba, qué requisitos debe tener el sistema para que las pruebas se ejecuten con éxito y, por último, los criterios a tener en cuenta por el equipo de desarrollo para determinar si una prueba es finalizada con éxito o no.

7.2.1 Elementos de prueba

En un proyecto software se hace imprescindible la definición de pruebas que verifiquen si el producto funciona correctamente o si, por el contrario, existen funcionalidades que no pasan los controles de calidad. Un paso previo y muy importante para que la definición de las pruebas sea la correcta es establecer qué elementos del sistema van a someterse a prueba.

Todas las funcionalidades del sistema deben ser sometidas a pruebas, pero es necesario identificar cada una de las partes involucradas en el proceso de verificación y validación. De esta forma se puede controlar de forma más exhaustiva las pruebas de carácter que se han de realizar, tanto de elementos particulares, como generales.

De esta forma se identifican los siguientes elementos que serán sometidos a pruebas:

- **Módulo de comunicación del sistema híbrido:** Como componente principal del proyecto se realizarán pruebas que permitan comprobar el correcto funcionamiento tanto de la comunicación entre redes de neuronas artificiales y sistema experto, como de las funcionalidades adicionales que proporciona.
- **Aplicación de tratamiento de ficheros:** Para la aplicación desarrollada de forma independiente al sistema híbrido se deben realizar pruebas de todas sus funcionalidades, cobrando especial importancia las técnicas de división de instancias, la importación de datos, y el funcionamiento genérico multidominio.
- **Simulador del sistema experto:** Se debe probar principalmente la comunicación con el módulo de comunicación desarrollado en el proyecto.



7.2.2 Características que se probarán

De los elementos de prueba identificados en el apartado anterior se probarán las características descritas anteriormente de manera general. A continuación se detalla con más profundidad el conjunto de características a probar.

Para el módulo de comunicación:

- Interacción con el sistema experto.
- Interacción con las redes de neuronas artificiales.
- Generación de informes de evaluación de resultados.
- Parámetros de ejecución.

Para la aplicación de tratamiento de ficheros:

- Arranque de la aplicación.
- Carga de ficheros de entrada en formato Excel.
- Selección de ruta de salida.
- Importación de datos de entrada.
- Selección del conjunto de atributos.
- Técnicas de división de instancias.
- Reparto de instancias en conjuntos de entrenamiento, test y validación.
- Almacenamiento y carga de ficheros de configuración.
- Generación de ficheros para los experimentos.

Para el simulador del sistema experto:

- Arranque de la aplicación.
- Interacción con el módulo de comunicación.
- Tratamiento de los parámetros de entrada.
- Tratamiento de los parámetros de salida.

7.2.3 Entregables

Para poder probar cada uno de los elementos y funcionalidades descritas en el apartado anterior es necesario tener cierto conocimiento acerca de cómo funciona el sistema y de los recursos que son necesarios para el mismo, además de saber qué criterios se han definido para que el sistema pueda ser considerado válido y, por tanto, cumpla con las necesidades del cliente.



Por ello, se hace indispensable disponer de una serie de documentos y recursos antes de la realización de las pruebas:

- Documento de Estudio de la Viabilidad.
- Documento de Análisis del Sistema.
- Documento de Diseño del Sistema.
- Plan de Validación y Verificación del Software.
- El software del sistema.
- Manuales de usuario del sistema.

Una vez que se dispone de los recursos necesarios para la realización de las pruebas, se procederá a ejecutar cada una de ellas en el sistema. Los resultados de las pruebas serán recogidos en un informe denominado **Informe de realización de pruebas del sistema**, en el cual se registrará toda información acerca de las pruebas realizadas. Dicho informe deberá contener una tabla por cada prueba realizada, siguiendo la plantilla definida para tal fin en el apartado 7.4. *Plantilla de informe de pruebas*.

7.2.4 Tareas de prueba

Desarrollar un sistema de calidad que cumpla con el resultado esperado por el cliente requiere definir un conjunto de pruebas lo más completo posible, de forma que cada una de las funcionalidades del sistema se compruebe correctamente.

Debido a que el conjunto de pruebas puede llegar a ser excesivamente numeroso se hace necesaria la definición de un conjunto de tareas que sirvan de guía para preparar y desarrollar cada una de las pruebas.

Por lo tanto, las tareas definidas para desarrollar las pruebas de este proyecto son las siguientes:

- **Definición de la prueba de aceptación:** Se debe realizar una descripción acerca de la prueba, especificando cual es el objetivo de la misma, qué elementos del sistema serán sometidos a prueba y las características que debe tener el sistema para que la prueba pueda ser ejecutada. Además se detallará cada uno de los pasos a realizar para la correcta ejecución de la prueba.
- **Ejecución de la prueba de aceptación:** Observando el comportamiento del sistema y los resultados obtenidos se podrá determinar si la prueba es superada con éxito o no.



- **Registro de la prueba de aceptación:** Deberá generarse una tabla con la información acerca de la prueba, los resultados y los posibles fallos encontrados. Esta tabla debe cumplir el formato definido en el apartado 7.4. Plantilla de informe de pruebas de este documento y ser incluida en el *Informe de realización de pruebas del sistema*. El documento con el resultado de las pruebas se adjuntará con la documentación del proyecto, pero no se incluirá en la memoria. El nombre de este documento será "Informe de Pruebas del Sistema.doc"

7.2.5 Necesidades del entorno

Para que las pruebas definidas puedan ser ejecutadas de manera satisfactoria y produzcan los mismos resultados que en el entorno en el que el sistema real debe ser implantado, es necesario definir las características que debe tener el sistema donde se van a realizar las pruebas.

En el caso de este proyecto, el entorno de pruebas tendrá las mismas características que el entorno de implantación del sistema, por lo que en este apartado no es necesario definir nuevas necesidades para el entorno.

7.2.6 Criterio de aceptación/rechazo de un caso de prueba

Al realizar la ejecución de cada uno de los casos de prueba definidos en el plan, es necesario determinar si la prueba se considera válida porque el sistema funciona como debería, o si presenta algún tipo de fallo en alguna de las funcionalidades requeridas.

El criterio establecido en este proyecto para considerar una prueba como *Aceptada*, considera necesario que dicha prueba se ejecute según lo definido, es decir, el sistema tendrá que responder de la manera exacta que detalle el caso de prueba correspondiente. En el caso de que el sistema produzca algún tipo de fallo o proporcione un resultado diferente al esperado, la prueba será *Rechazada*.



7.3 Procedimientos de prueba

En este apartado se especifican las pruebas que se deben realizar al sistema para comprobar su correcto funcionamiento y que cumple con todas las especificaciones.

A continuación se describe el formato de tabla utilizado para la especificación de las pruebas y los distintos campos que contiene:

- **Identificador:** Nombra de manera única a la prueba siguiendo el formato **PR-XXX**, donde **XXX** tomará valores numéricos dentro del rango 000-999.
- **Descripción:** Especifica el objetivo de la prueba.
- **Precondiciones:** Describe el estado en el que se debe encontrar el sistema de forma previa a la realización de la prueba.
- **Pasos a seguir:** Establece la secuencia de pasos que debe realizar el responsable de la prueba sobre el sistema.

Los procedimientos de prueba establecidos para este proyecto se describen a continuación.

7.3.1 Aplicación de tratamiento de ficheros

Identificador	PR-001
Descripción	Arrancar la aplicación correctamente.
Precondiciones	Ninguna.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer doble clic sobre el icono de la aplicación.2. Comprobar que la aplicación arranca correctamente.

Tabla 161. Prueba PR-001



Identificador	PR-002
Descripción	Seleccionar un fichero de entrada.
Precondiciones	Disponer de un fichero de datos en formato Excel.
Pasos a seguir	<ol style="list-style-type: none">1. Arrancar la aplicación (ver PR-001).2. Hacer clic sobre el botón “Examinar...” situado junto al cuadro de texto “Fichero de entrada”.3. Seleccionar un fichero.4. Hacer clic sobre “Abrir”.5. Comprobar que la ruta del fichero aparece correctamente en el cuadro de texto “Fichero de entrada”.

Tabla 162. Prueba PR-002

Identificador	PR-003
Descripción	Seleccionar una ruta de salida.
Precondiciones	Ninguna.
Pasos a seguir	<ol style="list-style-type: none">1. Arrancar la aplicación (ver PR-001).2. Hacer clic sobre el botón “Examinar...” situado junto al cuadro de texto “Ruta de salida”.3. Seleccionar una carpeta.4. Hacer clic sobre “Aceptar”.5. Comprobar que la ruta de salida aparece correctamente en el cuadro de texto “Ruta de salida”.

Tabla 163. Prueba PR-003



Identificador	PR-004
Descripción	Importar datos de un fichero de entrada con formato correcto (ver Anexo B: Formato para ficheros de entrada).
Precondiciones	Aplicación arrancada con un fichero de entrada seleccionado.
Pasos a seguir	<ol style="list-style-type: none">1. Seleccionar la hoja en la que se encuentren los datos en el cuadro de texto "Hoja del libro Excel".2. Hacer clic sobre el botón "Importar datos".3. Comprobar que la aplicación muestra la ventana de selección de atributos.4. Seleccionar al menos dos atributos entre los disponibles.5. Hacer clic sobre el botón "Importar".6. Comprobar que los atributos seleccionados aparecen correctamente en la lista "Atributos del fichero de datos".

Tabla 164. Prueba PR-004

Identificador	PR-005
Descripción	Importar datos de un fichero de entrada con formato correcto (ver Anexo B: Formato para ficheros de entrada) sin seleccionar una hoja de datos.
Precondiciones	Aplicación arrancada con un fichero de entrada seleccionado.
Pasos a seguir	<ol style="list-style-type: none">1. No seleccionar ninguna hoja en el cuadro de texto "Hoja del libro Excel".2. Hacer clic sobre el botón "Importar datos".3. Comprobar que la aplicación muestra el mensaje: "Debe seleccionar una hoja del libro Excel para importar sus datos".

Tabla 165. Prueba PR-005



Identificador	PR-006
Descripción	Seleccionar un atributo de salida.
Precondiciones	Aplicación arrancada con al menos dos atributos importados a la lista "Atributos del fichero de datos".
Pasos a seguir	<ol style="list-style-type: none">1. Seleccionar un atributo de la lista "Atributos del fichero de datos".2. Hacer clic sobre el botón "Seleccionar salida(s)".3. Comprobar que el atributo seleccionado aparece correctamente en la lista "Atributos de salida".

Tabla 166. Prueba PR-006

Identificador	PR-007
Descripción	Seleccionar múltiples atributos de salida.
Precondiciones	Aplicación arrancada con al menos tres atributos importados a la lista "Atributos del fichero de datos".
Pasos a seguir	<ol style="list-style-type: none">1. Seleccionar dos atributos de la lista "Atributos del fichero de datos".2. Hacer clic sobre el botón "Seleccionar salida(s)".3. Comprobar que los atributos seleccionados aparecen correctamente en la lista "Atributos de salida".

Tabla 167. Prueba PR-007

Identificador	PR-008
Descripción	Seleccionar un atributo de salida sin importar datos a la lista "Atributos del fichero de datos".
Precondiciones	Aplicación arrancada sin atributos importados a la lista "Atributos del fichero de datos".
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón "Seleccionar salida(s)".2. Comprobar que la aplicación muestra el siguiente mensaje: "Se debe cargar un fichero de datos y tener al menos un atributo de la lista seleccionado".

Tabla 168. Prueba PR-008



Identificador	PR-009
Descripción	Seleccionar un atributo de salida sin seleccionar ningún atributo de la lista “Atributos del fichero de datos”.
Precondiciones	Aplicación arrancada sin seleccionar atributos de la lista “Atributos del fichero de datos”.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Seleccionar salida(s)”.2. Comprobar que la aplicación muestra el siguiente mensaje: “Se debe cargar un fichero de datos y tener al menos un atributo de la lista seleccionado”.

Tabla 169. Prueba PR-009

Identificador	PR-010
Descripción	Cargar clases del atributo para reparto de instancias.
Precondiciones	Aplicación arrancada con atributos importados a la lista “Atributos del fichero de datos”.
Pasos a seguir	<ol style="list-style-type: none">1. Seleccionar un atributo de la lista “Atributos del fichero de datos”.2. Hacer clic sobre el botón “Cargar clases para reparto”.3. Comprobar que el atributo seleccionado aparece correctamente en el cuadro de texto “Atributo seleccionado” y aparecen las clases y número de instancias existentes de cada tipo en la tabla de “Reparto de instancias”.

Tabla 170. Prueba PR-010

Identificador	PR-011
Descripción	Cargar clases del atributo para reparto de instancias sin seleccionar ningún atributo de la lista “Atributos del fichero de datos”.
Precondiciones	Aplicación arrancada con atributos importados a la lista “Atributos del fichero de datos”.
Pasos a seguir	<ol style="list-style-type: none">1. No seleccionar ningún atributo de la lista “Atributos del fichero de datos”.2. Hacer clic sobre el botón “Cargar clases para reparto”.3. Comprobar que la aplicación muestra el siguiente mensaje: “Se debe cargar un fichero de datos y tener un atributo de la lista seleccionado”.

Tabla 171. Prueba PR-011



Identificador	PR-012
Descripción	Cargar clases del atributo para reparto de instancias sin importar ningún atributo a la lista "Atributos del fichero de datos".
Precondiciones	Aplicación arrancada sin atributos importados a la lista "Atributos del fichero de datos".
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón "Cargar clases para reparto".2. Comprobar que la aplicación muestra el siguiente mensaje: "Se debe cargar un fichero de datos y tener un atributo de la lista seleccionado".

Tabla 172. Prueba PR-012

Identificador	PR-013
Descripción	Asignar los porcentajes de reparto de instancias de forma aleatoria.
Precondiciones	Aplicación arrancada con un atributo seleccionado para el reparto de instancias (ver PR-010).
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón "Porcentajes aleatorios".2. Comprobar que a cada clase se le asigna un porcentaje mayor que cero, y que la suma de todos los porcentajes sea igual a cien.

Tabla 173. Prueba PR-013

Identificador	PR-014
Descripción	Repartir las instancias disponibles entre los conjuntos a generar.
Precondiciones	Aplicación arrancada con los datos del fichero de entrada importados.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir un número de instancias mayor que cero para los conjuntos de entrenamiento, validación y test.2. Comprobar que los valores se pueden introducir correctamente y que la etiqueta de texto "Restantes" va decrementando el número de instancias restantes.

Tabla 174. Prueba PR-014



Identificador	PR-015
Descripción	Añadir un experimento a la lista “Experimentos a generar”.
Precondiciones	Aplicación arrancada, habiendo configurado todos los parámetros de configuración de un experimento.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Añadir experimento(s)”.2. Comprobar que el experimento configurado se inserta correctamente en la última posición de la lista “Experimentos a generar” con un nombre que incluya la siguiente información:<ul style="list-style-type: none">▪ Número de experimento.▪ Número de pruebas.▪ Técnica de división.▪ Número de instancias de los conjuntos.▪ Atributo clase.▪ Porcentajes de reparto de las instancias.

Tabla 175. Prueba PR-015

Identificador	PR-016
Descripción	Añadir un experimento a la lista “Experimentos a generar” sin seleccionar atributos de salida.
Precondiciones	Aplicación arrancada, habiendo configurado todos los parámetros de configuración de un experimento excepto los atributos de salida.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Añadir experimento(s)”.2. Comprobar que el experimento no se añade a la lista “Experimentos a generar” y la aplicación muestra el siguiente mensaje: “Debe cargarse un fichero de entrada, seleccionar al menos un atributo de salida, y establecer el reparto de clases”.

Tabla 176. Prueba PR-016



Identificador	PR-017
Descripción	Añadir un experimento a la lista “Experimentos a generar” sin seleccionar un fichero de entrada.
Precondiciones	Aplicación arrancada, habiendo configurado todos los parámetros de configuración de un experimento excepto el fichero de entrada.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Añadir experimento(s)”.2. Comprobar que el experimento no se añade a la lista “Experimentos a generar” y la aplicación muestra el siguiente mensaje: “Debe cargarse un fichero de entrada, seleccionar al menos un atributo de salida, y establecer el reparto de clases”.

Tabla 177. Prueba PR-017

Identificador	PR-018
Descripción	Añadir un experimento a la lista “Experimentos a generar” sin realizar el reparto de instancias.
Precondiciones	Aplicación arrancada, habiendo configurado todos los parámetros de configuración de un experimento excepto el reparto de instancias.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Añadir experimento(s)”.2. Comprobar que el experimento no se añade a la lista “Experimentos a generar” y la aplicación muestra el siguiente mensaje: “Debe cargarse un fichero de entrada, seleccionar al menos un atributo de salida, y establecer el reparto de clases”.

Tabla 178. Prueba PR-018



Identificador	PR-019
Descripción	Añadir un experimento a la lista “Experimentos a generar” con un valor no numérico en el campo “Número de pruebas”.
Precondiciones	Aplicación arrancada, habiendo configurado todos los parámetros de configuración de un experimento.
Pasos a seguir	<ol style="list-style-type: none">1. En el campo “Número de pruebas” introducir un valor no numérico: “abc”.2. Hacer clic sobre el botón “Añadir experimento(s)”.3. Comprobar que el experimento no se añade a la lista “Experimentos a generar” y la aplicación muestra el siguiente mensaje: “Los números de pruebas e instancias deben contener valores numéricos”.

Tabla 179. Prueba PR-019

Identificador	PR-020
Descripción	Añadir un experimento a la lista “Experimentos a generar” con un valor en el campo “Número de pruebas” menor o igual a cero.
Precondiciones	Aplicación arrancada, habiendo configurado todos los parámetros de configuración de un experimento.
Pasos a seguir	<ol style="list-style-type: none">1. En el campo “Número de pruebas” introducir el valor: 0.2. Hacer clic sobre el botón “Añadir experimento(s)”.3. Comprobar que el experimento no se añade a la lista “Experimentos a generar” y la aplicación muestra el siguiente mensaje: “No se puede añadir un experimento con número de pruebas menor o igual a cero”.

Tabla 180. Prueba PR-020



Identificador	PR-021
Descripción	Añadir un experimento a la lista “Experimentos a generar” con una cantidad de instancias repartidas superior a la de instancias disponibles.
Precondiciones	Aplicación arrancada, habiendo configurado todos los parámetros de configuración de un experimento.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir valores en los conjuntos de entrenamiento, validación y test, cuya suma total sea superior al valor de “Número de instancias”.2. Hacer clic sobre el botón “Añadir experimento(s)”.3. Comprobar que el experimento no se añade a la lista “Experimentos a generar” y la aplicación muestra el siguiente mensaje: “La suma del número de instancias repartidas es superior al número de instancias disponibles”.

Tabla 181. Prueba PR-021

Identificador	PR-022
Descripción	Añadir un experimento a la lista “Experimentos a generar” con porcentajes de reparto de instancias cuya suma no sea 100.
Precondiciones	Aplicación arrancada, habiendo configurado todos los parámetros de configuración de un experimento.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir valores para los porcentajes de reparto de clases cuya suma sea distinta de 100.2. Hacer clic sobre el botón “Añadir experimento(s)”.3. Comprobar que el experimento no se añade a la lista “Experimentos a generar” y la aplicación muestra el siguiente mensaje: “El reparto de porcentajes entre las clases debe sumar 100%”.

Tabla 182. Prueba PR-022



Identificador	PR-023
Descripción	Añadir varios experimentos a la lista “Experimentos a generar” de forma automática.
Precondiciones	Aplicación arrancada, habiendo configurado todos los parámetros de configuración de un experimento.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir un valor superior a cero en el cuadro de texto “Número de decrementos” y en al menos uno de los decrementos asociados a los conjuntos de entrenamiento, validación y test.2. Hacer clic sobre el botón “Añadir experimento(s)”.3. Comprobar que los experimentos configurado se insertan al final de la lista “Experimentos a generar” con un nombre que incluya la siguiente información:<ul style="list-style-type: none">▪ Número de experimento.▪ Número de pruebas.▪ Técnica de división.▪ Número de instancias de los conjuntos.▪ Atributo de reparto.▪ Porcentajes de reparto de las instancias.4. Comprobar que los experimentos añadidos cumplen con los decrementos asignados en el paso 1.

Tabla 183. Prueba PR-023

Identificador	PR-024
Descripción	Eliminar un experimento de la lista “Experimentos a generar”.
Precondiciones	Aplicación arrancada, habiendo añadido al menos un experimento a la lista “Experimentos a generar”.
Pasos a seguir	<ol style="list-style-type: none">1. Seleccionar el experimento a eliminar.2. Hacer clic sobre el botón “Eliminar experimento(s)”.3. Comprobar que la aplicación muestra el siguiente mensaje: “Experimento(s) eliminado(s) correctamente” y que el experimento seleccionado desaparece de la lista “Experimentos a generar”.

Tabla 184. Prueba PR-024



Identificador	PR-025
Descripción	Eliminar varios experimentos de la lista “Experimentos a generar”.
Precondiciones	Aplicación arrancada, habiendo añadido al menos dos experimentos a la lista “Experimentos a generar”.
Pasos a seguir	<ol style="list-style-type: none">1. Seleccionar al menos dos experimentos a eliminar.2. Hacer clic sobre el botón “Eliminar experimento(s)”.3. Comprobar que la aplicación muestra el siguiente mensaje: “Experimento(s) eliminado(s) correctamente” y que los experimentos seleccionados desaparecen de la lista “Experimentos a generar”.

Tabla 185. Prueba PR-025

Identificador	PR-026
Descripción	Eliminar un experimento de la lista “Experimentos a generar” sin seleccionar ningún experimento.
Precondiciones	Aplicación arrancada, habiendo añadido al menos un experimento a la lista “Experimentos a generar”.
Pasos a seguir	<ol style="list-style-type: none">1. No seleccionar ningún experimento de la lista “Experimentos a generar”.2. Hacer clic sobre el botón “Eliminar experimento(s)”.3. Comprobar que la aplicación muestra el siguiente mensaje: “No hay experimentos seleccionados para eliminar”.

Tabla 186. Prueba PR-026



Identificador	PR-027
Descripción	Guardar configuración de la aplicación.
Precondiciones	Aplicación arrancada, habiendo añadido al menos un experimento a la lista “Experimentos a generar”, y habiendo seleccionado tanto fichero de entrada como ruta de salida.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Guardar configuración”.2. Introducir un nombre de archivo o seleccionar uno existente.3. Hacer clic sobre el botón “Guardar”.4. Comprobar que la aplicación muestra el siguiente mensaje: “Fichero de configuración guardado correctamente”.

Tabla 187. Prueba PR-027

Identificador	PR-028
Descripción	Guardar configuración de la aplicación sin seleccionar un fichero de entrada.
Precondiciones	Aplicación arrancada, habiendo seleccionado una ruta de salida.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Guardar configuración”.2. Comprobar que la aplicación muestra el siguiente mensaje: “Debe tener al menos la localización de los ficheros y un experimento para guardar la configuración”.

Tabla 188. Prueba PR-028



Identificador	PR-029
Descripción	Guardar configuración de la aplicación sin seleccionar una ruta de salida.
Precondiciones	Aplicación arrancada, habiendo seleccionado un fichero de entrada y habiendo añadido al menos un experimento a la lista “Experimentos a generar”.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Guardar configuración”.2. Comprobar que la aplicación muestra el siguiente mensaje: ”Debe tener al menos la localización de los ficheros y un experimento para guardar la configuración”.

Tabla 189. Prueba PR-029

Identificador	PR-030
Descripción	Guardar configuración de la aplicación sin añadir ningún experimento a la lista “Experimentos a generar”.
Precondiciones	Aplicación arrancada, habiendo seleccionado un fichero de entrada y una ruta de salida.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Guardar configuración”.2. Comprobar que la aplicación muestra el siguiente mensaje: ”Debe tener al menos la localización de los ficheros y un experimento para guardar la configuración”.

Tabla 190. Prueba PR-030

Identificador	PR-031
Descripción	Cargar configuración de la aplicación.
Precondiciones	Aplicación arrancada, habiendo almacenado previamente la configuración en un fichero (ver PR-027).
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Cargar configuración”.2. Seleccionar el fichero de configuración existente.3. Hacer clic sobre el botón “Abrir”.4. Comprobar que la aplicación carga la configuración correctamente y muestra el siguiente mensaje: “Fichero de configuración cargado correctamente”.

Tabla 191. Prueba PR-031



Identificador	PR-032
Descripción	Generar un único experimento de la lista “Experimentos a generar”.
Precondiciones	Aplicación arrancada, habiendo añadido al menos dos experimentos a la lista “Experimentos a generar”.
Pasos a seguir	<ol style="list-style-type: none">1. Seleccionar un experimento de la lista “Experimentos a generar”.2. Hacer clic sobre el botón “Generar Seleccionados”.3. Comprobar que la aplicación muestra el mensaje: “Ficheros generados correctamente”.4. Comprobar en la ruta de salida introducida que únicamente se han generado correctamente los archivos correspondientes al experimento seleccionado.

Tabla 192. Prueba PR-032

Identificador	PR-033
Descripción	Generar todos los experimentos de la lista “Experimentos a generar”.
Precondiciones	Aplicación arrancada, habiendo añadido al menos dos experimentos a la lista “Experimentos a generar”.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Generar Todos”.2. Comprobar que la aplicación muestra el mensaje: “Ficheros generados correctamente”.3. Comprobar en la ruta de salida introducida que se han generado correctamente los archivos para cada uno de los experimentos de la lista “Experimentos a generar”.

Tabla 193. Prueba PR-033



Identificador	PR-034
Descripción	Generar todos los experimentos de la lista “Experimentos a generar” sin ningún experimento añadido.
Precondiciones	Aplicación arrancada, no habiendo añadido ningún experimento a la lista “Experimentos a generar”.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Generar Todos”.2. Comprobar que la aplicación muestra el mensaje: “Debe añadir al menos un experimento para generar los ficheros”.

Tabla 194. Prueba PR-034

Identificador	PR-035
Descripción	Generar los experimentos seleccionados de la lista “Experimentos a generar” sin ningún experimento añadido.
Precondiciones	Aplicación arrancada, no habiendo añadido ningún experimento a la lista “Experimentos a generar”.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Generar Seleccionados”.2. Comprobar que la aplicación muestra el mensaje: “Debe añadir al menos un experimento para generar los ficheros”.

Tabla 195. Prueba PR-035

Identificador	PR-036
Descripción	Salir de la aplicación.
Precondiciones	Aplicación arrancada.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Salir”.2. Comprobar que la aplicación se cierra correctamente.

Tabla 196. Prueba PR-036



7.3.2 Simulador del sistema experto

Identificador	PR-037
Descripción	Arrancar la aplicación del simulador correctamente.
Precondiciones	Ninguna.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer doble clic sobre el icono de la aplicación.2. Comprobar que la aplicación arranca correctamente.

Tabla 197. Prueba PR-037

Identificador	PR-038
Descripción	Introducir parámetros de entrada.
Precondiciones	Aplicación arrancada.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre la primera línea de la tabla "Parámetros de entrada".2. Introducir un nombre.3. Introducir un valor.4. Comprobar que se puede repetir la secuencia de pasos del 1 al 3 tres veces.

Tabla 198. Prueba PR-038

Identificador	PR-039
Descripción	Introducir parámetros de salida.
Precondiciones	Aplicación arrancada.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre la tabla "Parámetros de salida".2. Comprobar que no es posible introducir ningún parámetro desde teclado.

Tabla 199. Prueba PR-039



Identificador	PR-040
Descripción	Llamar al módulo de comunicación.
Precondiciones	Aplicación arrancada.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir en el cuadro de texto “Librería” el nombre <i>ModuloCom</i>.2. Introducir en el cuadro de texto “Clase a instanciar” el nombre <i>Comunicacion</i>.3. Hacer clic sobre el botón “Ejecutar”.4. Comprobar que la ventana de estado muestra los mensajes: “Objeto tipo Comunicacion instanciado correctamente” y “Ejecutando función Execute()...”

Tabla 200. Prueba PR-040

Identificador	PR-041
Descripción	Ejecutar simulador sin introducir nombre de clase del módulo de comunicación.
Precondiciones	Aplicación arrancada y cuadro de texto “Clase a instanciar” vacío.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Ejecutar”.2. Comprobar que la aplicación muestra un cuadro de diálogo con el mensaje “Debe introducir una librería y una clase a instanciar de dicha librería”.

Tabla 201. Prueba PR-041

Identificador	PR-042
Descripción	Ejecutar simulador sin introducir nombre de la librería del módulo de comunicación.
Precondiciones	Aplicación arrancada y cuadro de texto “Librería” vacío.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Ejecutar”.2. Comprobar que la aplicación muestra un cuadro de diálogo con el mensaje “Debe introducir una librería y una clase a instanciar de dicha librería”.

Tabla 202. Prueba PR-042



Identificador	PR-043
Descripción	Salir de la aplicación.
Precondiciones	Aplicación arrancada.
Pasos a seguir	<ol style="list-style-type: none">1. Hacer clic sobre el botón “Salir”.2. Comprobar que la aplicación se cierra correctamente.

Tabla 203. Prueba PR-043

7.3.3 Módulo de comunicación del sistema híbrido

Dado que el módulo de comunicación implementado no cuenta con una interfaz que le permita interactuar directamente con el usuario, en las siguientes pruebas se utilizará el simulador del sistema experto para interactuar con el módulo de comunicación.

Identificador	PR-044
Descripción	Ejecutar módulo de comunicación con función Train.
Precondiciones	Simulador arrancado.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir en el cuadro de texto “Librería” el nombre <i>ModuloCom</i>.2. Introducir en el cuadro de texto “Clase a instanciar” el nombre <i>Comunicacion</i>.3. Introducir los siguientes parámetros de entrada:<ul style="list-style-type: none">▪ Función (TRAIN)▪ NumCiclosMax▪ CriterioParada▪ RutaFicheros▪ FicheroRed▪ Tipo4. Hacer clic sobre el botón “Ejecutar”.5. Comprobar que en la ventana de “Estado” aparece el mensaje: “Parámetros de salida cargados de ParamOut correctamente.”

Tabla 204. Prueba PR-044



Identificador	PR-045
Descripción	Ejecutar módulo de comunicación con función Test.
Precondiciones	Simulador arrancado.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir en el cuadro de texto “Librería” el nombre <i>ModuloCom</i>.2. Introducir en el cuadro de texto “Clase a instanciar” el nombre <i>Comunicacion</i>.3. Introducir los siguientes parámetros de entrada:<ul style="list-style-type: none">▪ Función (TEST)▪ RutaFicheros▪ FicheroRed▪ Tipo4. Hacer clic sobre el botón “Ejecutar”.5. Comprobar que en la ventana de “Estado” aparece el mensaje: “Parámetros de salida cargados de ParamOut correctamente.”6. Comprobar que el módulo de comunicación interacciona con el programa de redes de neuronas artificiales y que aparecen como parámetros de salida los parámetros ‘EvaluacionEXCEL’ y ‘EvaluacionXML’.

Tabla 205. Prueba PR-045



Identificador	PR-046
Descripción	Ejecutar módulo de comunicación con función Producción.
Precondiciones	Simulador arrancado.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir en el cuadro de texto “Librería” el nombre <i>ModuloCom</i>.2. Introducir en el cuadro de texto “Clase a instanciar” el nombre <i>Comunicacion</i>.3. Introducir los siguientes parámetros de entrada:<ul style="list-style-type: none">▪ Función (PRODUCCION)▪ RutaFicheros▪ FicheroRed▪ Tipo4. Hacer clic sobre el botón “Ejecutar”.5. Comprobar que en la ventana de “Estado” aparece el mensaje: “Parámetros de salida cargados de ParamOut correctamente.”6. Comprobar que el módulo de comunicación interacciona con el programa de redes de neuronas artificiales y que aparecen como parámetros de salida los parámetros ‘Resultado’, ‘EvaluacionEXCEL’ y ‘EvaluacionXML’.

Tabla 206. Prueba PR-046



Identificador	PR-047
Descripción	Ejecutar módulo de comunicación sin el parámetro “Tipo”.
Precondiciones	Simulador arrancado.
Pasos a seguir	<ol style="list-style-type: none">7. Introducir en el cuadro de texto “Librería” el nombre <i>ModuloCom</i>.8. Introducir en el cuadro de texto “Clase a instanciar” el nombre <i>Comunicacion</i>.9. Introducir los siguientes parámetros de entrada:<ul style="list-style-type: none">▪ Función▪ NumCiclosMax▪ CriterioParada▪ RutaFicheros▪ FicheroRed10. Hacer clic sobre el botón “Ejecutar”.11. Comprobar que aparece como parámetro de salida un error que indica lo siguiente: ”Parámetro obligatorio ‘Tipo’ no definido”.

Tabla 207. Prueba PR-047

Identificador	PR-048
Descripción	Ejecutar módulo de comunicación sin el parámetro “NumCiclosMax”.
Precondiciones	Simulador arrancado.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir en el cuadro de texto “Librería” el nombre <i>ModuloCom</i>.2. Introducir en el cuadro de texto “Clase a instanciar” el nombre <i>Comunicacion</i>.3. Introducir los siguientes parámetros de entrada:<ul style="list-style-type: none">▪ Función (TRAIN)▪ Tipo▪ CriterioParada▪ RutaFicheros▪ FicheroRed4. Hacer clic sobre el botón “Ejecutar”.5. Comprobar que aparece como parámetro de salida un error que indica lo siguiente: ”Parámetro obligatorio para Train ‘NumCiclosMax’ no definido”.

Tabla 208. Prueba PR-048



Identificador	PR-049
Descripción	Ejecutar módulo de comunicación sin el parámetro "CriterioParada".
Precondiciones	Simulador arrancado.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir en el cuadro de texto "Librería" el nombre <i>ModuloCom</i>.2. Introducir en el cuadro de texto "Clase a instanciar" el nombre <i>Comunicacion</i>.3. Introducir los siguientes parámetros de entrada:<ul style="list-style-type: none">▪ Función (TRAIN)▪ NumCiclosMax▪ Tipo▪ RutaFicheros▪ FicheroRed4. Hacer clic sobre el botón "Ejecutar".5. Comprobar que aparece como parámetro de salida un error que indica lo siguiente: "Parámetro obligatorio para Train 'CriterioParada' no definido".

Tabla 209. Prueba PR-049



Identificador	PR-050
Descripción	Ejecutar módulo de comunicación sin el parámetro "RutaFicheros".
Precondiciones	Simulador arrancado.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir en el cuadro de texto "Librería" el nombre <i>ModuloCom</i>.2. Introducir en el cuadro de texto "Clase a instanciar" el nombre <i>Comunicacion</i>.3. Introducir los siguientes parámetros de entrada:<ul style="list-style-type: none">▪ Función▪ NumCiclosMax▪ CriterioParada▪ Tipo▪ FicheroRed4. Hacer clic sobre el botón "Ejecutar".5. Comprobar que aparece como parámetro de salida un error que indica lo siguiente: "Parámetro obligatorio 'RutaFicheros' no definido".

Tabla 210. Prueba PR-050



Identificador	PR-051
Descripción	Ejecutar módulo de comunicación sin el parámetro "FicheroRed".
Precondiciones	Simulador arrancado.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir en el cuadro de texto "Librería" el nombre <i>ModuloCom</i>.2. Introducir en el cuadro de texto "Clase a instanciar" el nombre <i>Comunicacion</i>.3. Introducir los siguientes parámetros de entrada:<ul style="list-style-type: none">▪ Función▪ NumCiclosMax▪ CriterioParada▪ RutaFicheros▪ Tipo4. Hacer clic sobre el botón "Ejecutar".5. Comprobar que aparece como parámetro de salida un error que indica lo siguiente: "Parámetro obligatorio 'FicheroRed' no definido".

Tabla 211. Prueba PR-051



Identificador	PR-052
Descripción	Ejecutar módulo de comunicación sin el parámetro "Función".
Precondiciones	Simulador arrancado.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir en el cuadro de texto "Librería" el nombre <i>ModuloCom</i>.2. Introducir en el cuadro de texto "Clase a instanciar" el nombre <i>Comunicacion</i>.3. Introducir los siguientes parámetros de entrada:<ul style="list-style-type: none">▪ Tipo▪ NumCiclosMax▪ CriterioParada▪ RutaFicheros▪ FicheroRed4. Hacer clic sobre el botón "Ejecutar".5. Comprobar que aparece como parámetro de salida un error que indica lo siguiente: "Parámetro obligatorio 'Función' no definido".

Tabla 212. Prueba PR-052

Identificador	PR-053
Descripción	Ejecutar sin introducir ningún parámetro de entrada.
Precondiciones	Simulador arrancado.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir en el cuadro de texto "Librería" el nombre <i>ModuloCom</i>.2. Introducir en el cuadro de texto "Clase a instanciar" el nombre <i>Comunicacion</i>.3. Hacer clic sobre el botón "Ejecutar".4. Comprobar que aparece como parámetro de salida un error que indica lo siguiente: "Ningún parámetro obligatorio definido".

Tabla 213. Prueba PR-053



Identificador	PR-054
Descripción	Ejecutar módulo de comunicación con una función desconocida.
Precondiciones	Simulador arrancado.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir en el cuadro de texto “Librería” el nombre <i>ModuloCom</i>.2. Introducir en el cuadro de texto “Clase a instanciar” el nombre <i>Comunicacion</i>.3. Introducir como parámetro de entrada “Funcion” con valor “aaabbb”.4. Hacer clic sobre el botón “Ejecutar”.5. Comprobar que aparece como parámetro de salida un error que indica lo siguiente: “Valor del parámetro ‘Funcion’ debe ser TRAIN, TEST, PRODUCCION o TRAINTEST”.

Tabla 214. Prueba PR-054

Identificador	PR-055
Descripción	Ejecutar módulo de comunicación con un tipo desconocido.
Precondiciones	Simulador arrancado.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir en el cuadro de texto “Librería” el nombre <i>ModuloCom</i>.2. Introducir en el cuadro de texto “Clase a instanciar” el nombre <i>Comunicacion</i>.3. Introducir como parámetro de entrada “Tipo” con valor “aaabbb”.4. Hacer clic sobre el botón “Ejecutar”.5. Comprobar que aparece como parámetro de salida un error que indica lo siguiente: “Valor del parámetro ‘Funcion’ debe ser CLASIFICACION o REGRESION”.

Tabla 215. Prueba PR-055



Identificador	PR-056
Descripción	Ejecutar módulo de comunicación con un criterio de parada desconocido.
Precondiciones	Simulador arrancado.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir en el cuadro de texto “Librería” el nombre <i>ModuloCom</i>.2. Introducir en el cuadro de texto “Clase a instanciar” el nombre <i>Comunicacion</i>.3. Introducir como parámetro de entrada “Funcion” con valor “Train”.4. Introducir como parámetro de entrada “CriterioParada” con valor “aaabbb”.5. Hacer clic sobre el botón “Ejecutar”.6. Comprobar que aparece como parámetro de salida un error que indica lo siguiente: “Valor del parámetro ‘Funcion’ debe ser MSE, EPOCS o CROSSVALIDATION”.

Tabla 216. Prueba PR-056

Identificador	PR-057
Descripción	Ejecutar módulo de comunicación con un criterio de parada MSE y sin parámetro “UmbralMSE”.
Precondiciones	Simulador arrancado.
Pasos a seguir	<ol style="list-style-type: none">1. Introducir en el cuadro de texto “Librería” el nombre <i>ModuloCom</i>.2. Introducir en el cuadro de texto “Clase a instanciar” el nombre <i>Comunicacion</i>.3. Introducir como parámetro de entrada “Funcion” con valor “Train”.4. Introducir como parámetro de entrada “CriterioParada” con valor “MSE”.5. Hacer clic sobre el botón “Ejecutar”.6. Comprobar que aparece como parámetro de salida un error que indica lo siguiente: “Parámetro obligatorio para criterio de parada MSE ‘UmbralMSE’ no definido”.

Tabla 217. Prueba PR-057



7.4 Plantilla de informe de pruebas

Durante la realización de las pruebas del sistema se debe realizar un informe en el que se detallen los resultados obtenidos. A continuación se especifican los campos que deben ser incluidos en el informe y el formato establecido para este proyecto:

- **Identificador:** Identificador de la prueba realizada.
- **Fecha:** Fecha en la que se realizó la prueba.
- **Responsable:** Miembro del equipo del proyecto encargado de realizar la prueba.
- **Descripción:** Objetivo de la prueba realizada.
- **Resultado:** Este campo mostrará el valor *Aceptada* si el sistema ha realizado la prueba correctamente, o *Rechazada* en caso contrario.
- **Descripción de los fallos:** Si la prueba ha sido *Rechazada* este campo contendrá una descripción de los fallos encontrados durante la realización de dicha prueba.

Identificador	
Fecha	
Responsable	
Descripción	
Resultado	
Descripción de los fallos	

Tabla 218. Plantilla de informe de pruebas



7.5 Matriz de trazabilidad Pruebas/Requisitos de Capacidad

	RU-C-001	RU-C-002	RU-C-003	RU-C-004	RU-C-005	RU-C-006	RU-C-007	RU-C-008	RU-C-009	RU-C-010	RU-C-011	RU-C-012	RU-C-013	RU-C-014	RU-C-015	RU-C-016	RU-C-017	RU-C-018	RU-C-019	RU-C-020	RU-C-021	RU-C-022	
PR-001	X																						
PR-002		X																					
PR-003		X																					
PR-004			X																				
PR-005			X																				
PR-006				X																			
PR-007				X																			
PR-008				X																			
PR-009				X																			
PR-010				X																			
PR-011				X																			
PR-012				X																			
PR-013				X																			
PR-014				X																			
PR-015					X	X																	
PR-016						X																	
PR-017						X																	
PR-018						X																	
PR-019						X																	
PR-020						X																	



PR-021					X															
PR-022					X															
PR-023						X														
PR-024							X													
PR-025							X													
PR-026							X													
PR-027								X												
PR-028								X												
PR-029								X												
PR-030								X												
PR-031								X												
PR-032									X											
PR-033									X											
PR-034									X											
PR-035									X											
PR-036										X										
PR-037											X									
PR-038													X							
PR-039														X						
PR-040											X	X			X					
PR-041															X					
PR-042															X					
PR-043																X				
PR-																	X	X		X



8 Implementación

Durante la fase de implementación se ha generado el código necesario para el funcionamiento de los distintos módulos del sistema desarrollados en el proyecto.

En este apartado se describe el funcionamiento de los módulos implementados y la interacción existente entre los mismos.

8.1 Aplicación para tratamiento de ficheros

La aplicación desarrollada para el tratamiento de ficheros no forma parte del sistema híbrido como tal. Sin embargo, esta aplicación complementa el funcionamiento del sistema híbrido, ya que los ficheros que genera a partir de los datos de entrada son los empleados para el entrenamiento, validación y producción de las redes de neuronas artificiales.

Mediante una única interfaz la aplicación permite configurar todas las opciones de los experimentos que se quieren preparar, así como importar los datos del fichero de entrada en formato de Microsoft Excel. A continuación se muestra la interfaz de la aplicación:

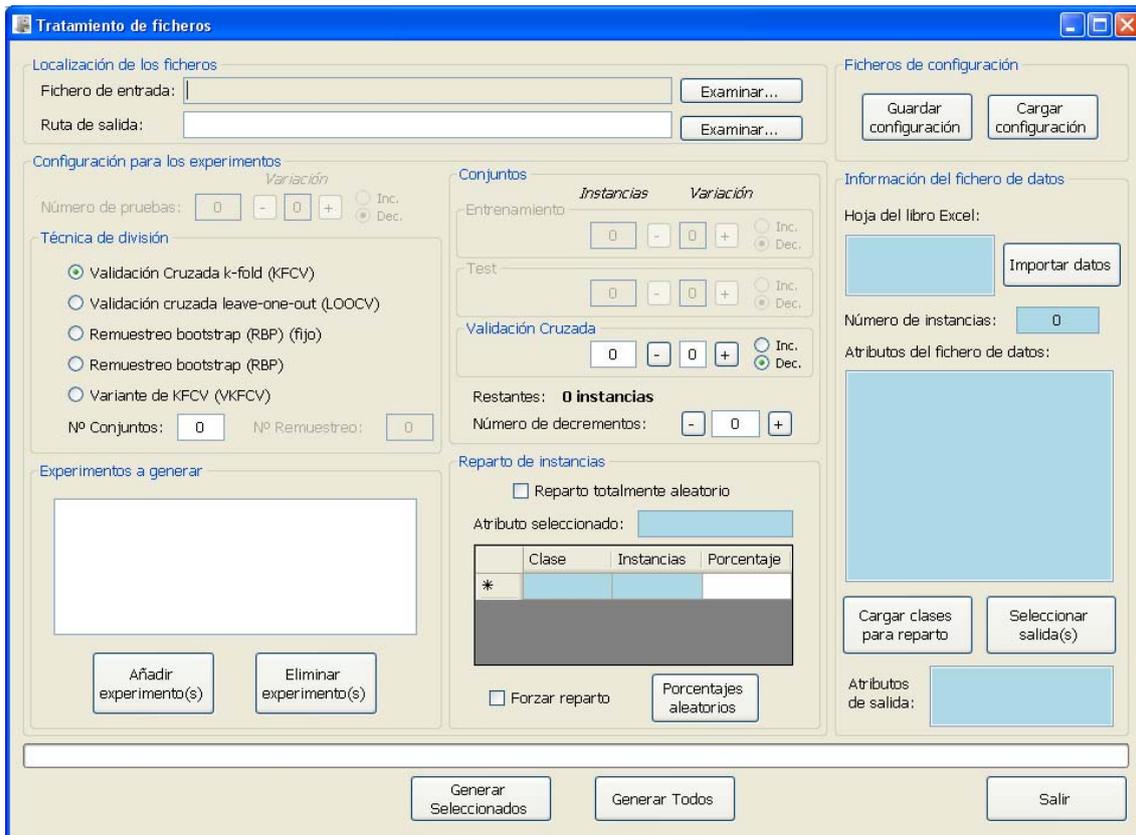


Figura 37. Interfaz Aplicación tratamiento de ficheros

El funcionamiento básico de la aplicación consiste en acceder a un fichero de datos en formato Excel (ver Anexo B: Formato para ficheros de entrada), y preparar los experimentos en función de la configuración seleccionada por el usuario. La preparación de los experimentos se realiza generando los ficheros correspondientes de entrenamiento, validación y test en el formato adecuado para las redes de neuronas artificiales. Estos ficheros deben ser generados de acuerdo a las cantidades de instancias y atributos seleccionados por el usuario. Asimismo, el reparto de las instancias disponibles se realiza en función de la técnica de división elegida (ver 12.4.1. Técnica de división).



8.2 Módulo de comunicación

El módulo de comunicación implementado es el módulo principal del proyecto, ya que es el encargado de la interacción entre redes de neuronas artificiales y sistema experto para la construcción de un sistema híbrido.

Se trata de una librería implementada sin interfaz gráfica, que debe ser utilizada desde el sistema experto o bien desde el simulador implementado para las pruebas de este proyecto.

Una vez instanciado, el módulo de comunicación recoge unos parámetros de entrada del sistema experto, interacciona con las redes de neuronas artificiales y devuelve como parámetros de salida los informes de evaluación generados.

Para la comunicación con el sistema experto el módulo de comunicación debe cumplir con un interfaz de comunicación que consiste en utilizar dos objetos del tipo de usuario *CParametro* (implementado previamente), en el cual se almacenan tanto los parámetros de entrada como los parámetros de salida introducidos.

Una vez obtenidos los parámetros de entrada desde el sistema experto, el módulo de comunicación interacciona con las redes de neuronas artificiales, en este caso *NeuroSolutions*, mediante la interfaz de comunicación que proporciona, permitiendo utilizar una red de neuronas creada previamente para el proceso que desee el usuario: validación, entrenamiento o producción.

Cuando el proceso con las redes de neuronas finaliza, el propio módulo de comunicación se encarga de la generación de un informe con los resultados. Esto se ha implementado para que realice el informe tanto en formato de Microsoft Excel, como en formato XML, ya que el proyecto en el futuro está orientado a funcionar en una plataforma Web.

Las rutas de estos informes generados se devuelven al Sistema Experto como parámetros de salida poniéndolos a su disposición.

Por otro lado, ha sido necesario implementar un control de errores para los parámetros introducidos, de manera que el propio módulo de comunicación sea capaz de indicar los parámetros necesarios y los posibles valores para cada uno de ellos.

8.3 Simulador

El simulador implementado es una aplicación que permite, mediante una interfaz sencilla, interactuar con el módulo de comunicación descrito en el apartado anterior de la misma forma que lo haría el sistema experto.

La interfaz de la aplicación del simulador se ha desarrollado de manera que permita un funcionamiento lo más ágil y sencillo posible. Los componentes principales con los que cuenta son dos tablas utilizadas para los parámetros de entrada y de salida del sistema, un cuadro de texto en el que indicar el módulo de comunicación a utilizar, un botón de ejecución, y una pantalla de estado. A continuación se pueden observar los elementos enumerados:

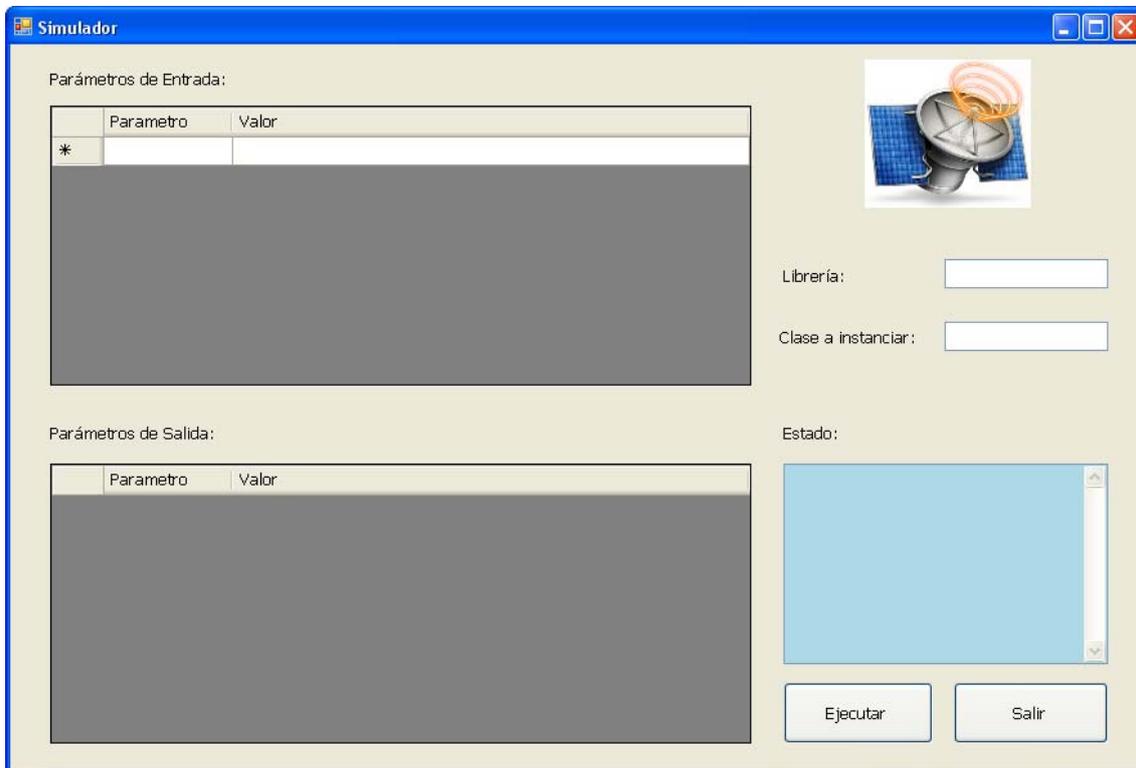


Figura 38. Interfaz Simulador



El objetivo del simulador es sustituir las funciones del sistema experto dentro del sistema híbrido, sobre todo a fin de agilizar el proceso de pruebas con el resto de componentes del sistema. Para ello, se ha implementado de manera que utilice la misma interfaz de comunicación descrita en el apartado anterior, introduciendo los parámetros deseados en la tabla de parámetros de entrada, comunicándoselos al módulo de comunicación, y por último, extrayendo los parámetros de salida recibidos. La selección de parámetros de entrada que en el simulador la realiza el usuario es lo que en el sistema híbrido final debe realizar el sistema experto.

Además, la implementación del simulador se ha realizado de manera que permita la instanciación de otros módulos de comunicación distintos al desarrollado en el presente proyecto, siempre y cuando respeten el interfaz de comunicación explicado anteriormente.

8.4 Interacción entre módulos

El siguiente diagrama proporciona una visión más intuitiva del funcionamiento de los módulos desarrollados y de la interacción existente entre ellos:

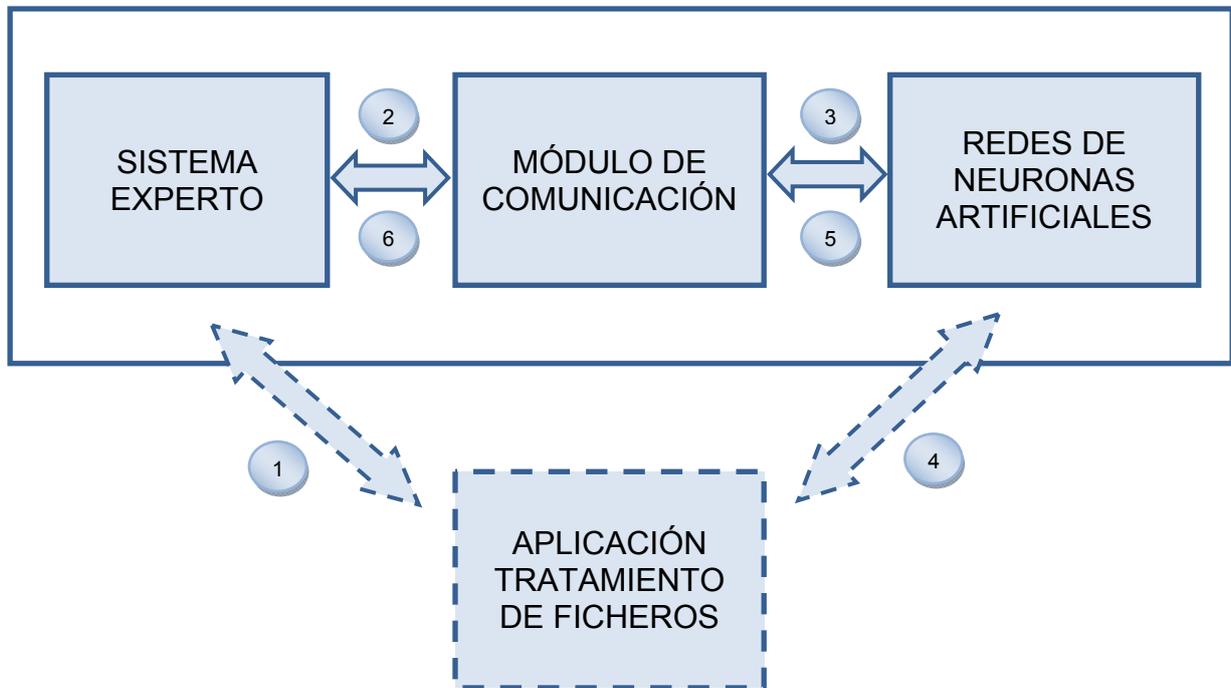


Figura 39. Interacción entre módulos del sistema

A continuación se describen las interacciones numeradas en el diagrama:

1. Esta no es una interacción directa entre estos módulos. El sistema experto hace uso de los archivos de experimentación generados por la aplicación de tratamiento de ficheros.
2. El sistema experto comunica al módulo de comunicación la red de neuronas elegida, la función a realizar (entrenamiento, validación o producción), los parámetros de configuración, y la ruta de los archivos generados por la aplicación externa.
3. El módulo de comunicación comunica a la aplicación de simulación de redes de neuronas artificiales la información obtenida del sistema experto acerca de la red neuronal, parámetros y archivos a utilizar.



4. Esta tampoco es una interacción directa entre estos módulos. La aplicación de redes de neuronas artificiales emplea los archivos generados por la aplicación de tratamiento de ficheros tal y como se lo comunica el módulo de comunicación.
5. La aplicación de redes de neuronas realiza la función ordenada por el sistema experto y, una vez finalizada, comunica al módulo de comunicación sus resultados.
6. Cuando el módulo de comunicación ha recibido los resultados de la aplicación de redes de neuronas artificiales y ha generado los correspondientes informes de evaluación en formato Excel y en XML, comunica al sistema experto la ruta de ambos archivos. En caso de que los parámetros enviados por el sistema experto (interacción 2) fueran erróneos, en esta interacción se enviarían los mensajes de error correspondientes, no produciéndose de las interacciones 3, 4, 5 y 6.

9 Pruebas de dominio

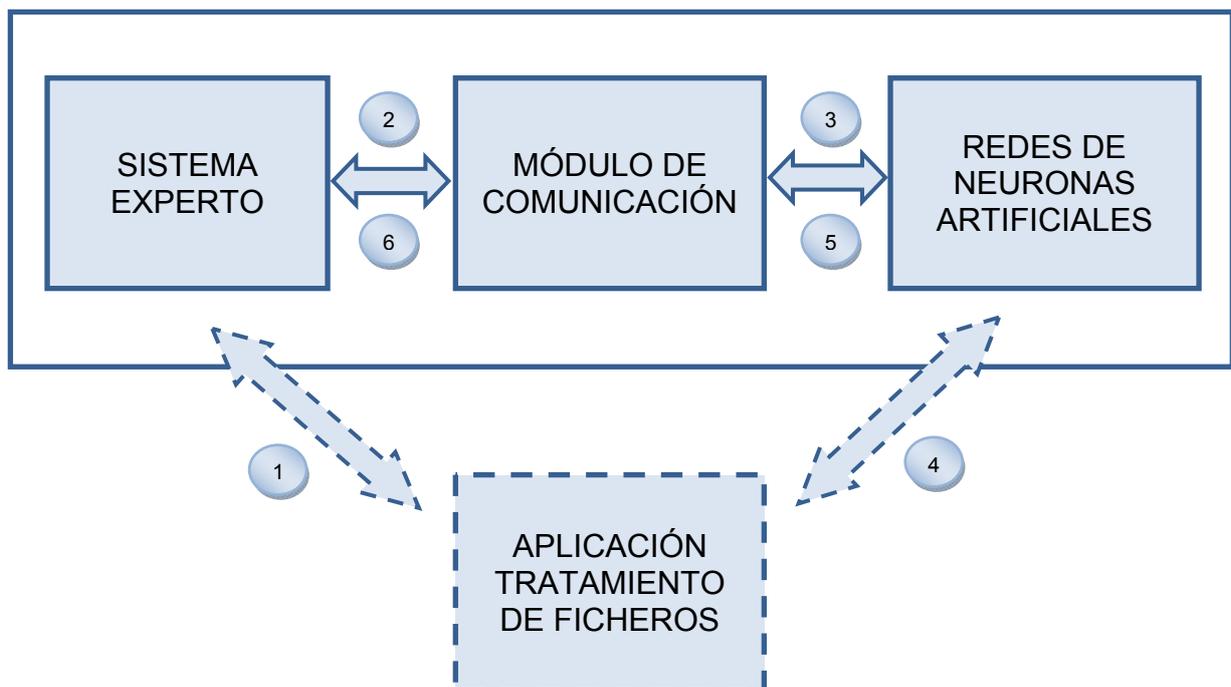
En este capítulo se presentan varias pruebas realizadas con el sistema desarrollado, mostrando así el funcionamiento del mismo. Para la realización de estas pruebas se han utilizado dos dominios de aplicación distintos: el impacto balístico y la estimación de esfuerzo en desarrollo de proyectos software. En los siguientes apartados se detallan en más profundidad tanto los dominios utilizados como el funcionamiento del sistema.

Como se podrá observar, las pruebas se han dividido en Clasificación y Regresión, los dos tipos de predicción que permite el sistema.

9.1 Prueba de dominio A: Clasificación

Para la prueba de Clasificación se ha utilizado el dominio del impacto balístico. Se cuenta con una serie de datos acerca de un proyectil, una placa de un material determinado y la información acerca del impacto del proyectil sobre dicha placa. El propósito de la prueba es que el sistema híbrido sea capaz de determinar, dados los datos de entrada, si el proyectil atraviesa o no la placa. La variable a predecir por el sistema será la variable '*Penetración*', que puede tomar los valores '*sí*' y '*no*'.

A continuación se muestra el diagrama de funcionamiento del sistema, a partir del cual se detallará el proceso realizado:





A continuación se describen las interacciones indicadas numéricamente en el diagrama anterior para facilitar la comprensión del proceso de ejecución de la prueba:

1. Esta interacción no representa una interacción directa entre los módulos del sistema híbrido, sino que representa la utilización por parte del sistema de los archivos generados por la aplicación de tratamiento de ficheros. En este caso se necesita únicamente un fichero con una instancia, ya que la prueba se realiza con la función de *Producción*, sobre una red ya entrenada y validada previamente. El contenido del fichero es el siguiente:

ATRIBUTO	VALOR
GROSOR_PLACA_(mm)	14
RADIO_PROY_(mm)	8
LONG_PROY_(mm)	25
VEL_(m/s)	713
MASA_INICIAL_(g)	94
MATERIAL_PROY	3
MATERIAL_PLACA	2
Masa_Final_(g)	89
$(M_i - M_f) / M_i$ (%)	5.3
(*)Penetración	si

Tabla 220. Instancia de prueba de Clasificación

(NOTA: El atributo 'Penetración' se muestra para conocer el resultado real de la instancia, pero el sistema únicamente recibe la información del resto de atributos para realizar su predicción).

2. El sistema experto (en este caso el simulador desarrollado) comunica al módulo de comunicación los parámetros necesarios para la ejecución. A continuación se muestran los parámetros enviados a través de la tabla de Parámetros de entrada del simulador:

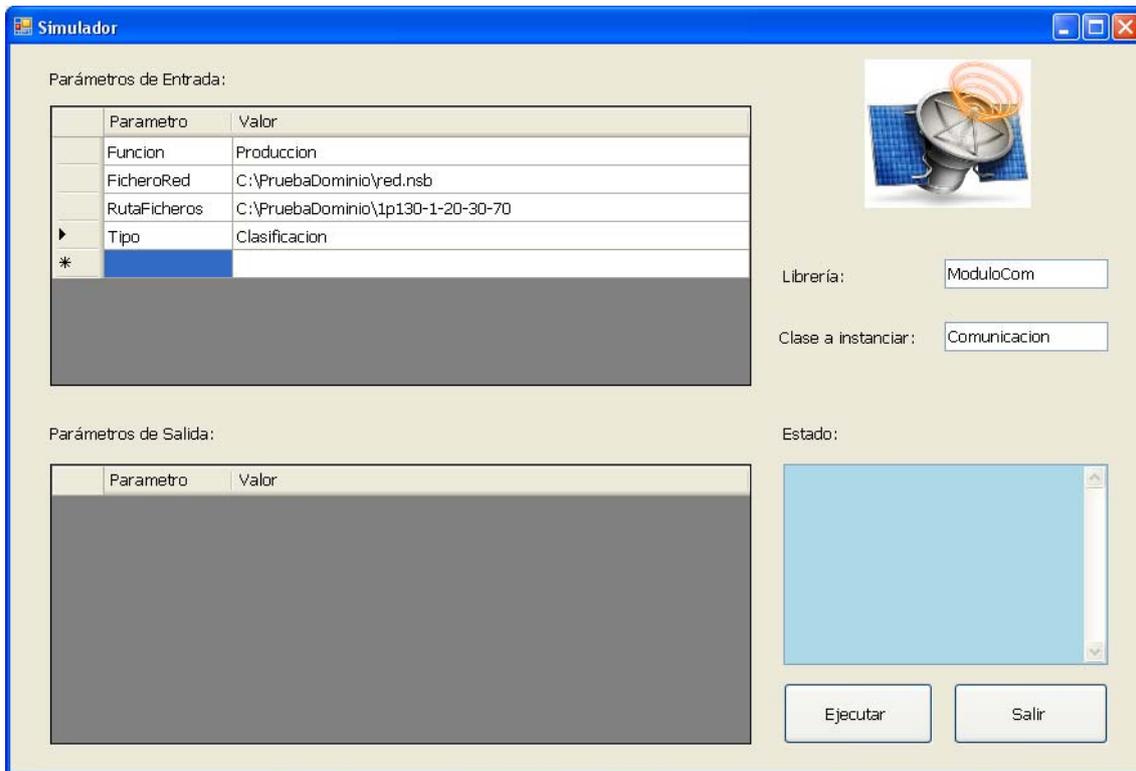


Figura 40. Parámetros de entrada del Sistema Experto

3. El módulo de comunicación comunica a la aplicación de simulación de redes de neuronas artificiales la información recibida del sistema experto, es decir, los parámetros de entrada anteriormente expuestos.
4. En este caso, la interacción tampoco representa una interacción directa entre los módulos del sistema híbrido. Representa la utilización del fichero detallado en el punto anterior por parte de la aplicación de redes de neuronas artificiales.

5. La aplicación de redes de neuronas realiza la función ordenada por el sistema experto a través del módulo de comunicación, tal y como se observa a continuación:

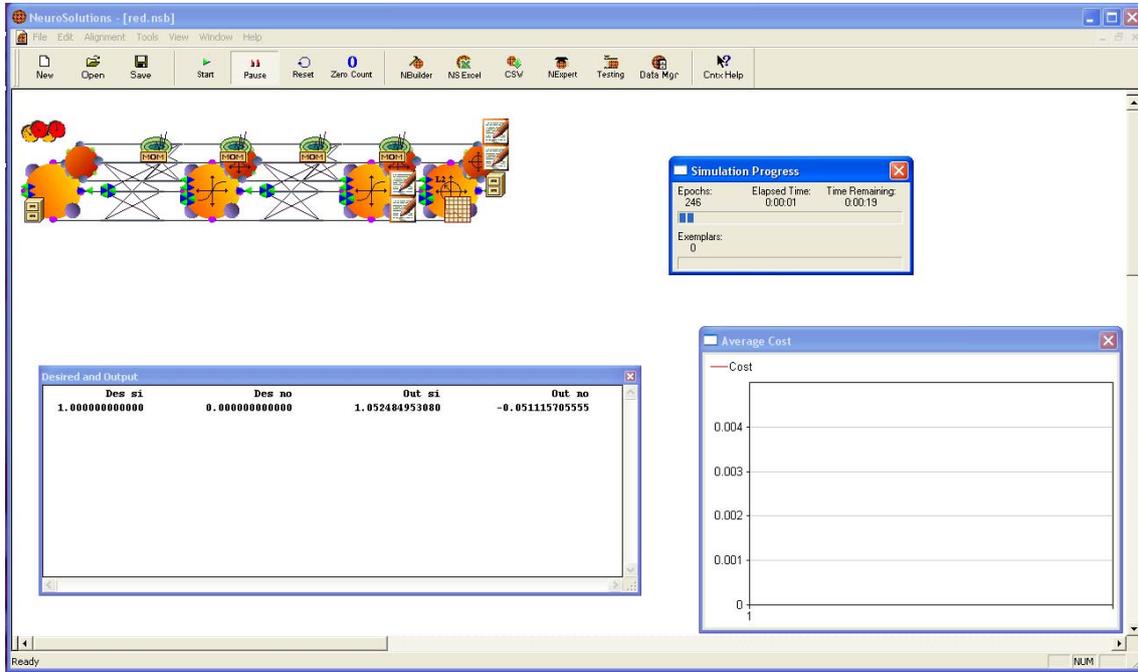


Figura 41. Ejecución en proceso

6. Una vez finalizado el proceso en la aplicación de redes de neuronas, ésta comunica la respuesta al módulo de comunicación, y éste a su vez al sistema experto a través de los parámetros de salida. En este caso, tal y como se observa a continuación, la predicción del sistema híbrido para el ensayo introducido es el valor 'si' para la variable 'Penetración'.

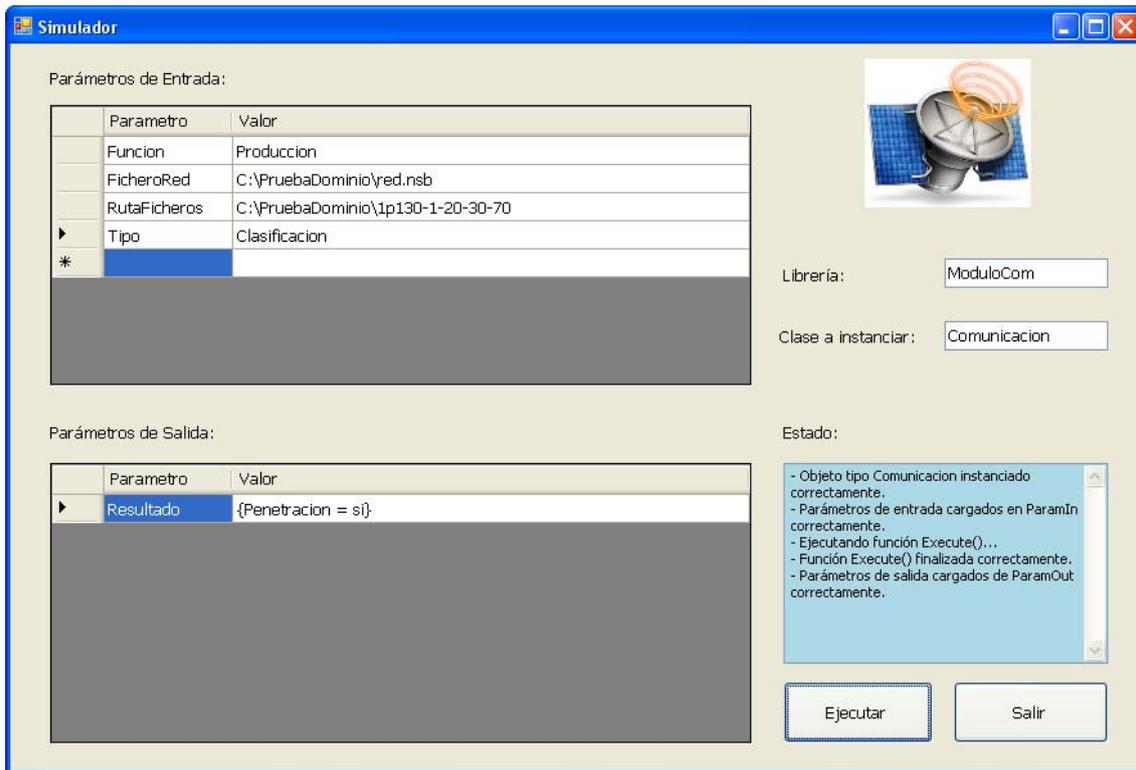


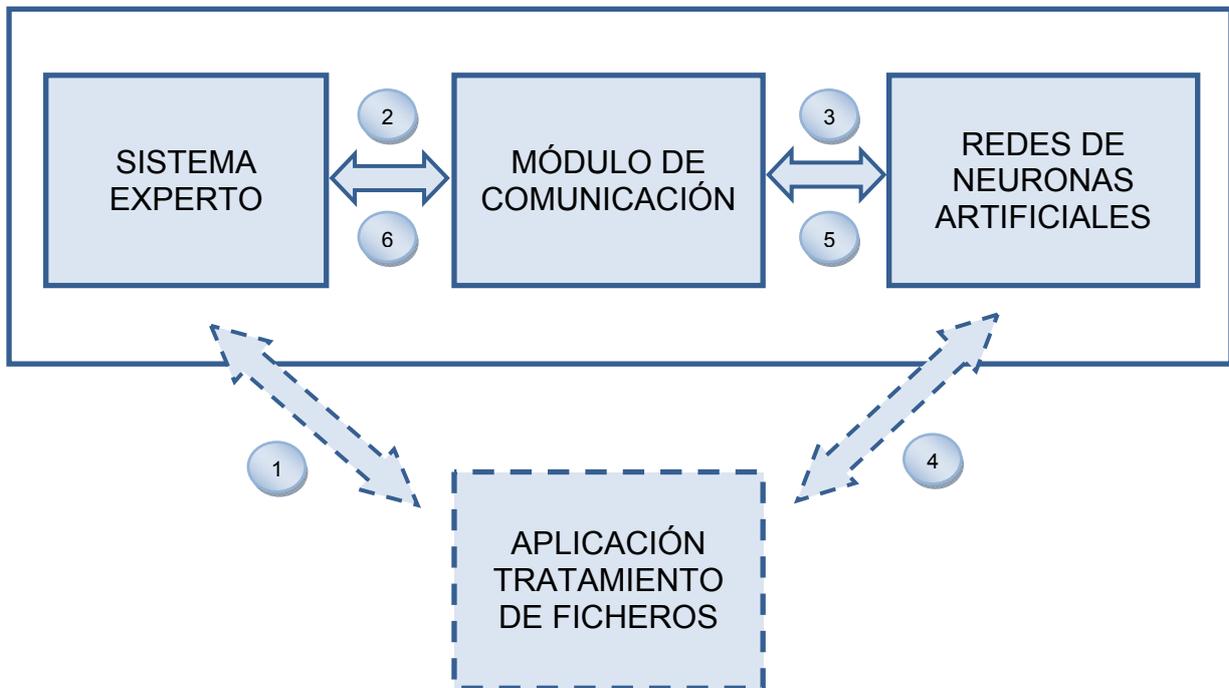
Figura 42. Resultado de la ejecución

Si se comparan la predicción proporcionada por el sistema híbrido con el valor real expuesto en el punto 1, se puede comprobar cómo la predicción efectuada es correcta.

9.2 Prueba de dominio B: Regresión

Para la prueba de Regresión el dominio utilizado ha sido el de la estimación de esfuerzo en el desarrollo de proyectos software. Se cuenta con información acerca de una amplia cantidad de proyectos, incluyendo datos como el lenguaje empleado en su desarrollo, plataforma de desarrollo, tipo de aplicación, cantidad de personas en el equipo de trabajo, y otras características acerca del sistema a desarrollar. Además de estas características del proyecto, se incluye como medida de estimación los puntos de función y la cantidad de esfuerzo necesario en meses/hombre. Son estas dos variables numéricas de estimación de esfuerzo las que se pretende predecir mediante el sistema híbrido desarrollado.

A continuación se muestra el diagrama de funcionamiento del sistema, a partir del cual se detallará el proceso realizado:



A continuación se describen las interacciones indicadas numéricamente en el diagrama anterior para facilitar la comprensión del proceso de ejecución de la prueba:

1. Esta interacción no representa una interacción directa entre los módulos del sistema híbrido, sino que representa la utilización por parte del sistema de los archivos generados por la aplicación de tratamiento de ficheros. En este caso se necesita únicamente un fichero con una instancia, ya que la prueba se realiza con la función de *Producción*,



sobre una red ya entrenada y validada previamente. El contenido del fichero es el siguiente:

ATRIBUTO	VALOR
(*)Summary_Work_Effort	3377
(*)Function_Points	779
Development_Platform	3
Development_Type	2
Business_Area_Type	8
Application_Type	1
Max_Team_Size	3
Language_Type	2
Primary_Programming_Language	6
User_Base_-_Business_Units	0
User_Base_-_Locations	0
User_Base_-_Concurrent_Users	0
Used_CASE	0
Used_Methodology	0
How_Methodology_Aquired	0

Tabla 221. Instancia de prueba de Regresión

(NOTA: Los atributos 'Summary_Work_Effort' y 'Function_Points' se muestran para conocer el resultado real de la instancia, pero el sistema únicamente recibe la información del resto de atributos para realizar su predicción).

2. El sistema experto (en este caso el simulador desarrollado) comunica al módulo de comunicación los parámetros necesarios para la ejecución. A continuación se muestran los parámetros enviados a través de la tabla de Parámetros de entrada del simulador:

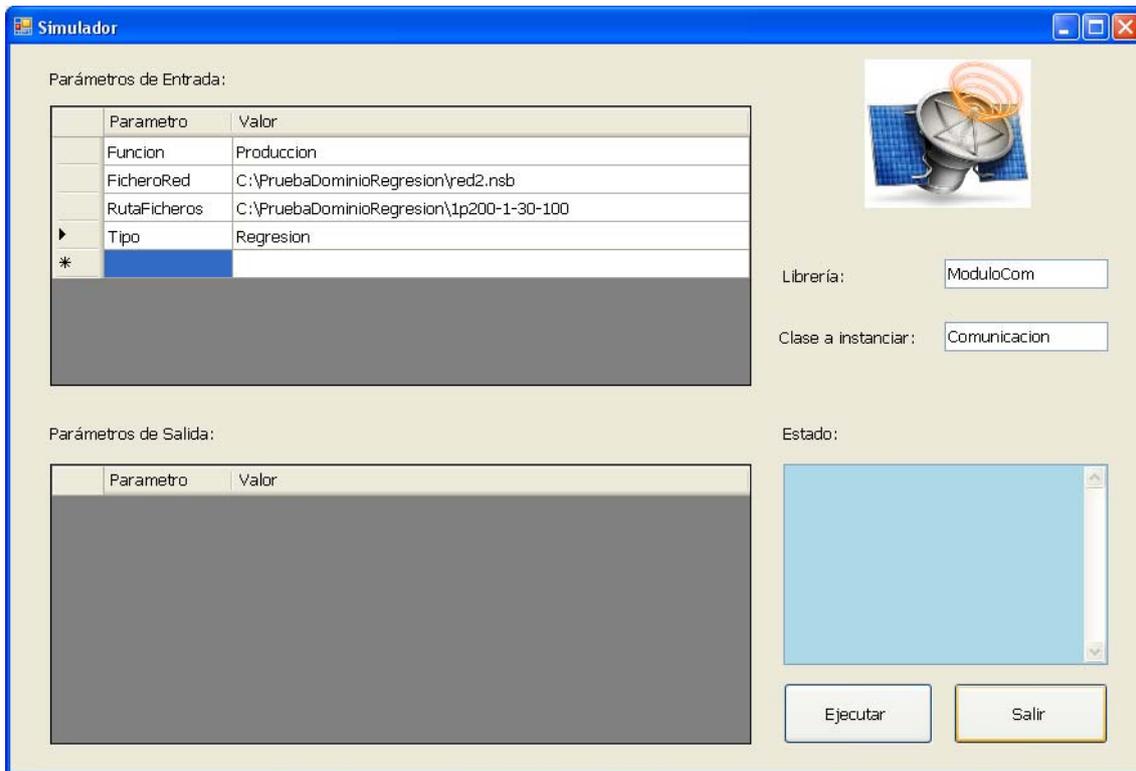


Figura 43. Parámetros de entrada del Sistema Experto

3. El módulo de comunicación comunica a la aplicación de simulación de redes de neuronas artificiales la información recibida del sistema experto, es decir, los parámetros de entrada anteriormente expuestos.
4. En este caso, la interacción tampoco representa una interacción directa entre los módulos del sistema híbrido. Representa la utilización del fichero detallado en el punto anterior por parte de la aplicación de redes de neuronas artificiales.

5. La aplicación de redes de neuronas realiza la función ordenada por el sistema experto a través del módulo de comunicación, tal y como se observa a continuación:

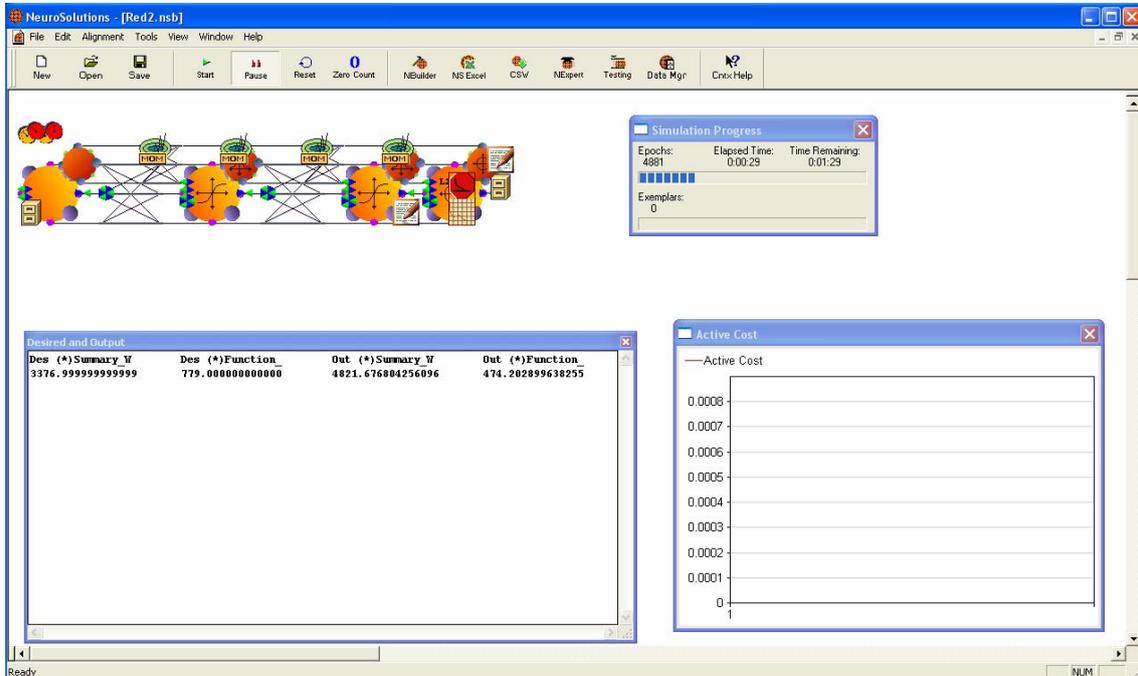


Figura 44. Ejecución en proceso

6. Una vez finalizado el proceso en la aplicación de redes de neuronas, ésta comunica la respuesta al módulo de comunicación, y éste a su vez al sistema experto a través de los parámetros de salida. En este caso, tal y como se observa a continuación, la predicción del sistema híbrido para el ensayo introducido es el valor '4821,67' para la variable 'Summary_Work' y '474,20' para la variable 'Function_Points'.

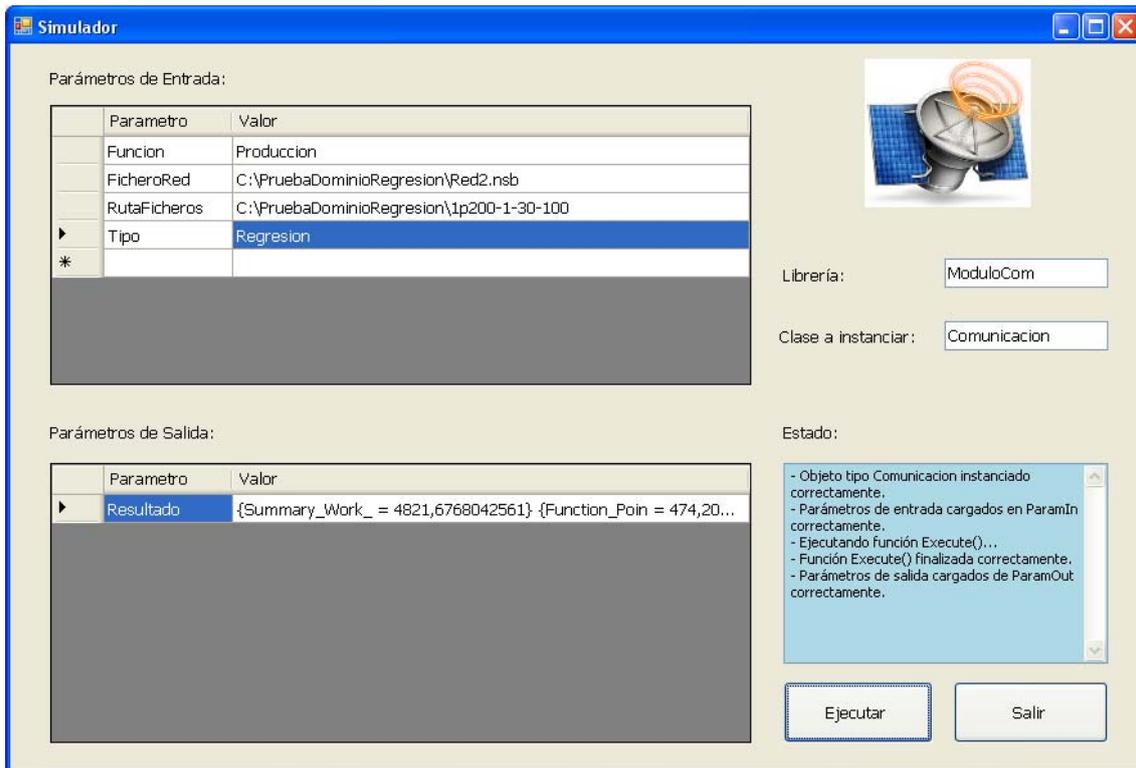


Figura 45. Resultado de la ejecución

Si se comparan la predicción proporcionada por el sistema híbrido con el valor real expuesto en el punto 1, se puede comprobar cómo la predicción efectuada no es del todo precisa, debido probablemente a los datos destinados al entrenamiento previo de la red, o a tratarse de una instancia con muchos atributos con valores desconocidos (representados por un cero).



10 Conclusiones

Con la realización de este proyecto se pretendía alcanzar una serie de objetivos marcados al inicio del mismo, tarea que se ha conseguido satisfactoriamente.

En primer lugar, se ha creado un sistema que proporciona la funcionalidad deseada, es decir, la interconexión de dos técnicas de inteligencia artificial como son las redes de neuronas artificiales y los sistemas expertos para la producción de sistemas híbridos.

Se ha llevado a cabo el proyecto siguiendo una adaptación de la metodología *Métrica Versión 3*, permitiendo desarrollar tanto el sistema en sí, como su documentación con un nivel de calidad suficiente.

Se ha seguido la planificación establecida al principio del proyecto, salvo en ligeras modificaciones, cumpliendo los plazos de entrega establecidos al comienzo del proyecto.

Se han puesto en práctica los conocimientos adquiridos a lo largo de la carrera, sobre todo en cuanto al uso de metodologías, programación orientada a objetos, técnicas de inteligencia artificial, y gestión de proyectos software. Además, se han adquirido nuevos conocimientos gracias al aprendizaje de la tecnología .NET utilizada en el desarrollo del sistema.

Se puede concluir finalmente, que se ha logrado alcanzar el objetivo de desarrollar un proyecto software completo, adquiriendo nuevos conocimientos acerca de los roles a desarrollar en este tipo de proyectos, y comprobando la dificultad existente en cuanto a esfuerzo, dedicación y coste de desarrollar un proyecto de forma correcta.



11 Bibliografía

- [1] Benachenhou, D. / Cader, M. / Szu, H. / Medsker, L. / Wittwer, C. / Garling, D., *“Neural networks for computing invariant clustering of a large open set of DNA-PCR primers generated by a feature-knowledge based system.”*, Proc. IJCNN-90, San Diego, CA, Vol. II, 1990.
- [2] European Space Agency, *“Guide to software verification and validation”*, ESA Board for Software Standardisation and Control. 1995.
- [3] Hanson, M. A. / Brekke, R. L., *“Workload management expert system – combining neural network and expert system programming in an operational application”*, Proc. Instrument Society of America, 1988.
- [4] Hendler, J., *“Developing hybrid symbolic/connectionist models”*, Barnden, 1991.
- [5] IEEE, *“IEEE Standard for Developing Software Life Cycle Processes”*, IEEE Std. 1074-1997, 1997.
- [6] Labate, F. / Medsker, L., *“Employee skills analysis using a hybrid neural network and expert system”*, Proceedings of the IEEE International Conference on Developing and Managing Intelligent System Projects, Washington, DC, 1993.
- [7] Martín Muñoz, David, *“Estudio de la viabilidad de aplicación de las redes de neuronas artificiales para la resolución del problema del impacto sobre materiales”*, Proyecto Fin de Carrera, Universidad Carlos III, 2007.
- [8] Medsker, L., *“Design and Development of Hybrid Neural Network and Expert Systems”*, Department of Computer Science and Information Systems, The American University, Washington, DC, 1994.
- [9] Medsker, L.R., *“Hybrid Neural Network and Expert Systems”*, Kluwer Academic Publishers, Boston, 1994.
- [10] Medsker, L. / Turban, E., *“Integrating expert systems and neural computing for decision support”*, Expert Systems with Applications, vol. 7, 1994.
- [11] Ministerio de Administraciones Públicas, *“Métrica Versión 3”*, <http://www.csi.map.es/csi/metrica3/>.



- [12] Schreinemakers, J. F. / Touretzky, D. S., *“Interfacing a neural network with a rule-based reasoner for diagnosing mastitis”*, Proc. IJCNN II, Washington, DC, 1990.
- [13] Wilson, A. / Hendler, J., *“Linking symbolic and subsymbolic computing”*, Technical Research Report TR 93-12, Institute for Systems Research, University of Maryland, 1993.

12 Anexo A: Manual de usuario Aplicación Tratamiento de Ficheros

A continuación se proporciona una guía para la utilización de las distintas funcionalidades disponibles en la aplicación. Esta guía describe al usuario el funcionamiento básico de la aplicación.

12.1 Arranque de la aplicación

Para iniciar la aplicación basta con hacer doble clic sobre el icono de la aplicación, tal y como muestra la imagen:

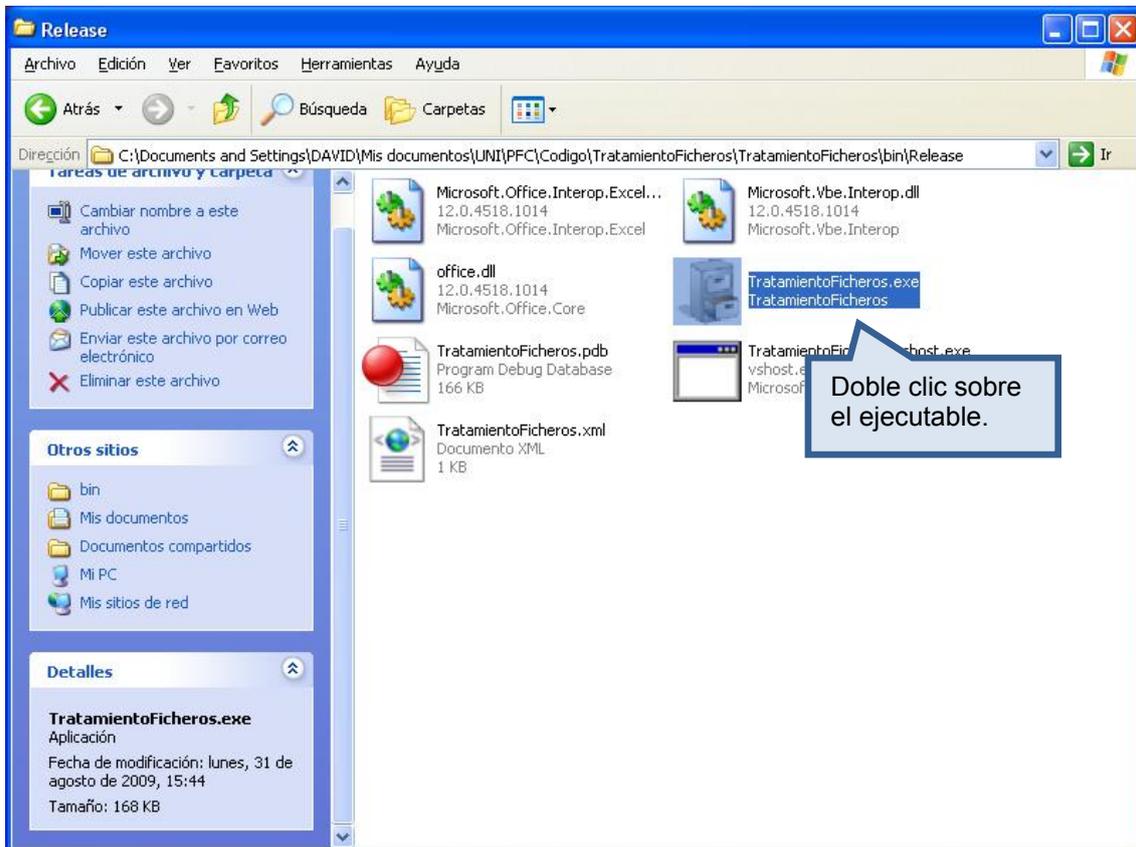


Figura 46. Arranque de la aplicación

Una vez iniciada la aplicación el usuario podrá visualizar el interfaz de la aplicación. A continuación se muestra la aplicación recién arrancada:

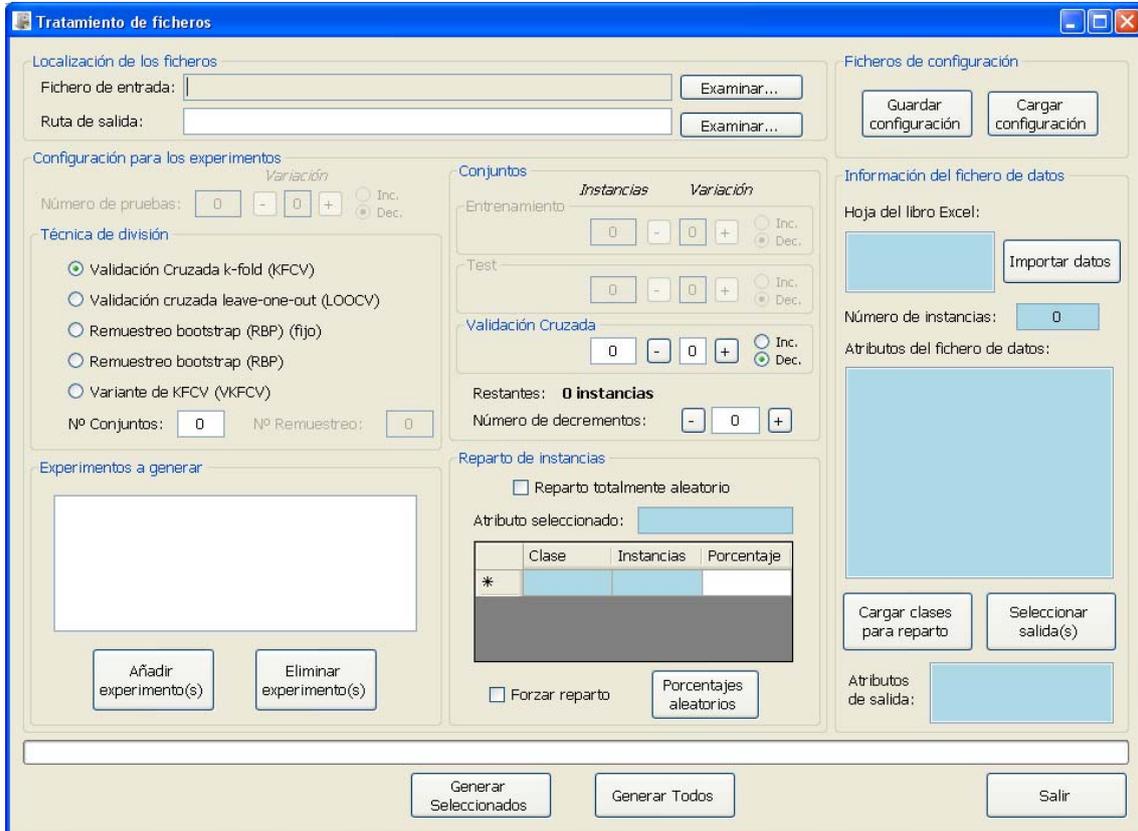


Figura 47. Interfaz inicial de la aplicación

12.2 Localización de ficheros

Una vez arrancada la aplicación, el primer paso es especificar la localización de los ficheros de datos. Esta operación debe incluir tanto el fichero de entrada original, como la ruta de salida para los ficheros generados por la aplicación.

12.2.1 Fichero de entrada

Para especificar el fichero de entrada, se debe seleccionar el fichero mediante el ratón utilizando el botón “Examinar...” situado junto al cuadro de texto “Fichero de entrada”, a través del cuadro de diálogo que mostrará la aplicación. La siguiente ilustración muestra el proceso:

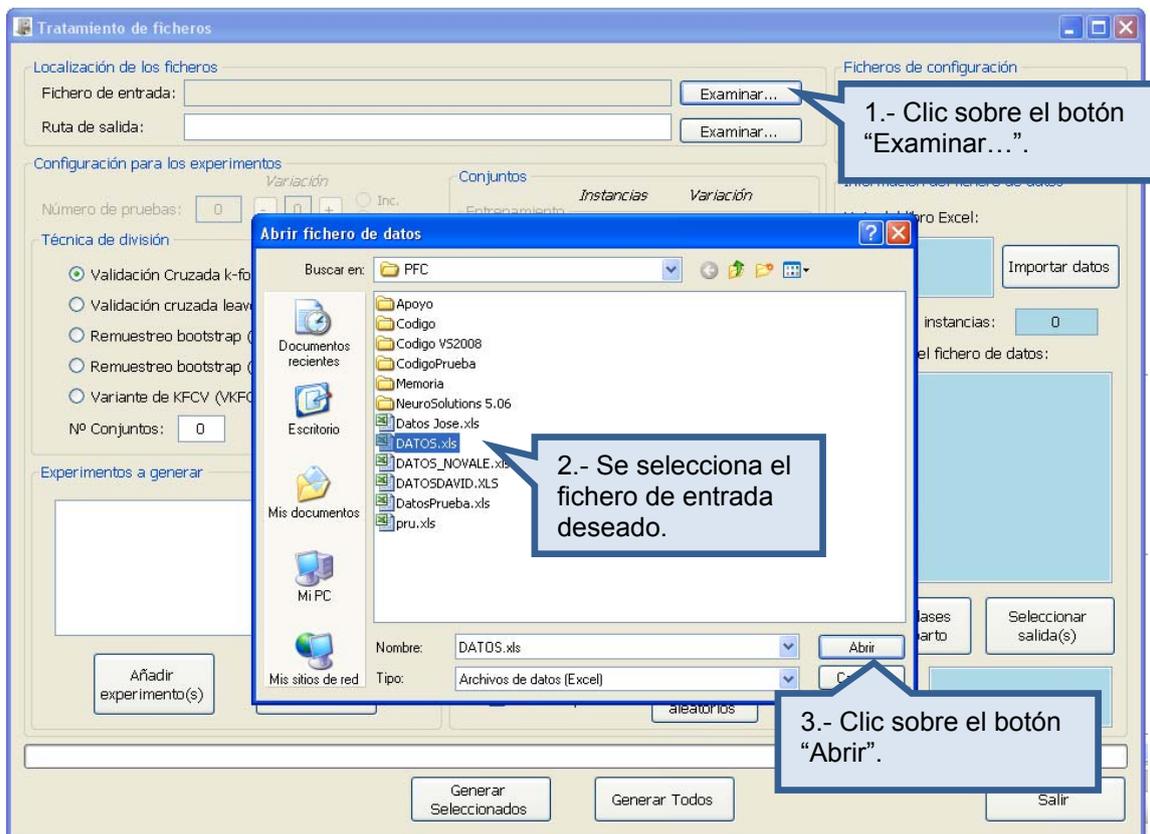


Figura 48. Seleccionando un fichero de entrada

NOTA: En cualquier caso, el fichero seleccionado debe cumplir con los requisitos de formato establecidos por la aplicación (véase Anexo B: Formato para ficheros de entrada).

12.2.2 Ruta de salida

Es necesario especificar una ruta de salida donde la aplicación almacenará los ficheros de entrenamiento, test y validación cruzada generados durante el proceso. Para ello, el usuario puede tomar dos opciones:

Puede especificar directamente la ruta en el cuadro de texto “Ruta de salida”, tal y como muestra la siguiente ilustración:

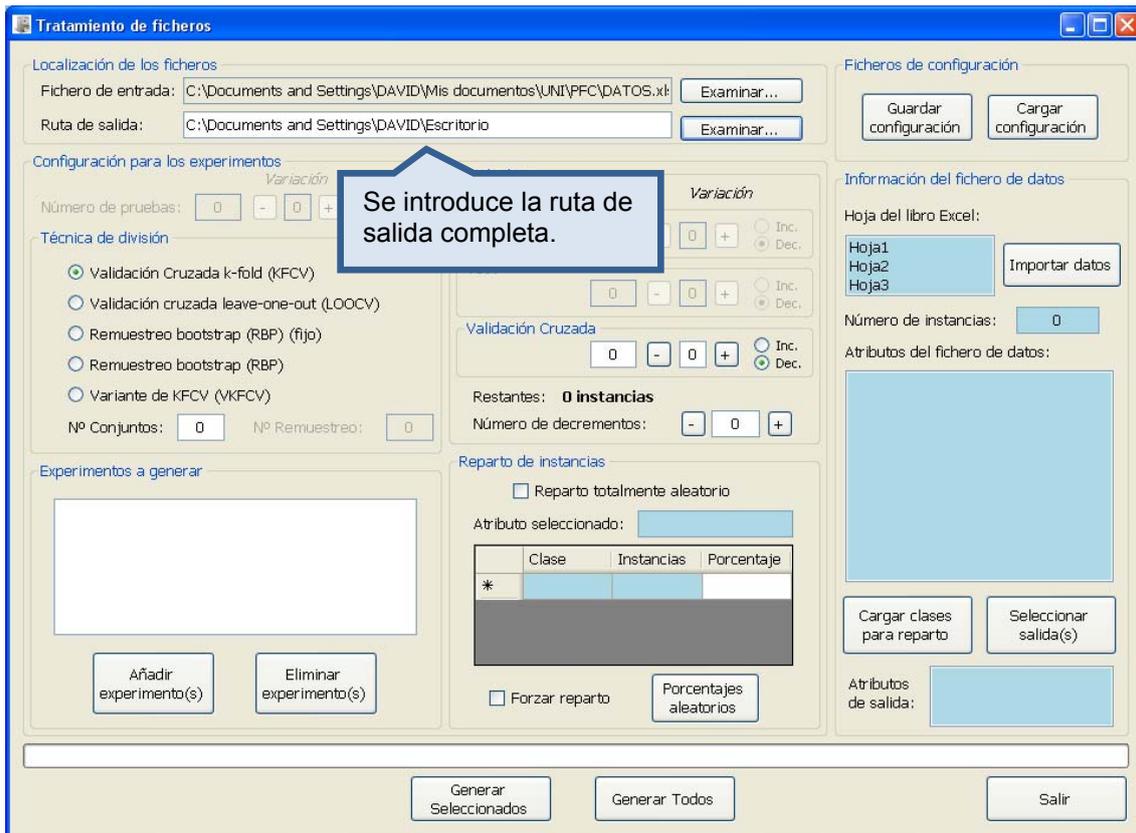


Figura 49. Introduciendo manualmente una ruta de salida

En caso de no conocer la ruta completa, o bien por mayor comodidad, puede seleccionar la ruta de salida a través del cuadro de diálogo que mostrará la aplicación al hacer clic sobre el botón “Examinar...” situado junto al cuadro de texto “Ruta de salida”.

La siguiente figura muestra el cuadro diálogo para seleccionar la ruta de salida:

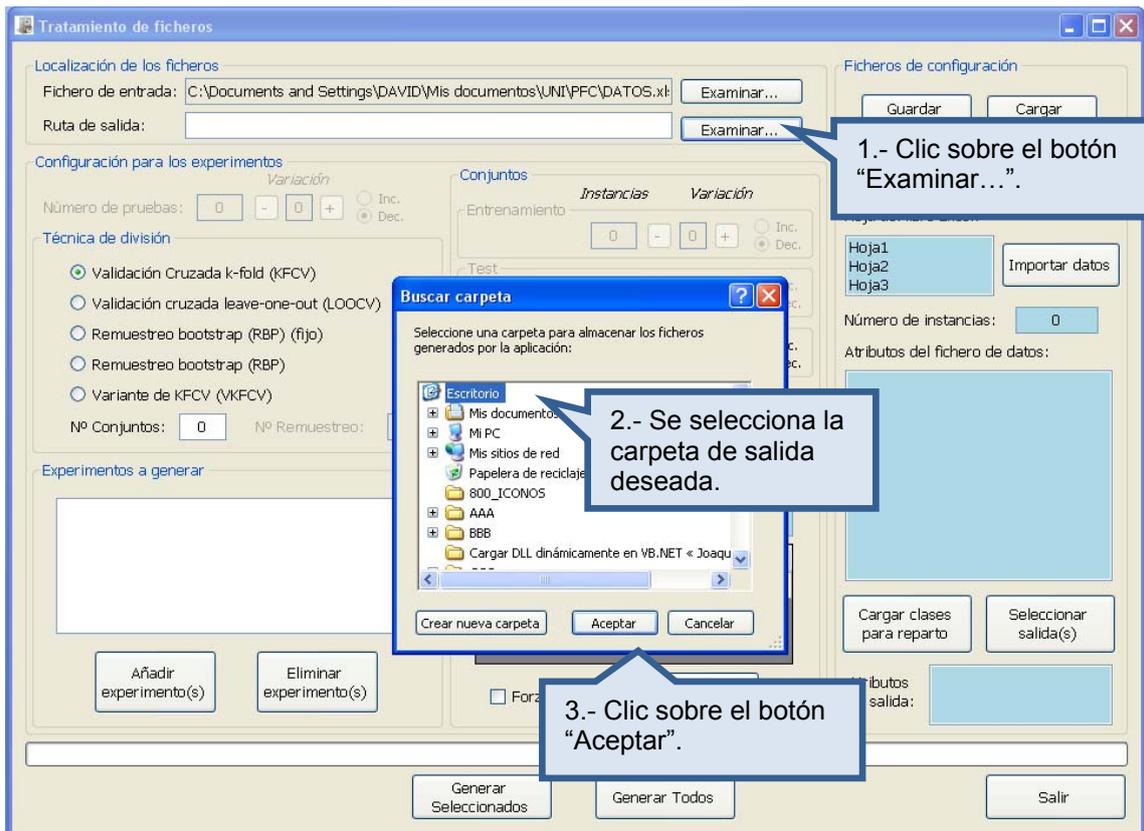


Figura 50. Seleccionando una ruta de salida

12.3 Importar datos

Cuando se ha seleccionado un fichero de entrada (con el formato adecuado) la aplicación permite importar la información necesaria a la aplicación. Para ello se debe especificar la hoja del libro Microsoft Excel donde se encuentran los datos y a continuación hacer clic sobre el botón “Importar datos”, tal y como se muestra a continuación:

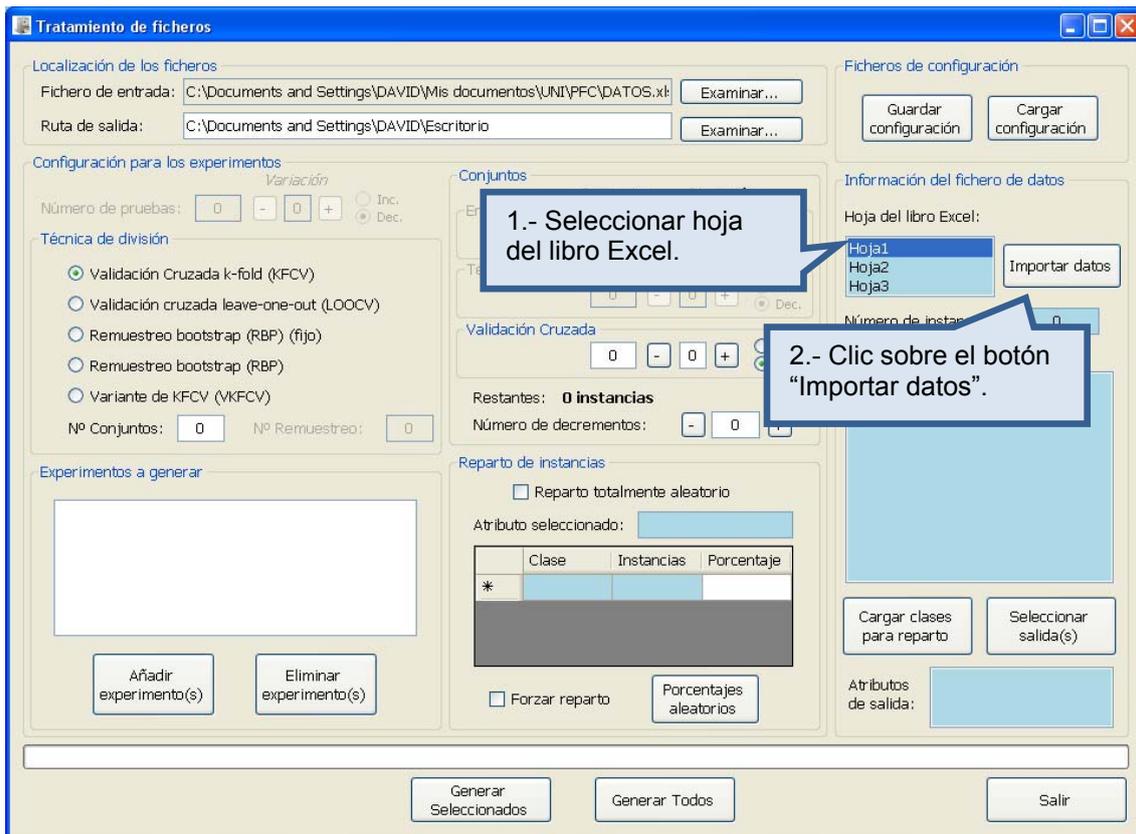


Figura 51. Importar datos

Al realizar esta operación se cargará el número de instancias encontradas en el cuadro de texto “Número de instancias” y la aplicación mostrará un cuadro de diálogo para la selección de atributos. Este proceso se detalla en el siguiente apartado.

12.3.1 Selección de atributos

Tras hacer clic sobre el botón “Importar datos” la aplicación mostrará un cuadro de diálogo como el que se muestra a continuación:

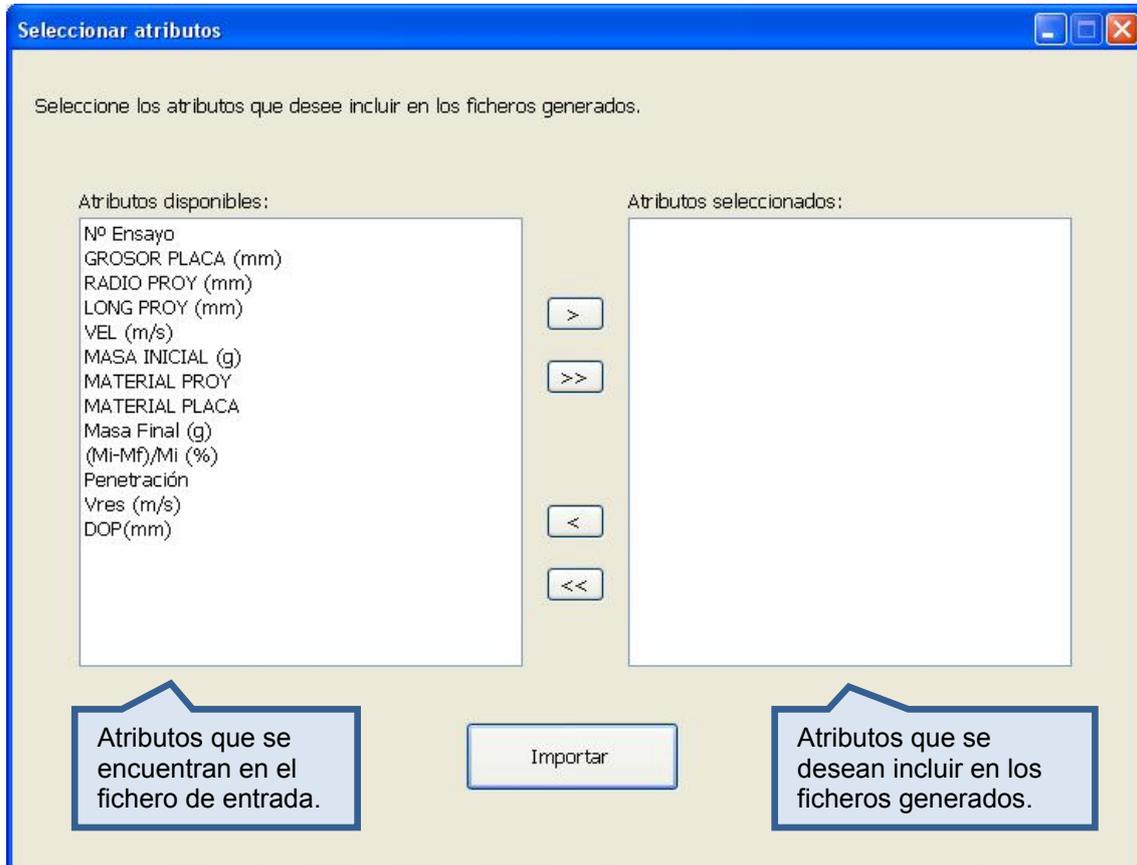


Figura 52. Selección de atributos

Este cuadro de diálogo muestra en dos listas los atributos disponibles en el fichero de entrada, y los atributos seleccionados por el usuario para sus ficheros (inicialmente vacía). Para realizar la selección de atributos deseada se encuentran disponibles las siguientes operaciones:

- **Añadir atributo(s):** Mediante el botón “>” se añaden los atributos seleccionados en la lista de “Atributos disponibles” a la lista “Atributos seleccionados”.
- **Eliminar atributo(s):** El botón “<” quita de la selección del usuario los atributos seleccionados previamente en la lista “Atributos seleccionados”, volviendo a aparecer estos en la lista “Atributos disponibles”.

- **Añadir todos:** Al hacer clic sobre el botón “>>” se añadirán todos los atributos que se encuentren en la lista “Atributos disponibles” a la lista “Atributos seleccionados”.
- **Eliminar todos:** Si el usuario quiere eliminar todos los atributos de la lista “Atributos seleccionados” podrá hacerlo mediante un clic sobre el botón “<<”.

Una vez terminada la selección de atributos deseados para los ficheros a generar se hace clic sobre el botón “Importar”, y dichos atributos aparecerán en la lista de “Atributos del fichero de datos” de la interfaz principal de la aplicación.

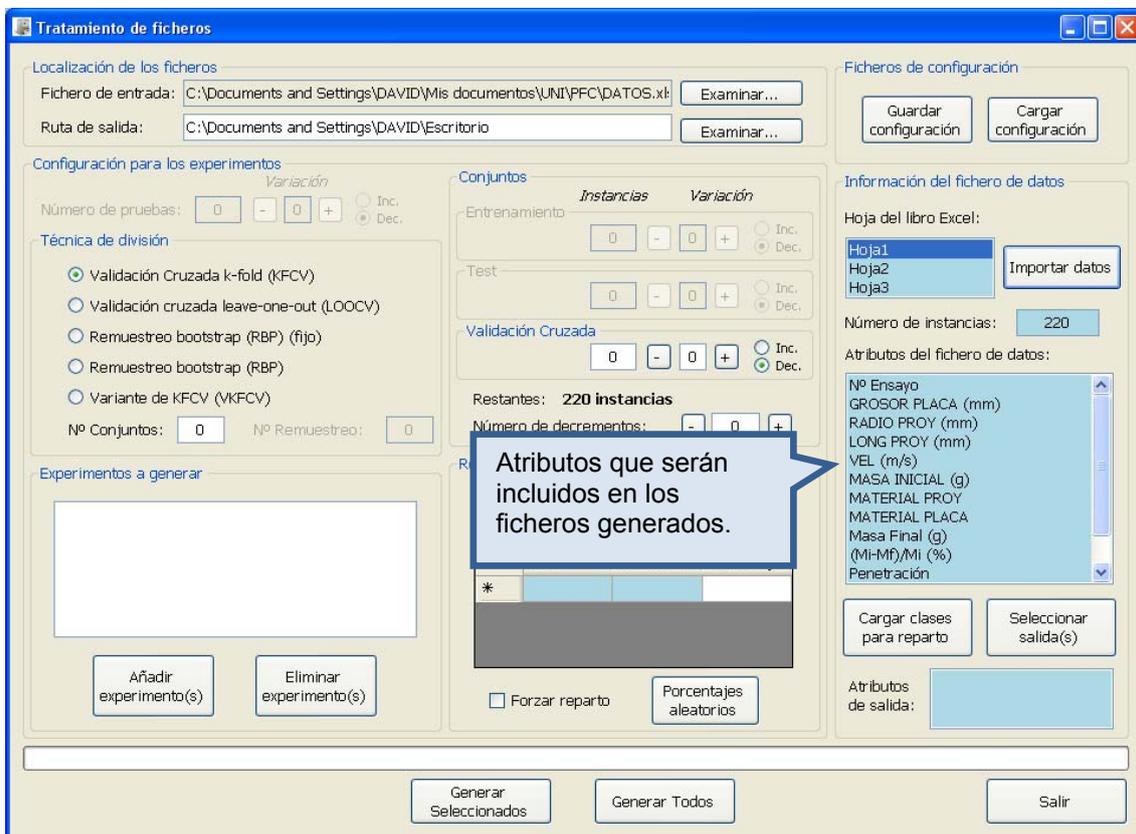


Figura 53. Atributos seleccionados

12.3.2 Atributos de salida

Tanto para clasificación como para regresión es necesario especificar que atributo o atributos serán los que la red de neuronas deberá predecir. Para ello se debe seleccionar uno o varios atributos cualesquiera de la lista de atributos seleccionados previamente y a continuación hacer clic sobre el botón “Seleccionar salida(s)”. Una vez seleccionados, los atributos aparecerán en el cuadro de texto “Atributos de salida”, tal y como se puede observar en la siguiente ilustración:

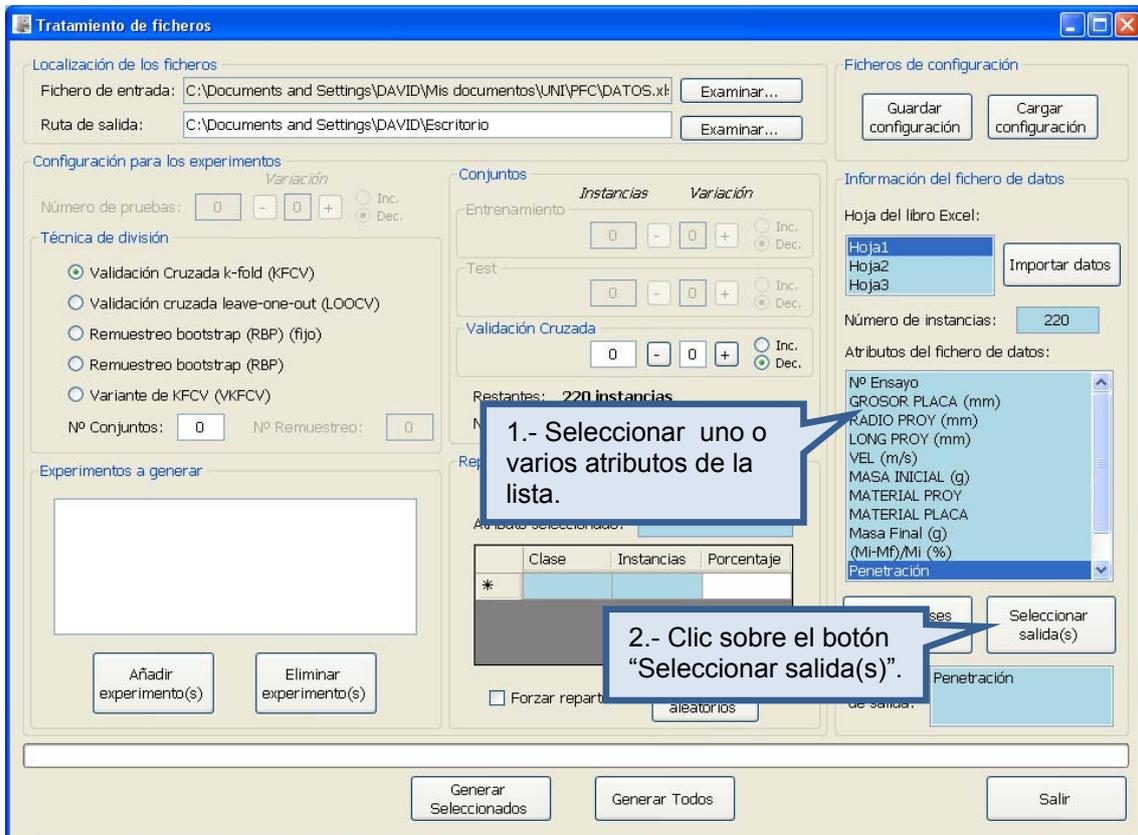


Figura 54. Atributos de salida



12.4 Configuración de experimentos

Una vez importados los datos del fichero de entrada y se han seleccionado los atributos del mismo, el usuario debe configurar cada uno de los experimentos que desee preparar. A continuación se detalla el proceso de configuración para un experimento nuevo.

12.4.1 Técnica de división

El primer parámetro de configuración que permite la aplicación es la técnica a emplear para la división del conjunto de instancias de partida para conformar los conjuntos de entrenamiento, test y validación cruzada. Haciendo clic sobre el nombre de la técnica deseada ésta queda seleccionada para la generación de los ficheros. Las técnicas disponibles en la aplicación son las siguientes:

- **Validación cruzada k-fold (KFCV):** Este método implica la partición de los datos en k conjuntos separados, donde k es el número de conjuntos elegidos. Esta separación se suele realizar generalmente de forma aleatoria en k disjuntos de un tamaño similar. El conjunto de ensayos de entrenamiento se utiliza para entrenar la red y se puede dividir según el experimento a realizar en entrenamiento y validación.
- **Validación cruzada leave-one-out (LOOCV):** Este método es similar a KFCV pero sólo utiliza un dato para el conjunto de test. Es igual al método KFCV cuando k es igual al tamaño del conjunto de datos, excepto en que consume más recursos que el anterior, ya que necesita utilizar todos los patrones como conjunto de test y se ejecuta tantas veces como patrones tenga el conjunto original, n .
- **Remuestreo bootstrap (RBP):** A partir del conjunto original de n elementos se seleccionan aleatoriamente b elementos con repetición para producir un nuevo subconjunto de tamaño n . Este proceso se repite varias veces dando lugar varios conjuntos de datos diferentes. Los datos que no han sido destinados al conjunto de entrenamiento forman parte del conjunto de test mientras que todos los que han sido seleccionados n , con sus repeticiones se utilizan para entrenar la red. Por tanto el número de patrones destinados a test puede variar, pero el resultado es que se consigue un conjunto de datos de mayor tamaño en entrenamiento y validación.
- **Remuestreo bootstrap (RBP fijo):** Variación del método RBP en el que se define explícitamente el número de datos a utilizar para entrenamiento (se pueden duplicar).

- **Variante de KFCV (VKFCV):** Este método comparte con el KFCV la misma técnica de reparto de patrones basado en la selección aleatoria para generar conjuntos mutuamente excluyentes. El conjunto original de patrones es dividido en dos o tres subconjuntos (train, test y CV) en función de las necesidades de la experimentación a realizar. Esta división se realiza de forma aleatoria pero con la característica de no existir repetición entre los elementos que forman, por lo que se trata de conjuntos disjuntos.

En función de la técnica seleccionada, la aplicación requerirá que se introduzca un valor en el cuadro de texto “Nº Conjuntos” o “Nº Remuestreo”.

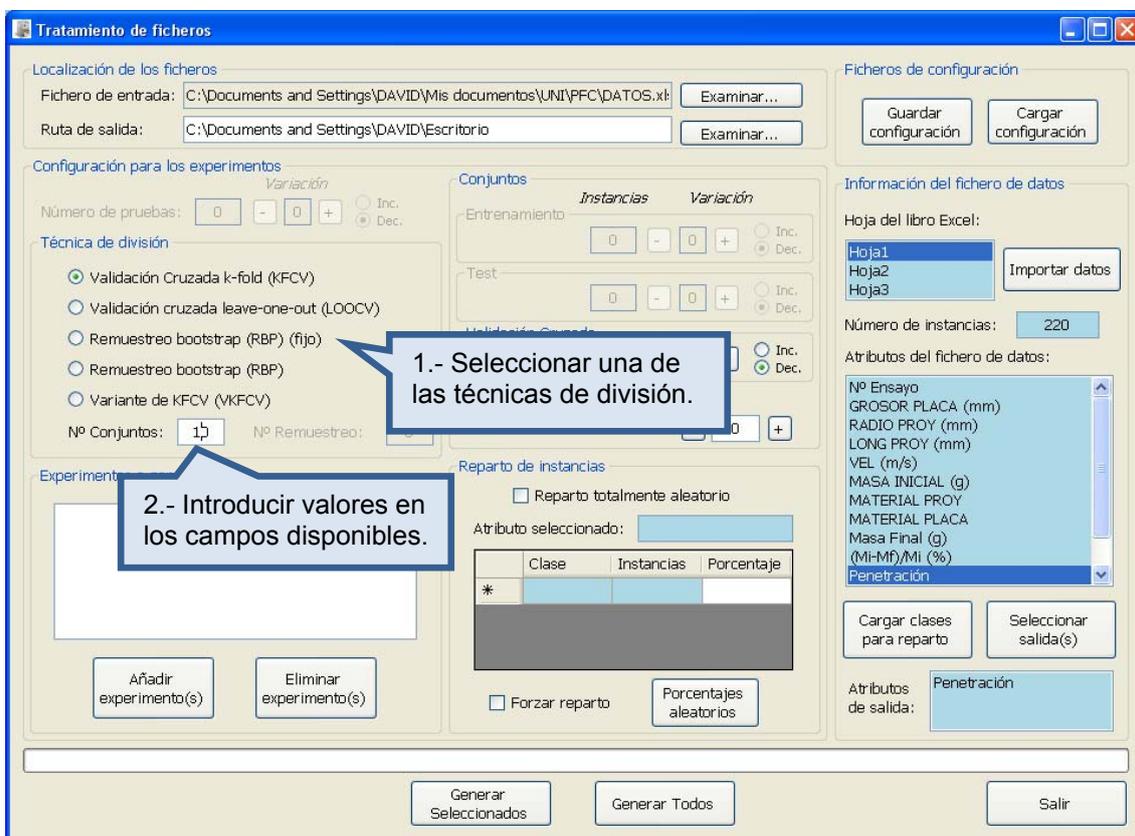


Figura 55. Seleccionar técnica de división

12.4.2 Número de pruebas

Una vez seleccionada la técnica de división, la aplicación permite indicar el número de pruebas que se desean generar para el experimento. Para ello el usuario debe introducir el número en el cuadro de texto “Número de pruebas”, tal y como se observa en la siguiente ilustración:

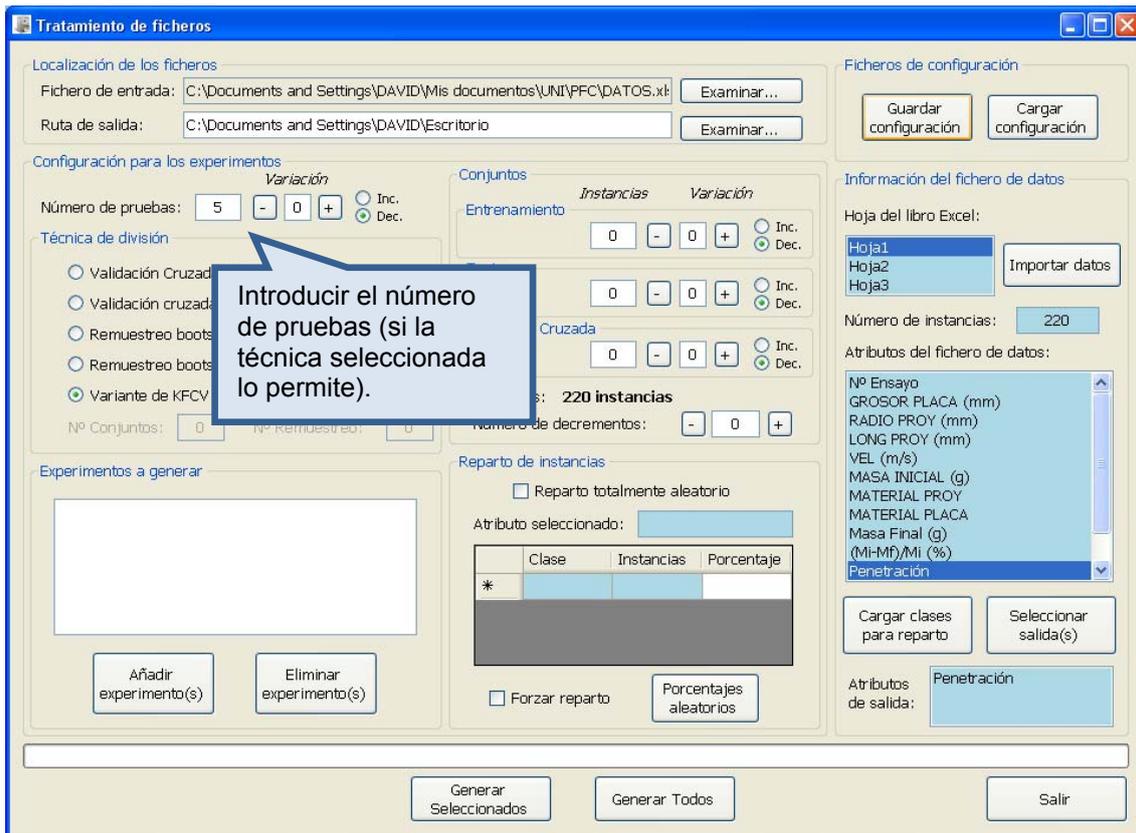


Figura 56. Número de pruebas

En caso de haber seleccionado la técnica de división “Validación cruzada k-fold (KFCV)” o “Validación cruzada leave-one-out (LOOCV)”, la aplicación no permite modificar el número de pruebas, ya que éste depende directamente del “Nº Conjuntos” introducido.

12.4.3 Conjuntos de instancias

Para configurar un experimento que se desee preparar la aplicación debe conocer la cantidad de instancias que se desean incluir en cada uno de los conjuntos de instancias. Estos conjuntos son tres: instancias de entrenamiento, instancias de test, e instancias de validación cruzada. En función de la técnica de división seleccionada por el usuario la aplicación permitirá modificar todos los conjuntos de instancias, o únicamente el conjunto de validación cruzada.

La siguiente imagen muestra los cuadros de texto en los que se debe introducir el número de instancias de cada conjunto:

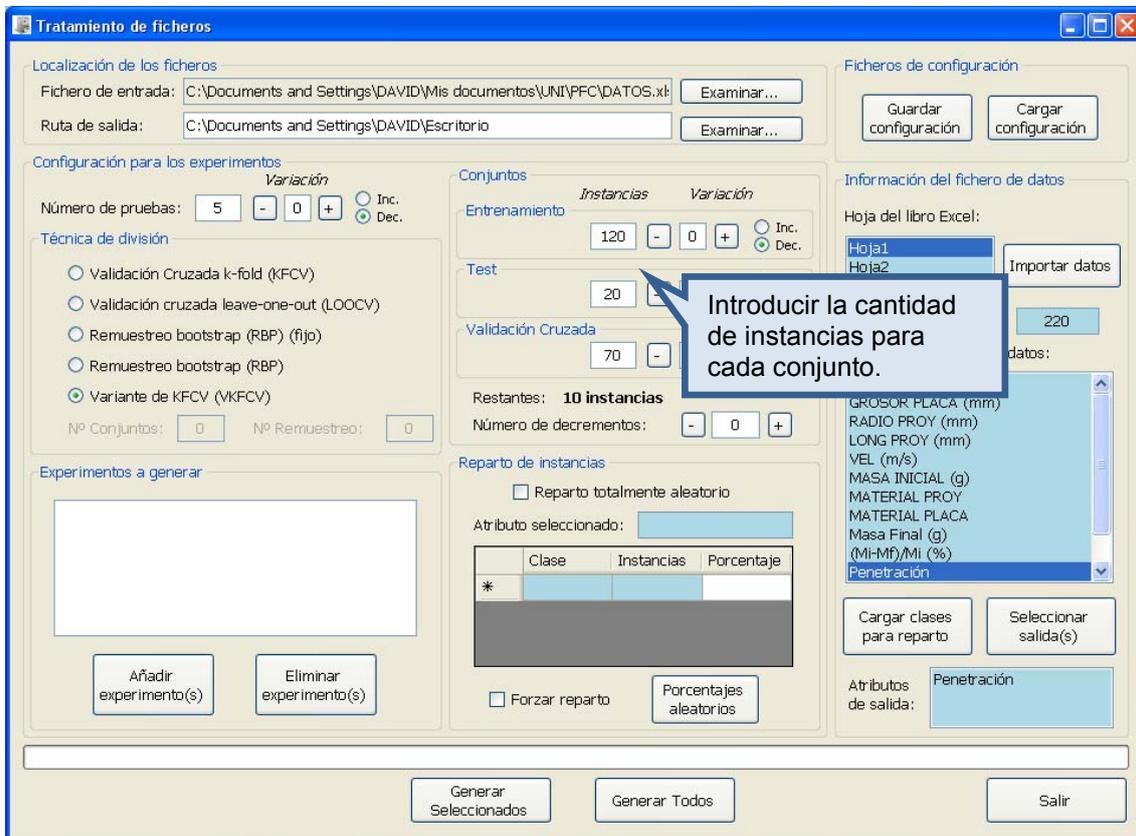


Figura 57. Conjuntos de instancias

12.4.4 Reparto de instancias

La aplicación permite que el reparto de instancias entre los distintos conjuntos se realice de dos formas: aleatoria o "dirigida", según el valor que tome alguno de los atributos seleccionados previamente.

Si se desea que el reparto de instancias sea completamente aleatorio el usuario debe seleccionar la opción “Reparto totalmente aleatorio”, tal y como muestra la siguiente imagen:

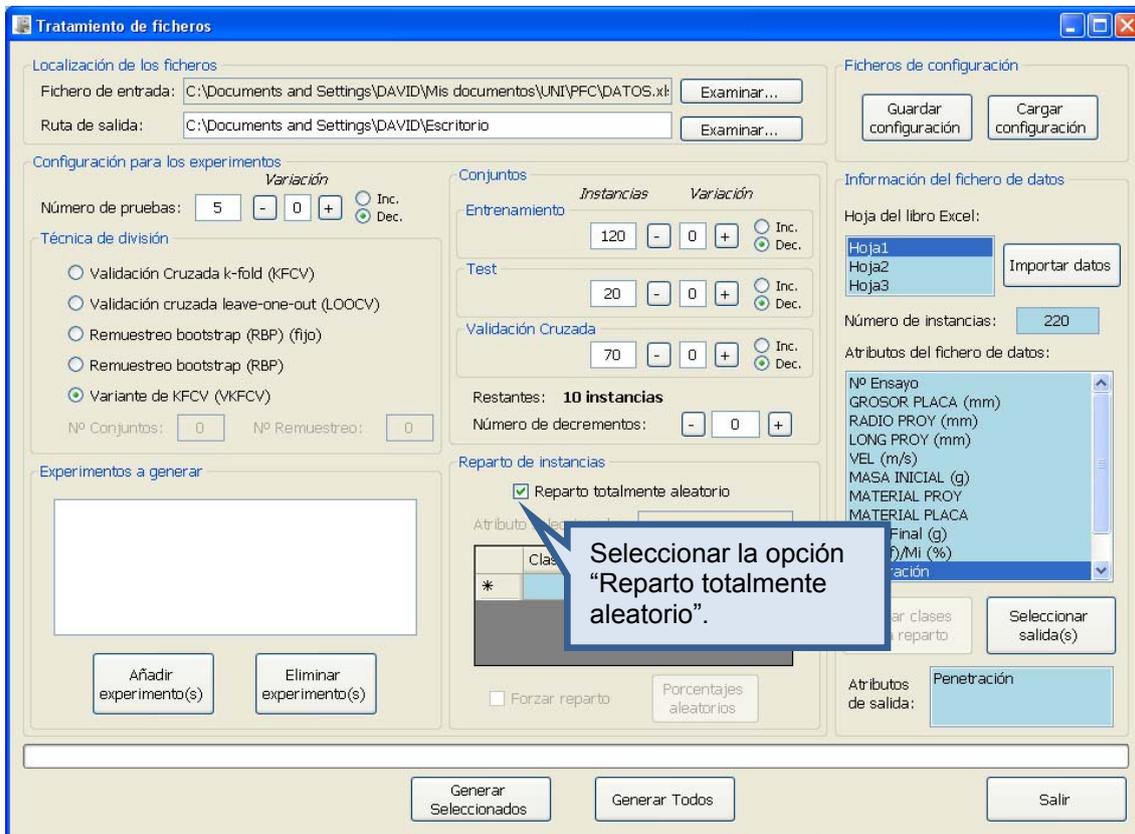


Figura 58. Reparto de instancias totalmente aleatorio

Para utilizar la opción de reparto “dirigido”, en primer lugar se debe seleccionar uno de los atributos y hacer clic sobre el botón “Cargar clases para reparto”. De este modo la aplicación cargará todos los posibles valores que encuentre para ese atributo en la tabla de “Reparto de instancias”. En dicha tabla el usuario deberá indicar los porcentajes que desea incluir de cada clase en cada conjunto de instancias, o bien hacer clic sobre el botón “Porcentajes aleatorios” para que la aplicación asigne los porcentajes de manera aleatoria.

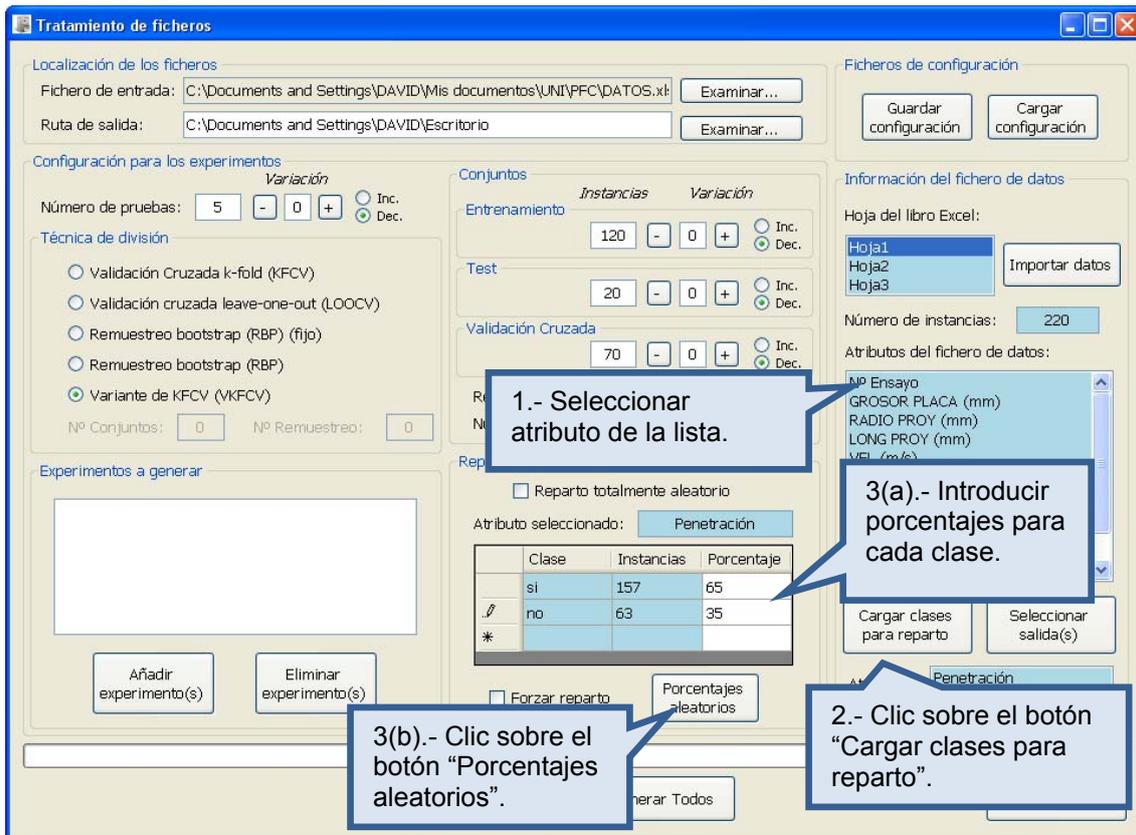


Figura 59. Reparto de instancias "dirigido"

La opción "Forzar reparto" da la posibilidad de completar la cantidad de instancias del conjunto con instancias de otra clase si la cantidad de instancias de la clase deseada no son suficientes. En caso de no seleccionar esta opción la aplicación mostrará un aviso si se ha tenido que modificar alguna cantidad de instancias durante el proceso.

12.4.5 Añadir experimentos a preparar

Una vez definidos todos los parámetros de configuración para el experimento se hace clic sobre el botón “Añadir experimento(s)” y este se añade a la lista de “Experimentos a generar”, tal y como se aprecia en la siguiente ilustración:

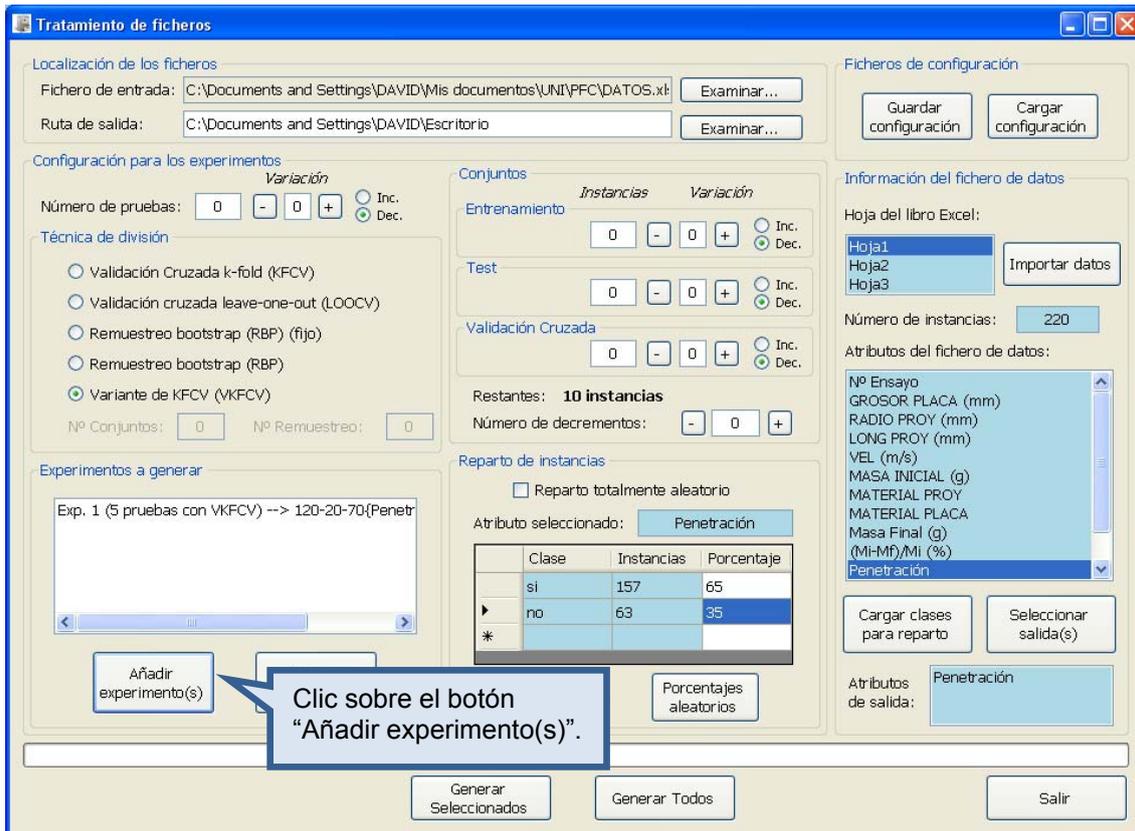


Figura 60. Añadir experimento a preparar

El experimento se añade a la lista de “Experimentos a preparar” con el nombre Experimento y un número secuencial correspondiente a su lugar en la lista. Además, se almacenan los parámetros definidos previamente de forma abreviada (número de pruebas, número de ensayos para Train, número de ensayos para Test, número de ensayos para Validación Cruzada, atributo de reparto y porcentajes de cada clase).

12.4.6 Añadir experimentos de forma automatizada

Para simplificar la función de añadir múltiples experimentos de forma automatizada, la aplicación permite, una vez definidos los parámetros del experimento, indicar el decremento a realizar sobre cada uno de ellos junto con el número de decrementos a realizar. En la siguiente figura se observa un ejemplo:

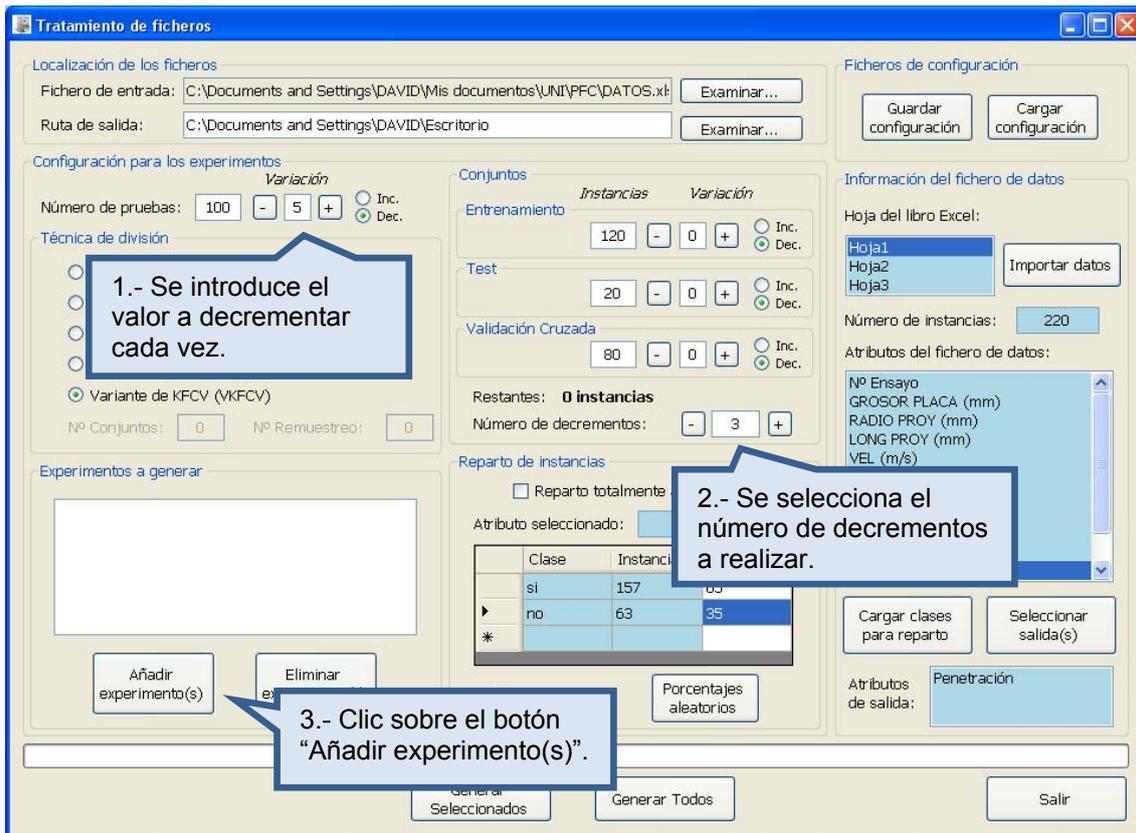


Figura 61. Añadir experimento de forma automatizada

En este caso se añadirán 4 experimentos a preparar a la lista una vez se haga clic en el botón "Añadir Experimento(s)", el experimento con los parámetros definidos junto con 3 decrementos del mismo, variando el número de pruebas (de 5 en 5 unidades), y el número de ensayos para Train (de 10 en 10 unidades).

El resultado es el siguiente:

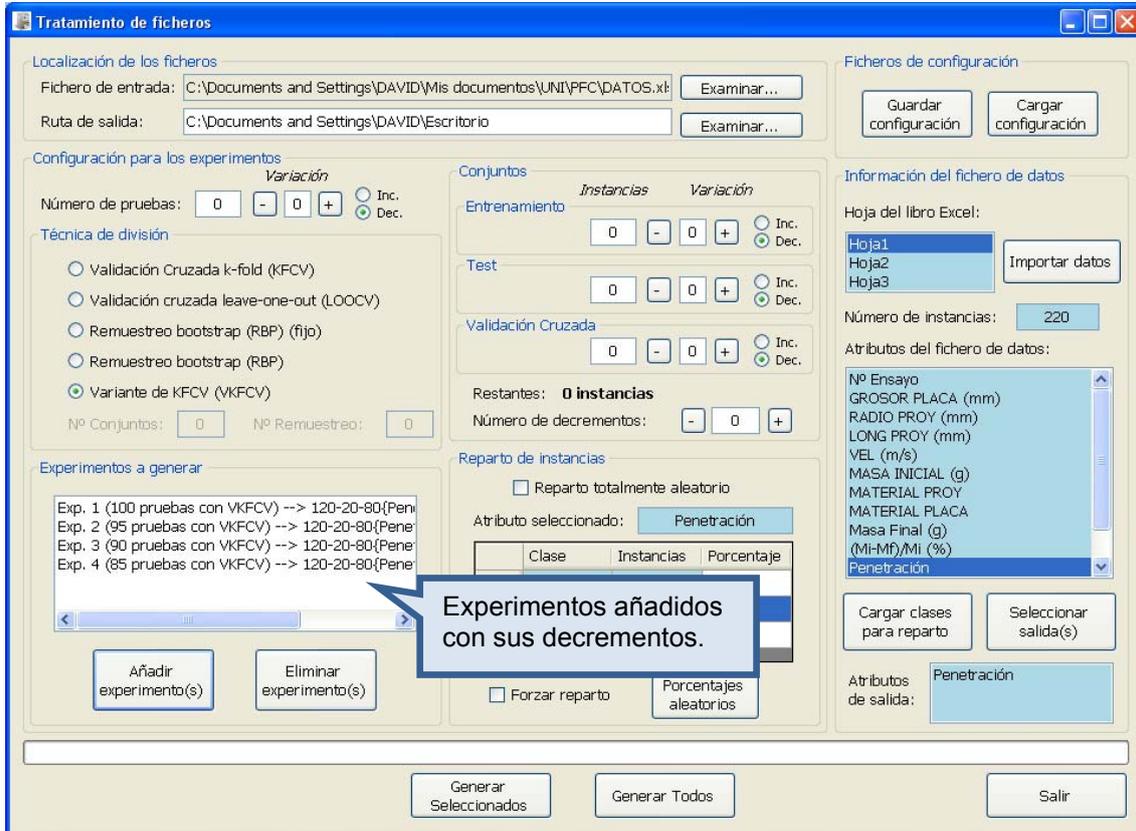


Figura 62. Ejemplo de experimentos añadidos de forma automatizada

12.4.7 Eliminar experimentos

Una vez añadido un experimento a la lista (o varios), la aplicación permite eliminarlos simplemente seleccionándolos en la lista y haciendo clic sobre el botón “Eliminar experimento(s)”. (NOTA: Para seleccionar varios usar CONTROL + CLIC).

En el siguiente ejemplo se quieren eliminar los experimentos 1 y 3:

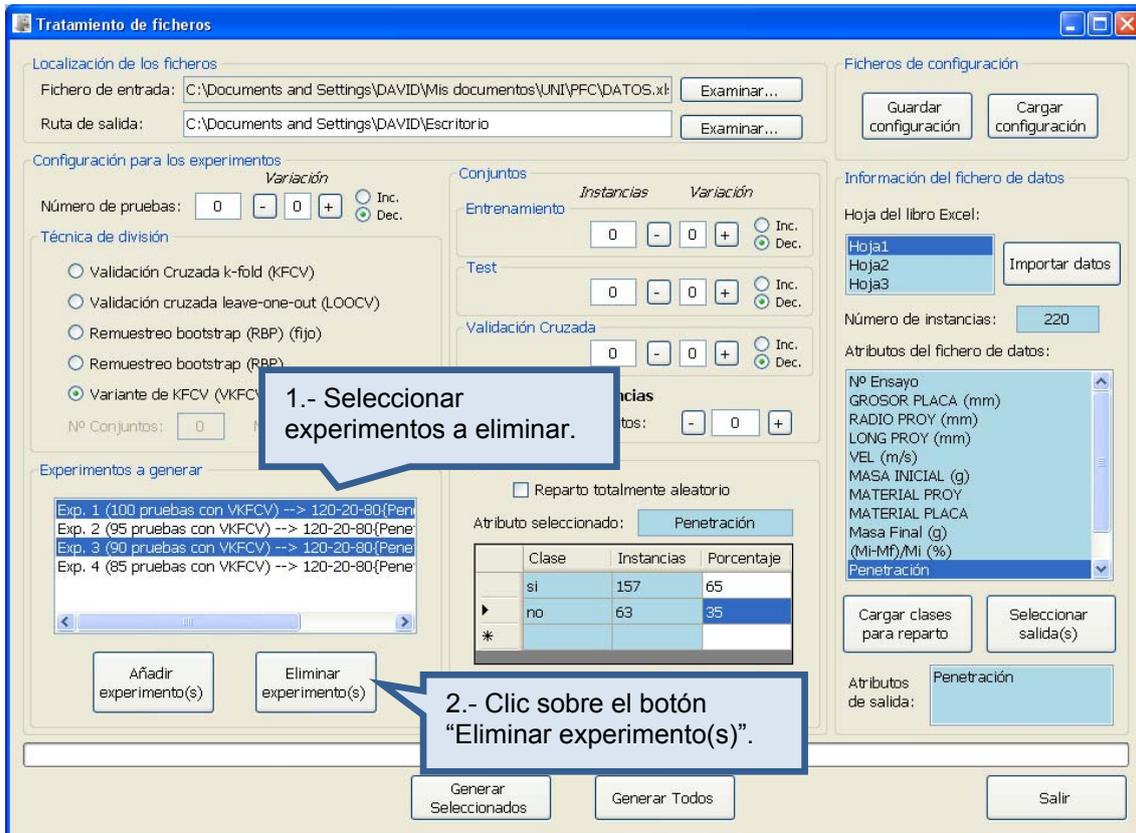


Figura 63. Eliminar varios experimentos

El resultado es el siguiente:

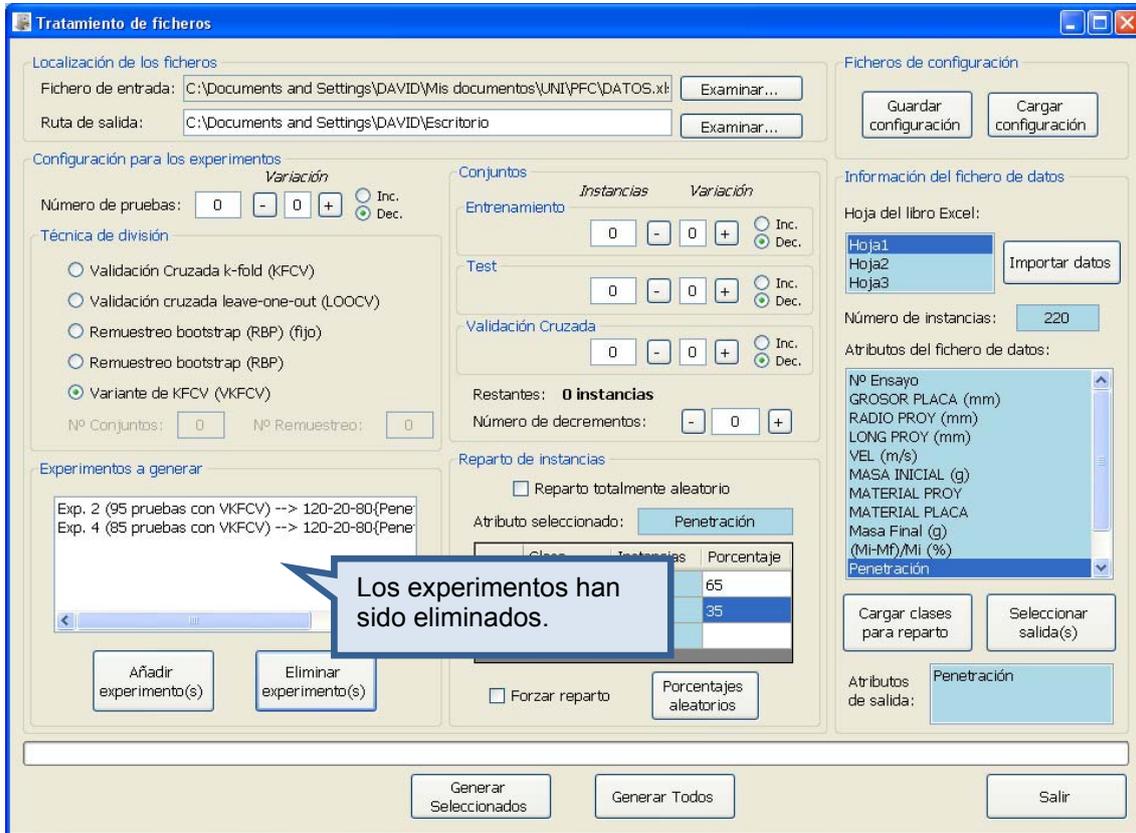


Figura 64. Ejemplo de experimentos eliminados

12.5 Generación de ficheros

Cuando el usuario ha añadido los experimentos que desea preparar a la lista “Experimentos a generar”, la aplicación ofrece dos posibilidades:

- **Generar seleccionados:** Haciendo clic sobre el botón “Generar seleccionados” la aplicación generará únicamente los ficheros de los experimentos que hayan sido seleccionados de la lista. (NOTA: Para seleccionar varios usar CONTROL + CLIC).
- **Generar todos:** El botón “Generar todos” ordena a la aplicación que genere los ficheros correspondientes a todos y cada uno de los experimentos que se encuentren en la lista “Experimentos a generar”.

Durante todo el proceso de generación la aplicación muestra una barra de progreso del experimento que se esté procesando, permitiendo al usuario supervisar dicho proceso. A continuación se puede observar el indicador situado en la parte inferior del interfaz:

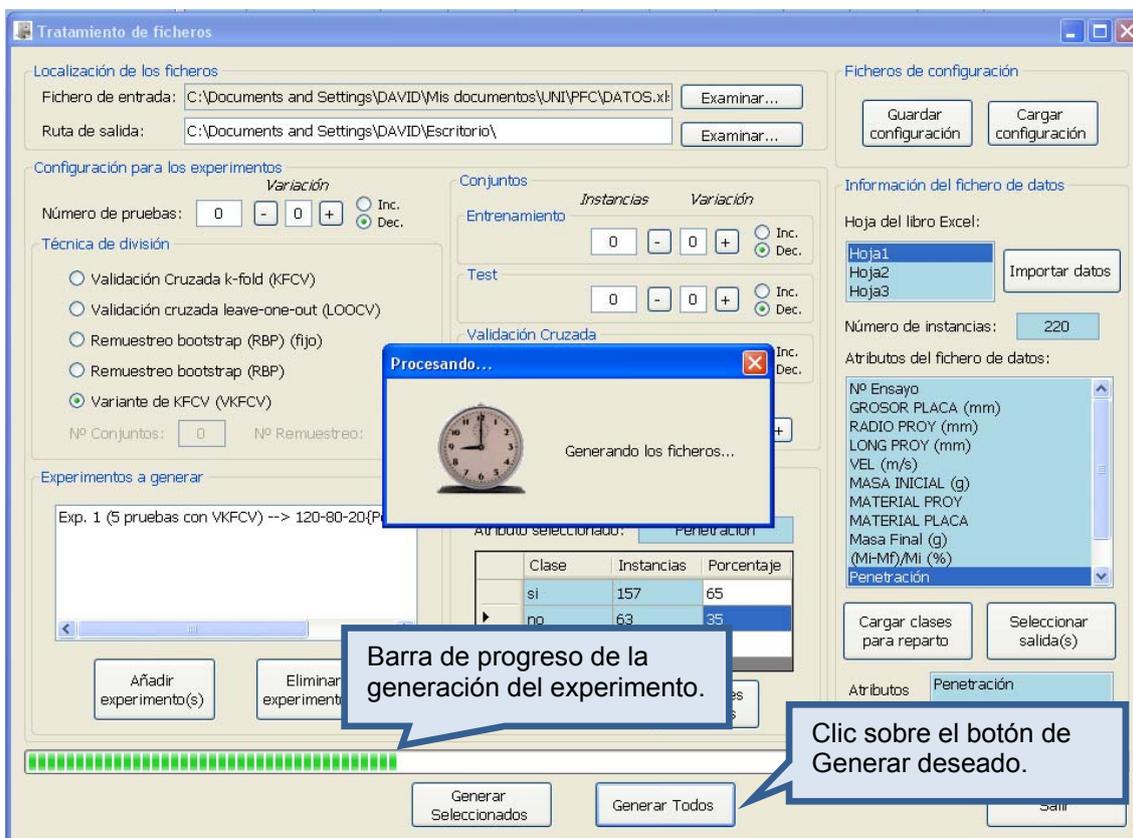


Figura 65. Proceso de generación de ficheros

Finalmente, cuando el proceso de generación finaliza, la aplicación muestra un mensaje por pantalla informando al usuario:

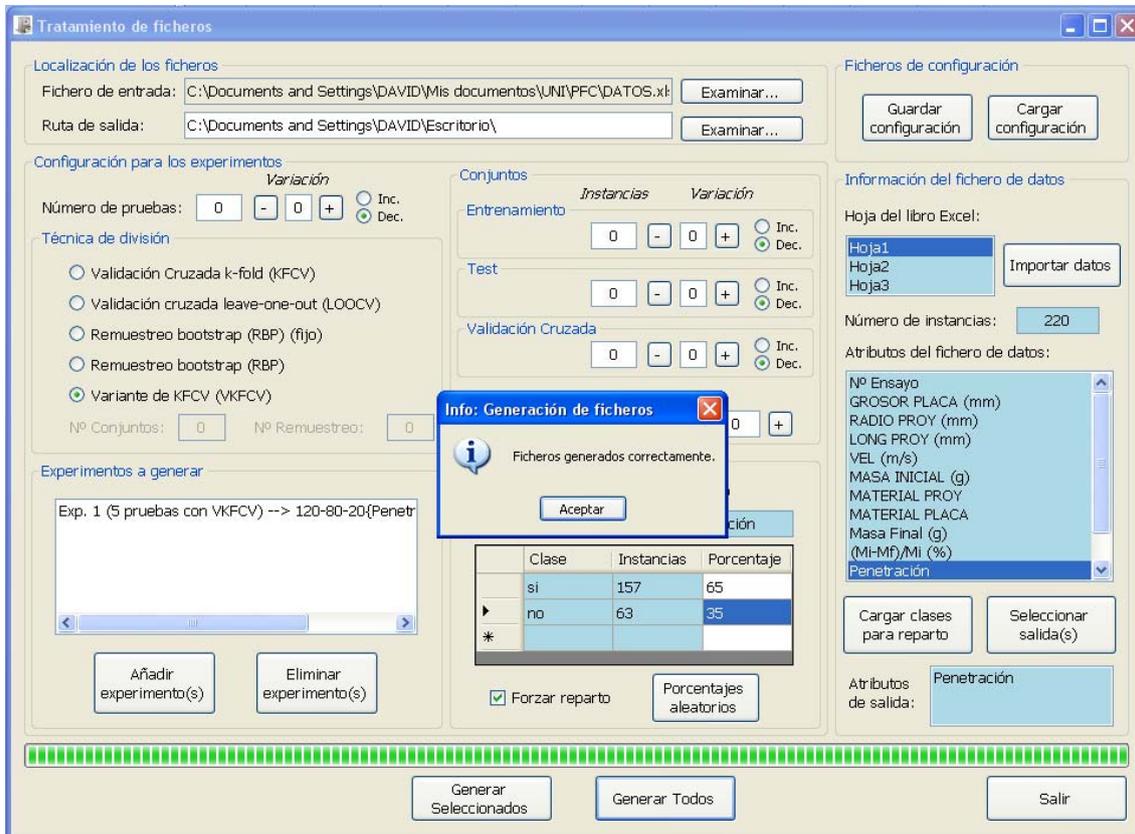


Figura 66. Ficheros generados correctamente

El usuario podrá encontrar los ficheros generados en la carpeta indicada en el cuadro de texto “Ruta de salida”. La aplicación crea una carpeta independiente para cada uno de los experimentos generados. Dentro de cada carpeta de experimento se puede encontrar tres tipos de ficheros:

- **Ficheros de entrenamiento o train:** Son los ficheros designados para el entrenamiento de la red de neuronas artificiales. Su nomenclatura sigue el siguiente formato: “*TrainXX.txt*”, donde XX indica el número de prueba a la que corresponde.
- **Ficheros de validación o test:** Son los ficheros designados para la validación de la red, una vez entrenada. Su nomenclatura es del tipo: “*TestXX.txt*”, donde XX indica el número de prueba a la que corresponde.

- **Ficheros de validación cruzada o cross validation:** Son los ficheros designados para el entrenamiento de la red de neuronas artificiales, en caso de utilizar como criterio de parada la validación cruzada. Su nomenclatura es: “*ValXX.txt*”, donde XX indica el número de prueba a la que corresponde. Si el usuario ha configurado el experimento sin este tipo de ficheros, no se encontrará ninguno dentro de la carpeta del mismo.

La siguiente imagen muestra un posible resultado de la generación de los ficheros de un experimento:

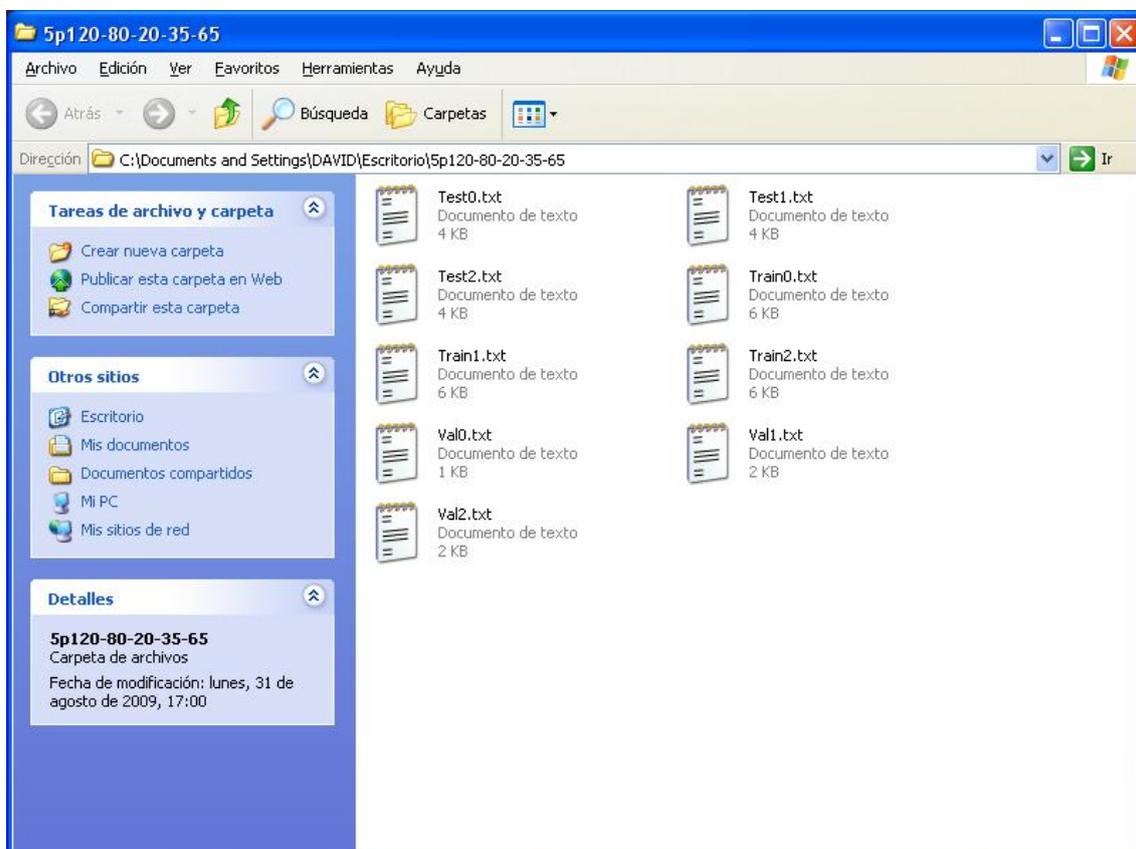


Figura 67. Ejemplo de ficheros generados

12.6 Gestión de la configuración

Es posible que el usuario desee almacenar la configuración de datos y experimentos realizada previamente en la aplicación. De este modo podrá realizar la generación de ficheros en otro momento, o bien trabajar con una configuración en cualquier momento que desee. La aplicación permite tanto guardar como cargar un fichero de configuración, tal y como se describe a continuación:

12.6.1 Guardar configuración

Cuando se ha completado la configuración de los datos y experimentos, el usuario debe hacer clic sobre el botón “Guardar configuración” situado en la parte superior derecha de la pantalla. La aplicación mostrará un cuadro de diálogo en el que se podrá seleccionar el fichero en el que almacenar la configuración actual. La siguiente imagen muestra un ejemplo de esta operación:

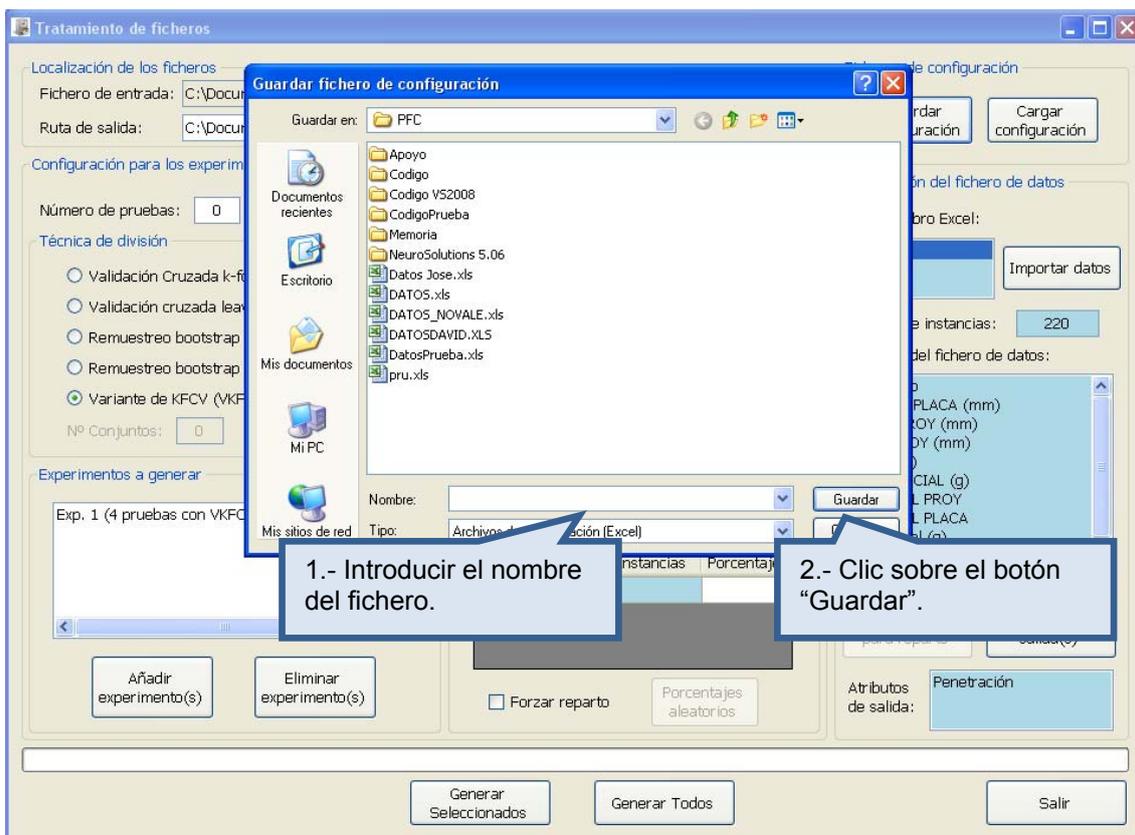


Figura 68. Guardar configuración

12.6.2 Cargar configuración

Si previamente se ha guardado la configuración de la aplicación en un fichero, éste podrá ser cargado en cualquier momento, devolviendo la aplicación al estado de configuración en el que se encontraba en el momento de guardar el fichero en cuestión. Para ello se debe hacer clic sobre el botón “Cargar configuración” y seleccionar el archivo de configuración en el cuadro de diálogo que muestra la aplicación. A continuación se puede observar el funcionamiento:

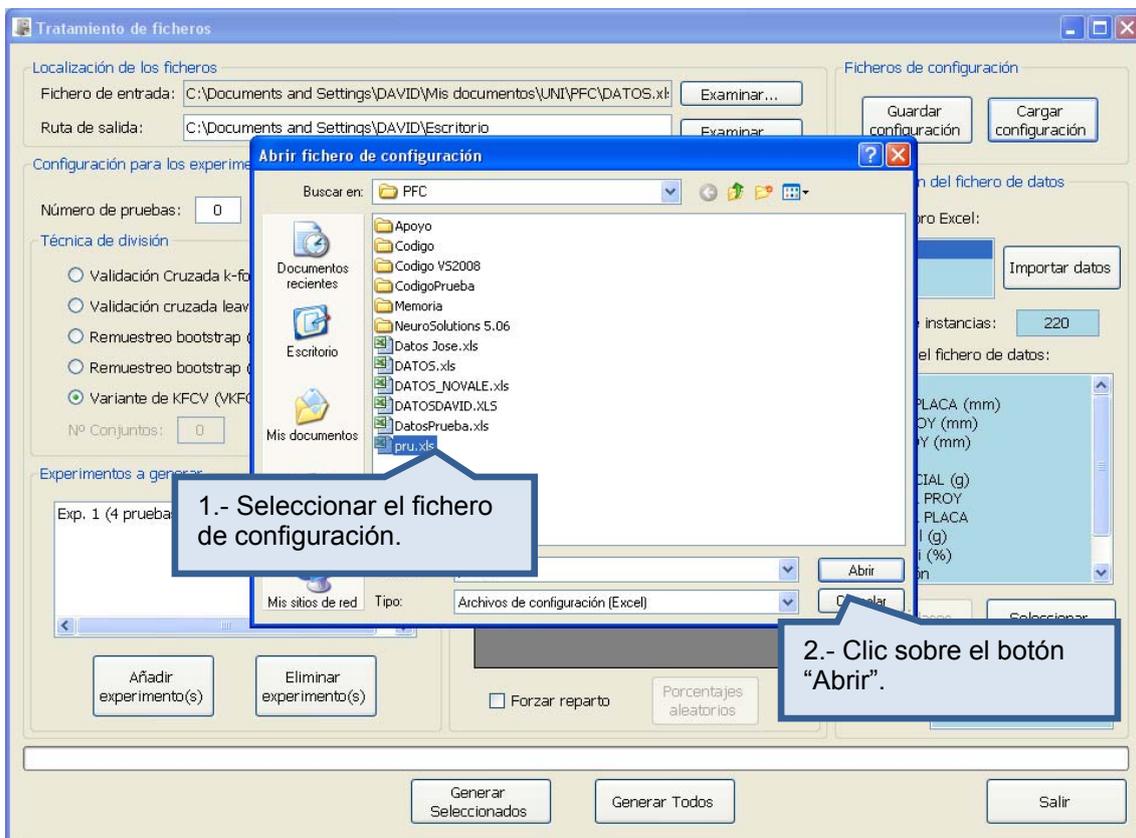


Figura 69. Cargar configuración



13 Anexo B: Formato para ficheros de entrada

La aplicación Tratamiento de ficheros de entrada requiere para su correcto funcionamiento que los ficheros de entrada empleados cumplan los requisitos enumerados a continuación:

- El fichero que incluya los datos a procesar debe ser un fichero creado por Microsoft Excel con la extensión “.xls”.
- La fila número uno de la hoja donde se encuentren los datos debe contener los nombres de los atributos en cada una de las columnas.
- Cada atributo se encontrará en una columna de la hoja, empezando por la columna uno de la hoja, añadiendo una columna para cada atributo de forma consecutiva a la anterior.
- Cada instancia estará representada en una fila de la hoja, empezando por la fila dos de la hoja, añadiendo una fila por cada instancia de forma consecutiva a la anterior.
- Es recomendable que el fichero no contenga macros o similares, ya que podría darse el caso de ser incompatibles con el correcto funcionamiento de la aplicación.

14 Anexo C: Creación de una red de neuronas en NeuroSolutions

Para crear una nueva red, se abre el programa NeuroSolutions y se hace clic sobre el botón “NBuilder” indicado en la figura:



Figura 70. Barra de herramientas de NeuroSolutions

A continuación se mostrará la primera pantalla del “NeuralBuilder” donde se selecciona el modelo para la red (en el presente proyecto se ha utilizado “Multilayer Perceptron”):

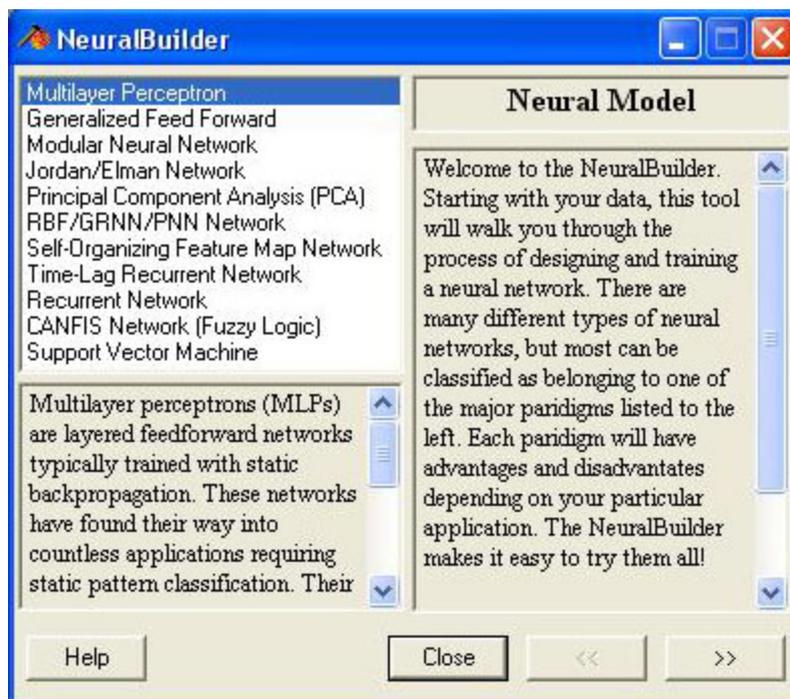


Figura 71. Modelo Neuronal

Se hace clic en el botón “>>” y aparece la pantalla para seleccionar los datos de entrenamiento (Train):

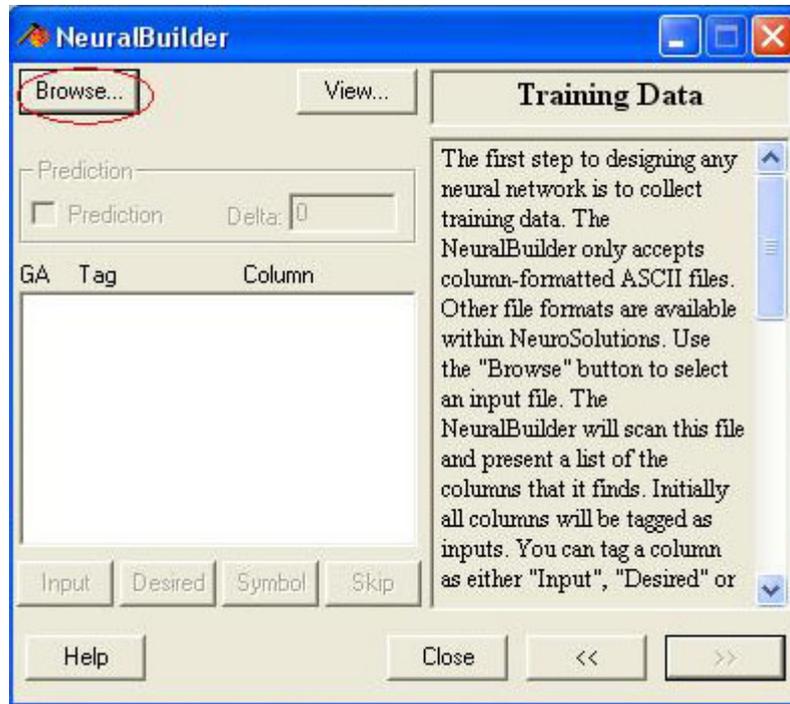
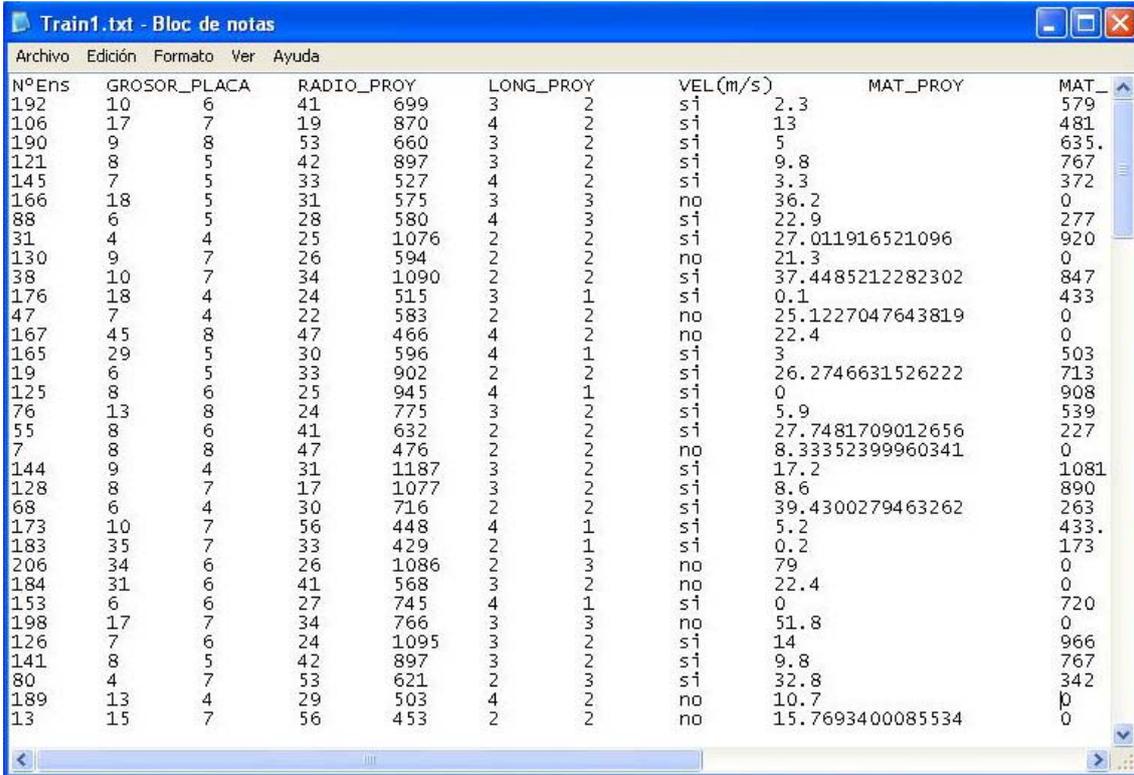


Figura 72. Fichero de entrenamiento

Para crear la red desde el propio NeuroSolutions, los ficheros donde se encuentran los datos deben ser ficheros ASCII, organizados en filas y cada variable en una columna, es decir, en forma de “tabla”. Un ejemplo de fichero de datos sería el que se observa en la siguiente figura:



Nº Ens	GROSOR_PLACA	RADIO_PROY	LONG_PROY	VEL (m/s)	MAT_PROY				
192	10	6	41	699	3	2	si	2.3	579
106	17	7	19	870	4	2	si	13	481
190	9	8	53	660	3	2	si	5	635.
121	8	5	42	897	3	2	si	9.8	767
145	7	5	33	527	4	2	si	3.3	372
166	18	5	31	575	3	3	no	36.2	0
88	6	5	28	580	4	3	si	22.9	277
31	4	4	25	1076	2	2	si	27.011916521096	920
130	9	7	26	594	2	2	no	21.3	0
38	10	7	34	1090	2	2	si	37.4485212282302	847
176	18	4	24	515	3	1	si	0.1	433
47	7	4	22	583	2	2	no	25.1227047643819	0
167	45	8	47	466	4	2	no	22.4	0
165	29	5	30	596	4	1	si	3	503
19	6	5	33	902	2	2	si	26.2746631526222	713
125	8	6	25	945	4	1	si	0	908
76	13	8	24	775	3	2	si	5.9	539
55	8	6	41	632	2	2	si	27.7481709012656	227
7	8	8	47	476	2	2	no	8.33352399960341	0
144	9	4	31	1187	3	2	si	17.2	1081
128	8	7	17	1077	3	2	si	8.6	890
68	6	4	30	716	2	2	si	39.4300279463262	263
173	10	7	56	448	4	1	si	5.2	433.
183	35	7	33	429	2	1	si	0.2	173
206	34	6	26	1086	2	3	no	79	0
184	31	6	41	568	3	2	no	22.4	0
153	6	6	27	745	4	1	si	0	720
198	17	7	34	766	3	3	no	51.8	0
126	7	6	24	1095	3	2	si	14	966
141	8	5	42	897	3	2	si	9.8	767
80	4	7	53	621	2	3	si	32.8	342
189	13	4	29	503	4	2	no	10.7	0
13	15	7	56	453	2	2	no	15.7693400085534	0

Figura 73. Fichero de datos en formato de texto plano

Para seleccionar el fichero deseado se hace clic en el botón “Browse...” y se selecciona el fichero.

A continuación se deben etiquetar las variables que aparecen en el fichero, como entrada (Input), salida (Desired) o ignorada (Skip), dependiendo de las variables que se quieran utilizar en la simulación. El resultado puede ser como el de la figura:

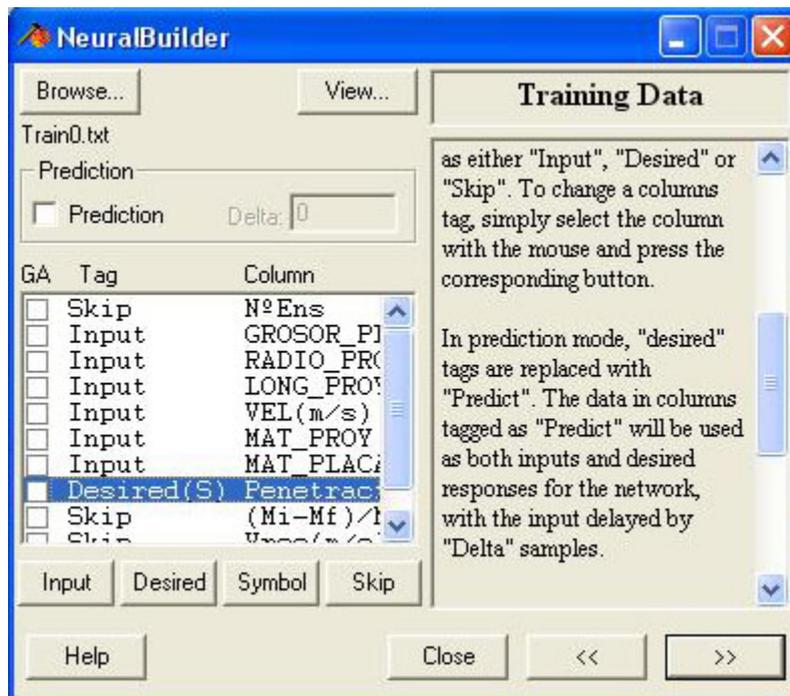


Figura 74. Variables de entrenamiento

Se hace clic en “>>” y en la siguiente ventana que aparece, hay que seleccionar el fichero Test0.txt para realizar las pruebas durante la simulación. Para ello, basta con hacer clic en el botón “Input Test...” como se observa a continuación:

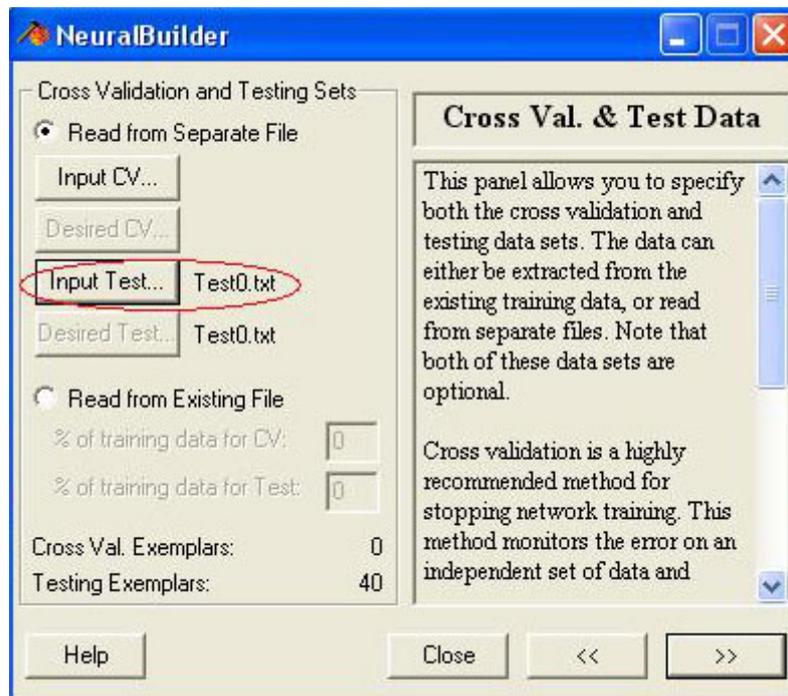


Figura 75. Fichero de Test

El siguiente paso es indicar el número de capas ocultas que se desea que tenga la red para la simulación, escribiéndolo en el apartado “Hidden Layers” como indica la figura:

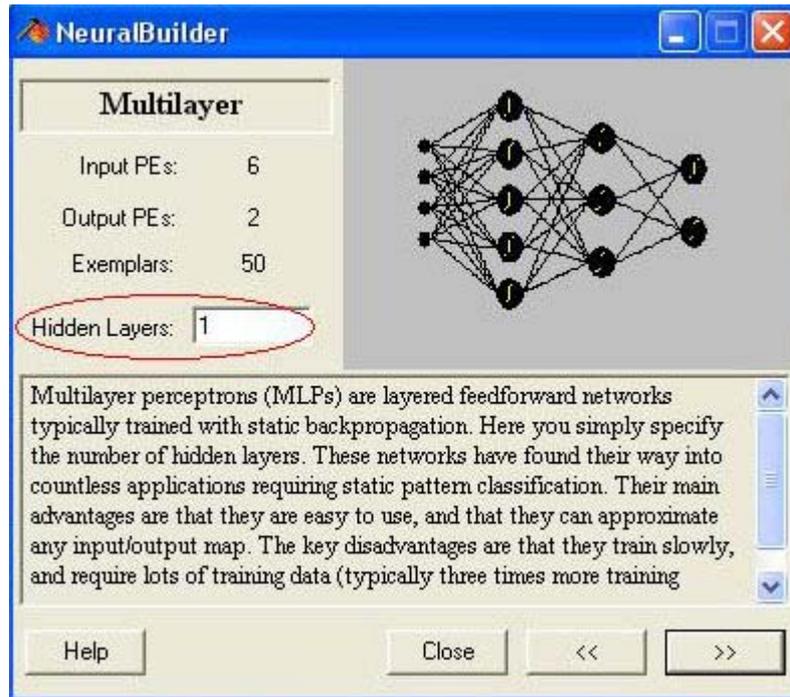


Figura 76. Número de capas ocultas

Dependiendo del número de capas ocultas elegido a continuación saldrá una nueva ventana por cada una de ellas donde permite modificar algunos parámetros, aunque para este caso no es necesario. Un ejemplo para la capa oculta #1 sería:

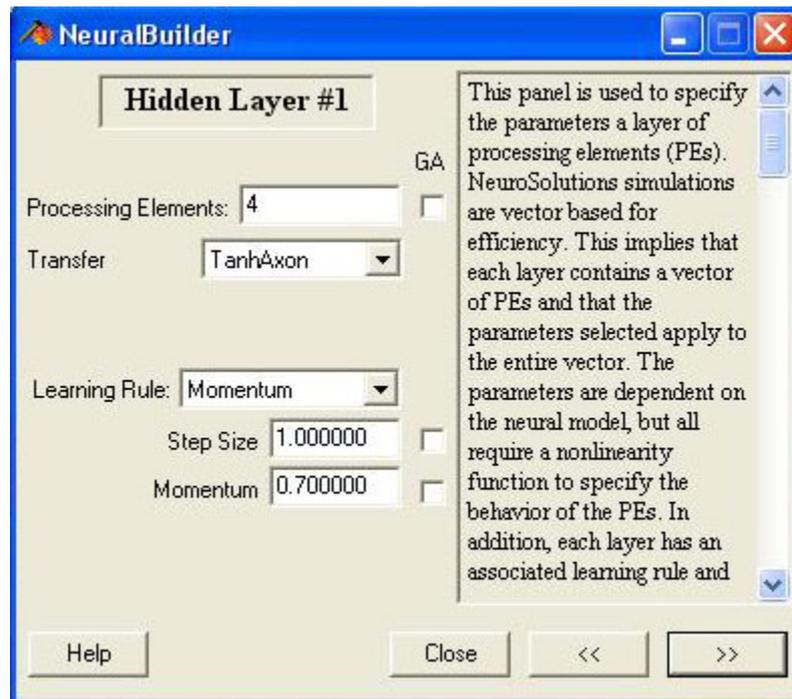


Figura 77. Número de neuronas ocultas

Una vez hecho clic en el botón “>>” de la última capa oculta, aparece una ventana para configurar la capa de salida de la red (en principio no es necesario tampoco en este caso):

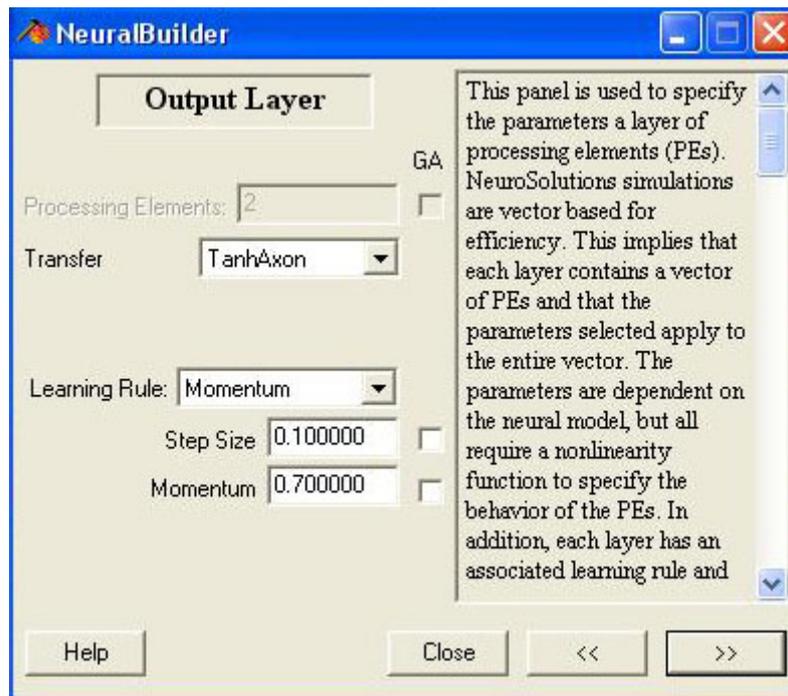


Figura 78. Capa de salida de la red

El siguiente paso consiste en modificar el control de aprendizaje y el criterio de parada de la red. Sin embargo, no es necesario modificar ninguna de estas opciones aquí, ya que la propia aplicación Asistente se encarga de ello como se detalla más adelante.

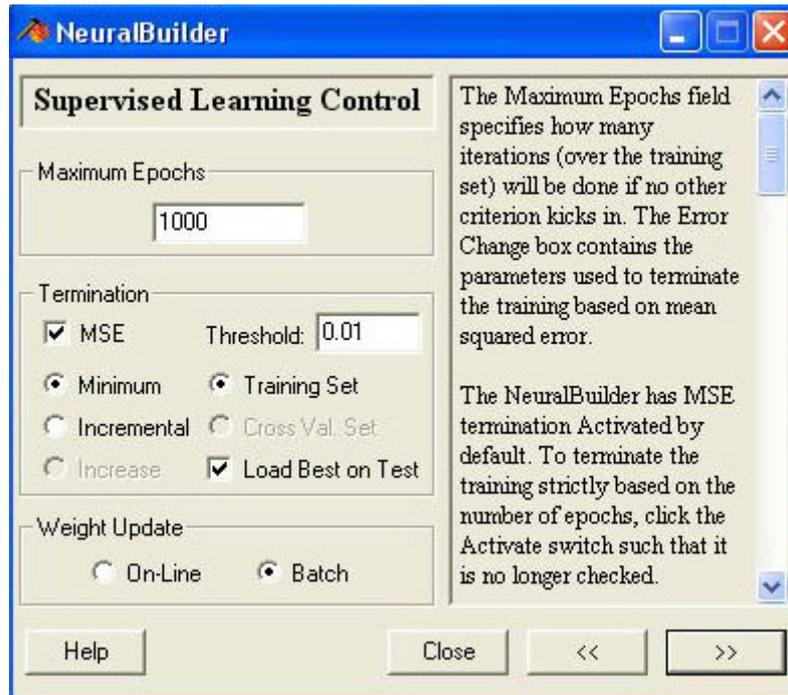


Figura 79. Criterio de parada del entrenamiento

Por último, la siguiente ventana permite configurar la salida de los resultados, y hacer clic en el botón “Build” para crear la red configurada:

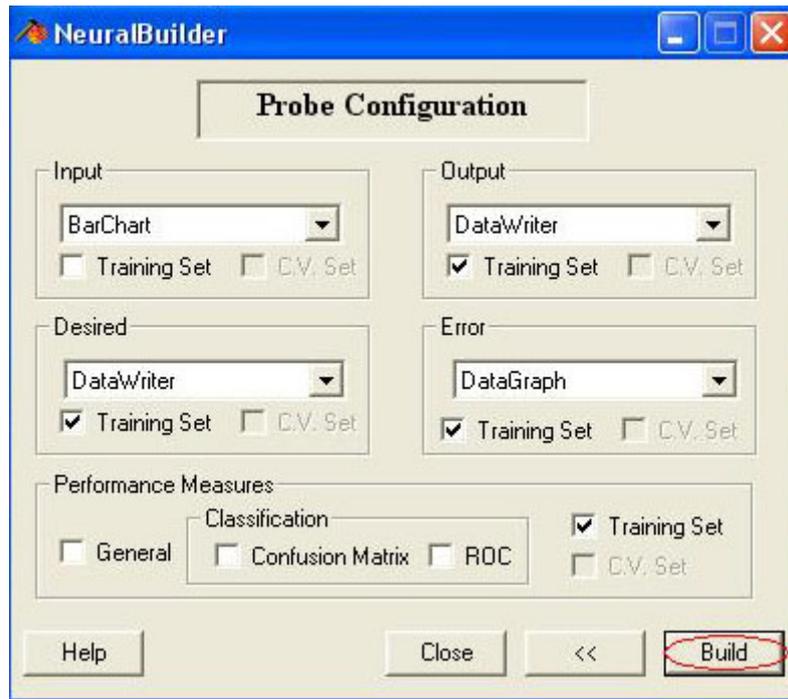


Figura 80. Opciones de salida

Un ejemplo de red creada por NeuroSolutions podría ser el siguiente:

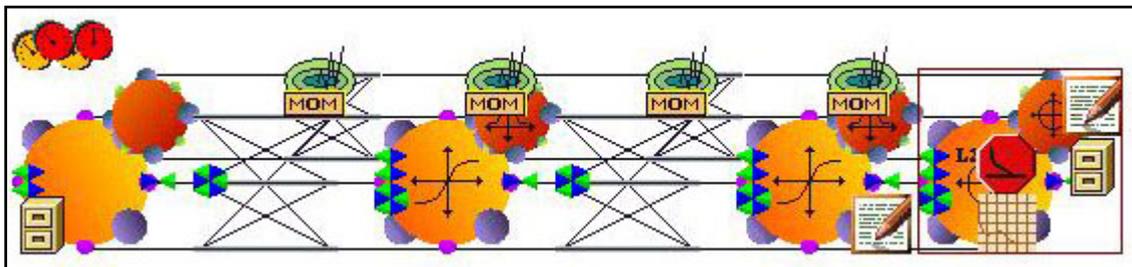


Figura 81. Red neuronal creada por NeuroSolutions



15 Anexo D: Manual de usuario Módulo de Comunicación

Aunque el módulo de comunicación desarrollado en este proyecto no ofrece interacción directa con el usuario puede resultar útil proporcionar cierta información acerca de sus requisitos y su modo de funcionamiento.

15.1 Requisitos de funcionamiento

Como requisito imprescindible para su correcto funcionamiento, el módulo de comunicación requiere lo siguiente:

- La librería desarrollada, denominada *ModuloCom.dll*, debe estar registrada en el sistema *Microsoft Windows XP* previamente a su utilización.
- *NeuroSolutions 5.0* instalado en el ordenador donde se vaya a utilizar el sistema.

15.2 Ejecución

Tanto el sistema experto como el simulador desarrollado en el proyecto tienen en común la forma de interaccionar con el módulo de comunicación. En ambos casos han de seguir tres pasos principales:

1. Introducir los parámetros de entrada en el objeto *ParamIn* de la clase *Comunicacion*.
2. Llamar al procedimiento *Execute()*.
3. Obtener los parámetros de salida del objeto *ParamOut* de la clase *Comunicacion*.

15.3 Comprobación de errores

Cuando el sistema experto o el simulador desarrollado ejecutan el procedimiento *Execute()* del módulo de comunicación, éste comprueba los parámetros introducidos en busca de errores, ya sean parámetros no encontrados o valores introducidos no válidos. La siguiente tabla detalla los parámetros de entrada y valores aceptados, indicando además el tipo de función a la que pertenecen: Train, Test o Producción:

Parámetro	Tipo	Posibles valores	Tipo de función
Funcion	Obligatorio	Train, Test, Produccion, TrainTest	(general)
FicheroRed	Obligatorio	<ruta fichero red de neuronas>	Train/Test/Produccion
RutaFicheros	Obligatorio	<ruta ficheros de datos>	Train/Test/Produccion
Tipo	Obligatorio	Clasificacion, Regresion	Train/Test/Produccion
NumCiclosMax	Obligatorio	<valor numérico>	Train
CriterioParada	Obligatorio	MSE, Epocs, CrossValidation	Train
UmbralMSE	Obligatorio (si CriterioParada=MSE)	<valor numérico>	Train
Repeticiones	Opcional	<valor numérico>	Train/Test/Produccion
Multiplicador	Opcional	<valor numérico>	Train
Identificador	Opcional	<cadena de caracteres>	Test/Produccion

Tabla 222. Parámetros de entrada válidos

16 Anexo E: Manual de usuario Simulador

16.1 Arranque de la aplicación

Para iniciar la aplicación basta con hacer doble clic sobre el icono de la aplicación, tal y como muestra la imagen:

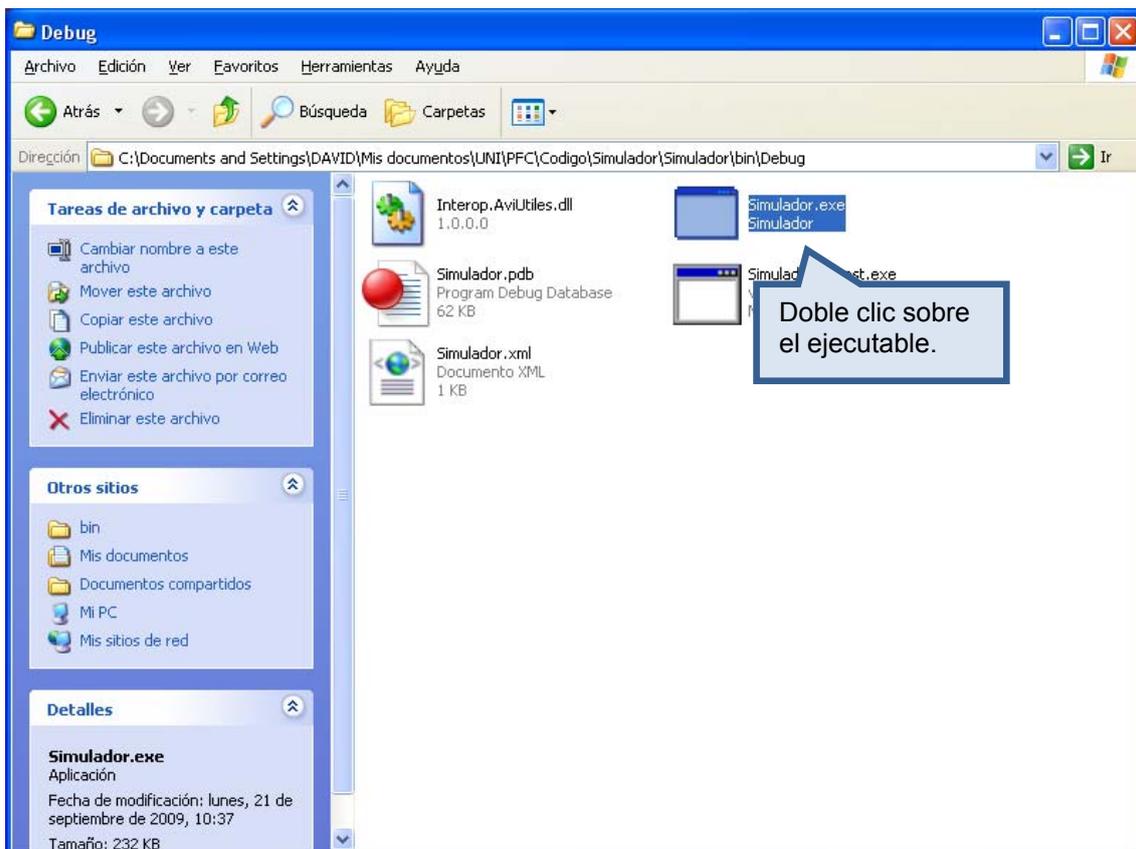


Figura 82. Arranque del Simulador

Una vez iniciada la aplicación el usuario podrá visualizar el interfaz de la aplicación. A continuación se muestra la aplicación recién arrancada:

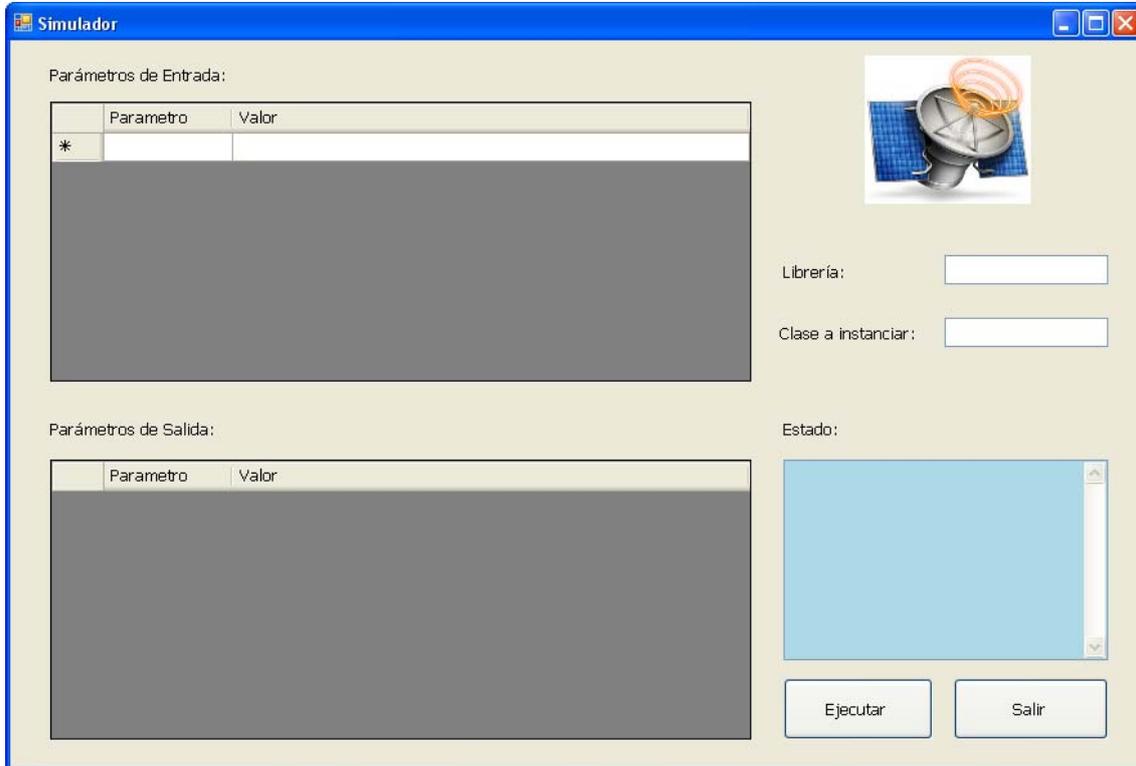


Figura 83. Interfaz del Simulador

16.2 Introducción de parámetros de entrada

Para el correcto funcionamiento del sistema se deben introducir algunos parámetros de entrada que serán enviados al módulo de comunicación desarrollado. Mediante una sencilla tabla, el simulador permite introducir los parámetros de entrada indicando su nombre y su valor. La siguiente imagen muestra un ejemplo:

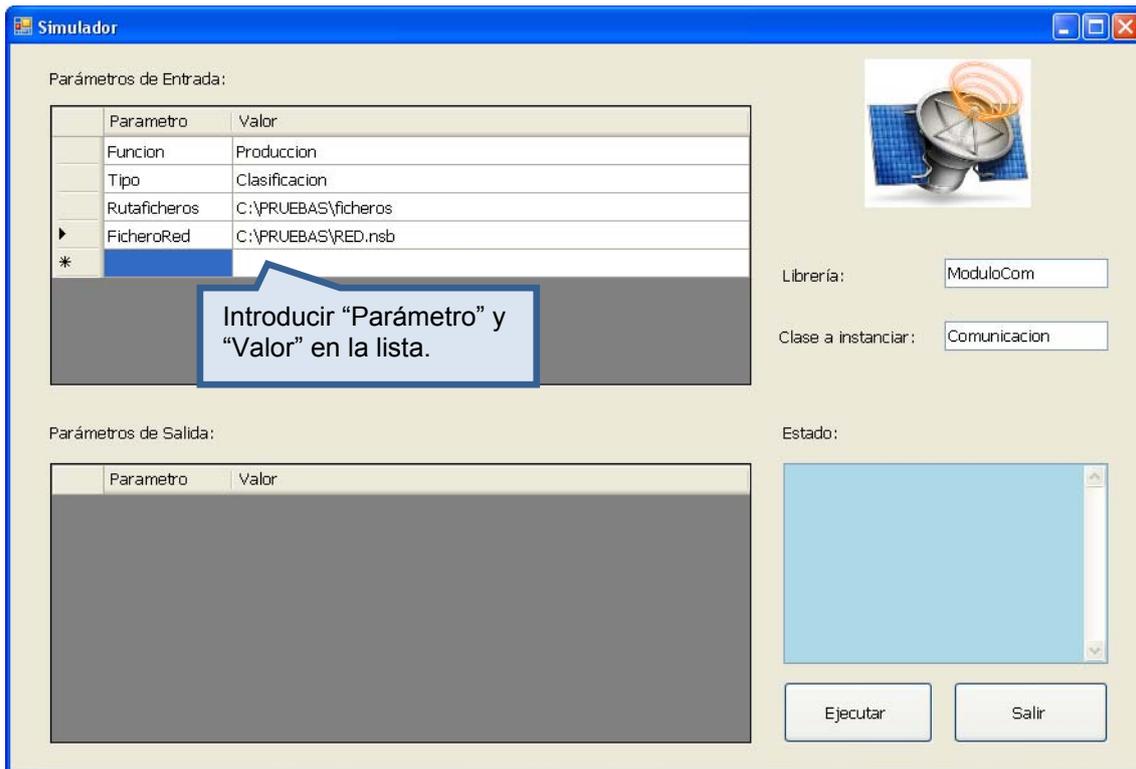


Figura 84. Introducir parámetros de entrada

NOTA: Para conocer los parámetros aceptados por el módulo de comunicación y sus posibles valores ver el apartado 14.3. Comprobación de errores.

16.3 Ejecución del sistema

Una vez introducidos los parámetros de entrada el simulador permite comenzar la ejecución del sistema haciendo clic sobre el botón “Ejecutar”. Previamente se debe haber introducido el nombre de la clase del módulo de comunicación a utilizar en el cuadro de texto “Clase a instanciar”, siendo en este caso *ModuloCom.Comunicacion*.

La siguiente imagen muestra el comienzo de la ejecución:

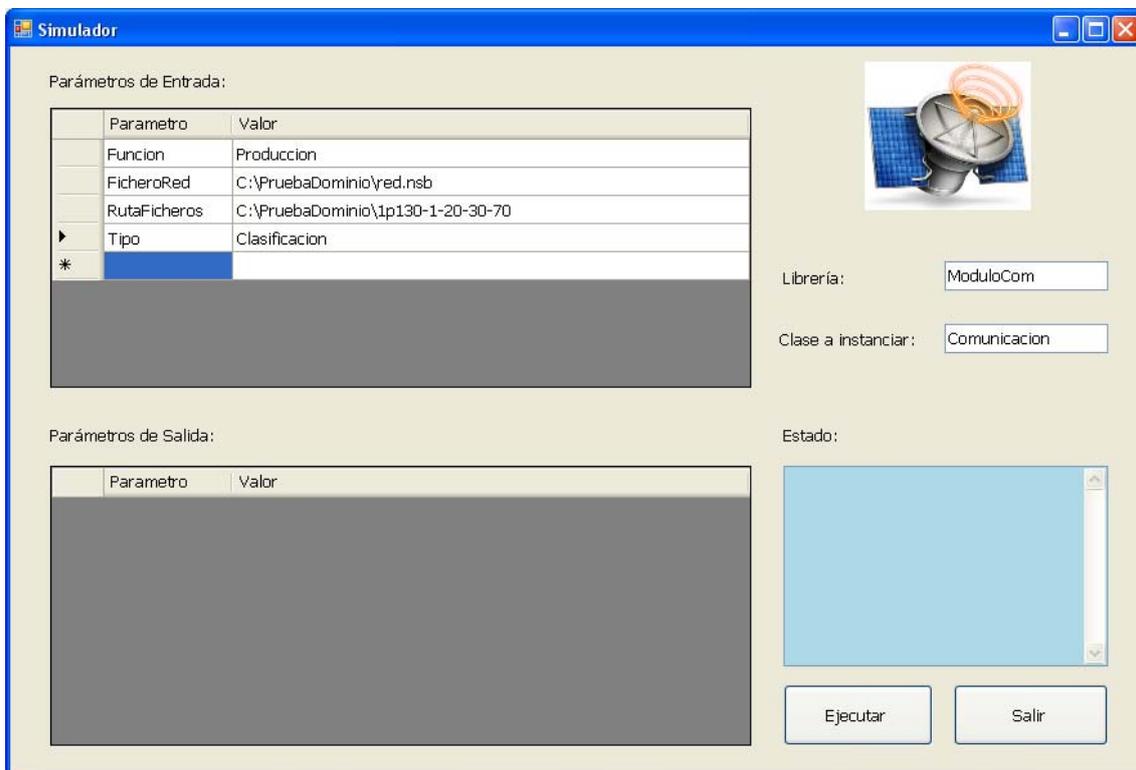


Figura 85. Comienzo ejecución simulador

En la ventana de “Estado” se puede observar el desarrollo del proceso de ejecución. En el momento que éste haya finalizado el simulador lo indicará con el mensaje “Ejecución finalizada”, tal y como se observa a continuación:

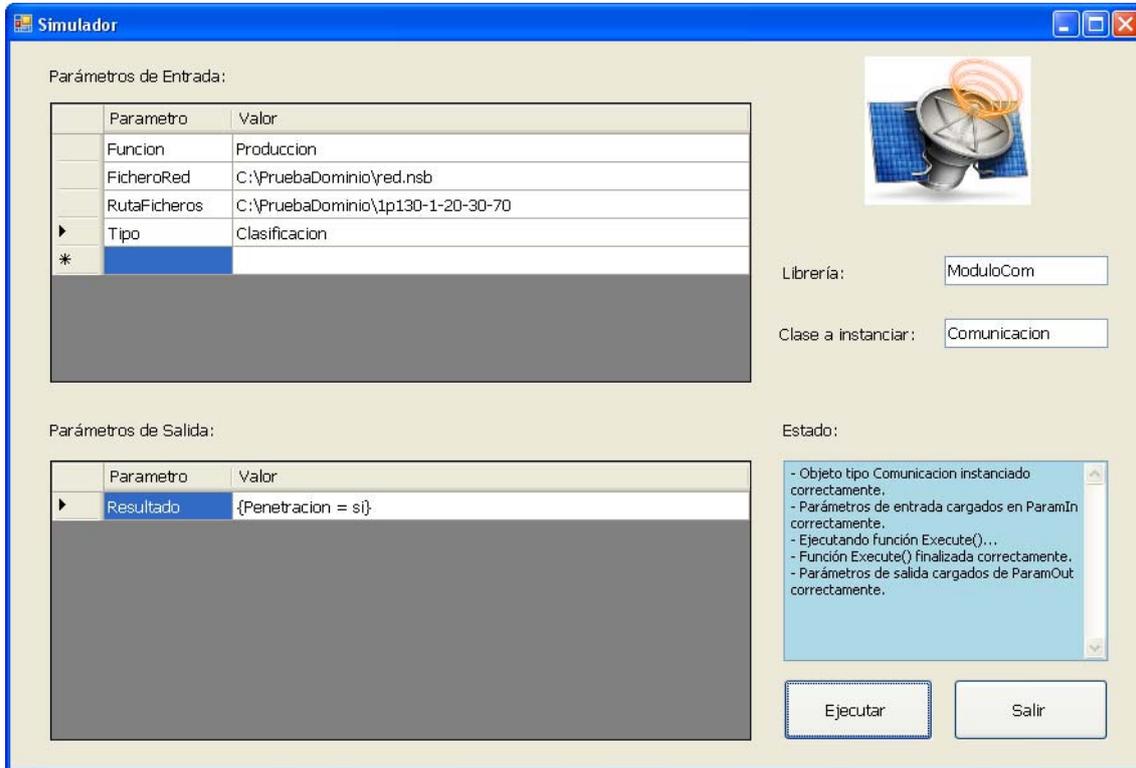


Figura 86. Fin ejecución simulador