

Curso 2006/07
CIENCIAS Y TECNOLOGÍAS/19
I.S.B.N.: 978-84-7756-766-0

VANESA MUÑOZ CRUZ

Diseño e implementación de planificadores instruccionales en sistemas tutoriales inteligentes mediante el uso combinado de metodologías borrosa y multiagente

Directores

**LORENZO MORENO RUIZ
ROSA MARÍA AGUILAR CHINEA**



SOPORTES AUDIOVISUALES E INFORMÁTICOS
Serie Tesis Doctorales

*Para mi Madre,
mis Tíos y Javier,
Con todo mi cariño*

Quiero expresar mi más sincero agradecimiento a todas las personas que han colaborado en la realización este trabajo.

En primer lugar, quiero agradecer al Dr. D. Lorenzo Moreno Ruiz su labor de dirección de esta tesis, su profesionalidad y por abrirme las puertas de la carrera universitaria y contribuir a mi formación docente e investigadora.

A la Dra. Rosa M^a Aguilar China mi especial agradecimiento por sus consejos, su generosidad, su motivación y por toda la ayuda que me ha prestado como codirectora de esta Tesis.

Me gustaría que el Dr. D. Evelio González González, la Dra. Dña. M^a Aurelia Noda Herrera y la Dra. Dña. Alicia Bruno Castañeda supieran lo valiosos que han sido sus conocimientos y su apoyo a lo largo de toda esta Tesis.

A los miembros del Dpto. de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadores de la Universidad de La Laguna: Dr. D. Leopoldo Acosta Sánchez, Dr. D. José Luis Sánchez De la Rosa, Dr. D. José Demetrio Piñeiro Vera, D. Juan Julián Merino Rubio, Dr. D. Alberto Francisco Hamilton Castro, Dr. D. Juan Albino Méndez Pérez, Dr. D. Graciliano Nicolás Marichal Plasencia, Dr. D. José Ignacio Estévez Damas, Dr. D. José Francisco Sigut Saavedra, Dra. Dña. Marta Sigut Saavedra, Dr. D. Roberto Luis Marichal Plasencia, Dra. Dña. Carina Soledad González González, Dra. Dña. Silvia Alayón Miranda, D. Santiago Torres Álvarez, D. Jonay Tomás Toledo Carrillo, D. Jesús Miguel Torres Jorge, D. Germán Carlos González Rodríguez, D. Carlos Alberto Martín Galán, D. Héctor Rebozo Morales, D. Ginés Coll Barbuzano, D. Eladio Hernández Díaz, D. Iván Castilla Rodríguez, D. Pedro Antonio Toledo Delgado, Dña. Hilda Mercedes Sanabria Medina, D. Jonatán Felipe García, y a D. Roberto Muñoz González. Gracias por su ayuda y por ser unos fantásticos compañeros.

Quiero agradecer también a la Asociación Tinerfeña de Trisómicos 21 (ATT21) y a Acamán que hayan confiado en nosotros y nos hayan abierto las puertas para poder utilizar el ITS desarrollado.

Agradecer al Ministerio de Educación, Cultura y Deportes por la financiación de este trabajo con una beca del Programa de Formación de Profesorado Universitario (F.P.U.) con número de referencia AP2002-3850.

Finalmente, a mi familia y a mi marido Javier, gracias por su apoyo incondicional.

ÍNDICE

Introducción	19
Capítulo I: El Ordenador y la Inteligencia Artificial en el Aula: Introducción Histórica y Conceptual	23
1 Introducción	23
2 La Sociedad de la Información	23
3 E-Formación	26
4 Soluciones TIC en el Aula	28
4.1 Aplicaciones educativas.....	29
4.1.1 Contenidos educativos.....	30
4.1.2 Herramientas de apoyo al aprendizaje.....	30
4.1.3 Herramientas de apoyo a la enseñanza	31
4.2 Servicios Básicos Comunes: aplicaciones básicas y servicios de gestión	32
4.3 Portales	32
4.4 Conectividad e Infraestructura	33
5 Ventajas e Inconvenientes del uso del ordenador en el aula	34
5.1 Perspectiva del aprendizaje y de los estudiantes	34
5.2 Perspectiva del profesorado	35
5.3 Perspectiva de los Centros Educativos.....	35
6 Sistema Tutorial Inteligente: Perspectiva Histórica	36
7 Software para educación infantil	41
8 Software educativo para las matemáticas	43
8.1 Revisión de Software Educativo en Matemáticas.....	48
8.2 Revisión de Sistemas Expertos y Tutoriales	57
Capítulo II: ITS para el Refuerzo de los Conceptos Lógicos de Número, Suma y Resta	59
1 Introducción	59
2 Dominio	61
3 Base de Datos	64
4 Interfaz Gráfica.....	74
4.1 Posibles interacciones.....	75

4.2 Aspectos Motivacionales	77
4.3 Agentes Pedagógicos.....	78
4.4 Introducción a los Lenguajes de Marcas y JavaScript	81
4.4.1 SGML	82
4.4.2 DTD (Document Type Definition)	82
4.4.3 HTML (Lenguaje de Marcas de Hipertexto)	83
4.4.4 XML (Lenguaje de Marcas Extensible).....	84
4.4.4.1 XSL y XSLT	85
4.4.4.2 XML Schemas	86
4.4.5 Javascript	88
4.5 Definición de la Plantilla XML	89
4.5.1 Ejemplo de actividad de Pinchar	96
4.5.2 Ejemplo de actividad de Pinchar y Mover	99
4.5.3 Ejemplo de actividad de Hacer Parejas (Unir colecciones)	101
4.5.4 Ejemplo de actividad de tipo Orden.....	103
4.5.5 Ejemplo de tipo de actividad de Cantidad.....	105
5 Generador de Actividades	107
5.1 Pinchar	108
Capítulo III: El Planificador Instruccional	115
1 Introducción	115
1.1 Razonamiento basado en casos.....	115
1.2 Procesos de decisión de Markov	117
1.3 Neuro-fuzzy.....	117
1.4 Redes bayesianas.....	118
1.5 Arquitectura de pizarra	119
1.6 Reglas	119
1.7 Belief vectors.....	120
2 Lógica Borrosa y Sistemas Multiagente	120
3 Teoría borrosa	122
3.1 Conjuntos Borrosos	122
3.2 Números Borrosos.....	123
3.3 Variables lingüísticas	123
3.4 Operaciones con Conjuntos Borrosos	124
3.5 Reglas borrosas si-entonces	126

3.6	El razonamiento borroso	127
3.7	Una regla con un antecedente.....	133
3.8	Una regla con múltiples antecedentes	134
3.9	Múltiples reglas con múltiples antecedentes.....	136
3.10	Sistemas de inferencia borrosa	139
3.10.1	Modelo Borroso de Mamdani.....	140
4	Teoría de sistemas multiagente	143
4.1	Sistemas Multiagente.....	145
4.1.1	KQML (Knowledge Query Manipulation Language)	146
4.1.2	FIPA-ACL (FIPA Agent Communication Language)	148
4.1.3	Comparativa entre KQML y FIPA-ACL.....	150
4.1.4	Arquitecturas de Agentes	152
4.1.5	Marcos de trabajo de agentes.....	155
4.1.6	FIPA-OS.....	157
4.1.7	Negociación en Sistemas Multiagente	157
4.1.7.1	Votaciones.....	158
4.1.7.2	Subastas	160
4.1.7.3	Regateo	160
4.1.7.4	Redes de Contrato	161
4.1.7.5	Coaliciones	161
4.1.7.6	Negociación basada en argumentación.	161
5	Planificador instruccional borroso y multiagente.....	162
6	El Planificador Instruccional Borroso	163
6.1	Esquema del Planificador Instruccional Borroso	166
6.2	Entradas y Salidas del Controlador Borroso	167
6.3	Descripción lingüística.....	169
6.4	Base de Reglas	171
6.5	Funciones de pertenencia	174
6.6	Mecanismo de Borrosificación, Inferencia y Desborrosificación.	179
6.7	Simulación del Planificador Instruccional Borroso	180
6.8	Implementación del Planificador Instruccional Borroso y resultados.....	184
7	Planificación multiagente.....	186
7.1	Cálculo de la Función Objetivo	188
7.1.1	Ajuste de Pesos de la Función Objetivo	189
7.2	Cálculo de la Función Actividad	193

Capítulo IV: Verificación y Validación	203
1 Introducción	203
2 Métodos de verificación.....	205
2.1 Métodos de programación estructurada.....	205
2.2 Testeo	206
2.3 Simulación	206
2.4 Verificación de modelos analíticos	207
3 Métodos de validación	207
3.1 Métodos de Validación Cuantitativa.....	208
3.2 Métodos de Validación Cualitativa.....	210
3.2.1 Validación Superficial	210
3.2.2 Pruebas de turing	211
3.2.3 Test de Campo	211
3.2.4 Validación de Subsistemas.....	212
3.2.5 Análisis de Sensibilidad	212
3.2.6 Grupos de control	212
3.2.7 Usabilidad del Sistema	213
3.2.7.1 Métodos heurísticos	213
3.2.7.2 Métodos Subjetivos.....	214
3.2.7.3 Métodos Empíricos.....	215
4 Verificación del ITS	216
4.1 Casos de Uso	219
4.1.1 Calcular Nivel.....	219
4.1.2 Calcular Objetivo	219
4.1.3 Calcular Tipo de Actividad.....	220
4.1.4 Mostrar Actividad.....	220
4.1.5 Insertar Alumno	221
4.1.6 Modificar Alumno.....	222
4.1.7 Eliminar Alumno	223
4.1.8 Consultar Resultados	223
5 Validación del ITS	224
5.1 Validación cuantitativa	224
5.1.1 Validación Cuantitativa del Sistema Borroso.....	225
5.1.2 Validación Cuantitativa del Sistema MultiAgente	226
5.1.2.1 Validación Cuantitativa del Sistema MASObj.....	227

5.1.2.2 Validación Cuantitativa del Sistema MASAct	228
5.2 Validación cualitativa	229
5.2.1 Usabilidad de la Interfaz Gráfica	230
5.2.2 Validación Cualitativa del ITS	230
Conclusiones	241
Apéndice A	245
1 Ejercicios Fase 1 (Lógica).....	245
1.1 Clasificación	245
1.1.1 Nivel Bajo	245
1.1.1.1 Actividad Tipo 1.....	245
1.1.1.2 Actividad Tipo 2.....	246
1.1.2 Nivel Alto.....	246
1.1.2.1 Actividad Tipo 3.....	246
1.1.2.2 Actividad Tipo 4.....	247
1.2 Relaciones de Orden	248
1.2.1 Nivel Bajo	248
1.2.1.1 Actividad Tipo 1.....	248
1.2.1.2 Actividad Tipo 2.....	248
1.2.2 Nivel Alto.....	249
1.2.2.1 Actividad Tipo 3.....	249
1.3 Correspondencia término a término	251
1.3.1 Nivel Bajo	251
1.3.1.1 Actividad Tipo 1.....	251
1.3.2 Nivel Alto.....	251
1.3.2.1 Actividad Tipo 2.....	251
1.4 Cuantificadores.....	252
1.4.1 Nivel Bajo	252
1.4.1.1 Actividad Tipo 1.....	252
1.4.1.2 Actividad Tipo 2.....	253
1.4.1.3 Actividad Tipo 3.....	254
1.4.1.4 Actividad Tipo 4.....	254
1.4.2 Nivel Alto.....	255
1.4.2.1 Actividad Tipo 1.....	255
1.4.2.2 Actividad Tipo 2.....	256

1.4.2.3	Actividad Tipo 3.....	257
1.4.2.4	Actividad Tipo 4.....	257
2	Ejercicios Fase 2 (Números y Operaciones)	258
2.1	Contar	258
2.1.1	Nivel Bajo	258
2.1.1.1	Actividad Tipo 1.....	258
2.1.1.2	Actividad Tipo 2.....	259
2.1.1.3	Actividad Tipo 4.....	260
2.1.2	Nivel Medio	260
2.1.2.1	Actividad Tipo 1.....	260
2.1.2.2	Actividad Tipo 2.....	261
2.1.2.3	Actividad Tipo 3.....	262
2.1.2.4	Actividad Tipo 4.....	262
2.1.3	Nivel Alto	263
2.1.3.1	Actividad Tipo 1.....	263
2.1.3.2	Actividad Tipo 2.....	264
2.1.3.3	Actividad Tipo 3.....	264
2.1.3.4	Actividad Tipo 5.....	265
2.1.3.5	Actividad Tipo 6.....	266
2.2	Reconocimiento del número.....	266
2.2.1	Nivel Bajo	266
2.2.1.1	Actividad Tipo 2.....	266
2.2.1.2	Actividad Tipo 3.....	267
2.2.1.3	Actividad Tipo 4.....	268
2.2.2	Nivel Medio	268
2.2.2.1	Actividad Tipo 2.....	268
2.2.2.2	Actividad Tipo 3.....	269
2.2.2.3	Actividad Tipo 4.....	270
2.2.3	Nivel Alto	270
2.2.3.1	Actividad Tipo 2.....	270
2.2.3.2	Actividad Tipo 3.....	271
2.2.3.3	Actividad Tipo 4.....	272
2.3	Cardinalidad	272
2.3.1	Nivel Bajo	272
2.3.1.1	Actividad Tipo 2.....	272

2.3.1.2	Actividad Tipo 3.....	273
2.3.1.3	Actividad Tipo 4.....	274
2.3.1.4	Actividad Tipo 5.....	274
2.3.1.5	Actividad Tipo 6.....	275
2.3.1.6	Actividad Tipo 7.....	276
2.3.1.7	Actividad Tipo 8.....	277
2.3.2	Nivel Medio	277
2.3.2.1	Actividad Tipo 2.....	277
2.3.2.2	Actividad Tipo 3.....	278
2.3.2.3	Actividad Tipo 4.....	279
2.3.2.4	Actividad Tipo 5.....	280
2.3.2.5	Actividad Tipo 6.....	280
2.3.2.6	Actividad Tipo 7.....	281
2.3.2.7	Actividad Tipo 8.....	282
2.3.3	Nivel Alto	282
2.3.3.1	Actividad Tipo 2.....	282
2.3.3.2	Actividad Tipo 3.....	283
2.3.3.3	Actividad Tipo 4.....	284
2.3.3.4	Actividad Tipo 5.....	284
2.3.3.5	Actividad Tipo 6.....	285
2.3.3.6	Actividad Tipo 7.....	286
2.3.3.7	Actividad Tipo 8.....	286
2.3.3.8	Actividad Tipo 10.....	287
2.4	Orden	288
2.4.1	Nivel Bajo	288
2.4.1.1	Actividad Tipo 1.....	288
2.4.1.2	Actividad Tipo 3.....	288
2.4.1.3	Actividad Tipo 4.....	289
2.4.1.4	Actividad Tipo 5.....	290
2.4.1.5	Actividad Tipo 6.....	291
2.4.2	Nivel Medio	292
2.4.2.1	Actividad Tipo 1.....	292
2.4.2.2	Actividad Tipo 2.....	292
2.4.2.3	Actividad Tipo 3.....	293
2.4.2.4	Actividad Tipo 4.....	294

2.4.2.5	Actividad Tipo 5.....	295
2.4.2.6	Actividad Tipo 6.....	295
2.4.3	Nivel Alto.....	296
2.4.3.1	Actividad Tipo 1.....	296
2.4.3.2	Actividad Tipo 2.....	297
2.4.3.3	Actividad Tipo 3.....	297
2.4.3.4	Actividad Tipo 4.....	298
2.4.3.5	Actividad Tipo 5.....	299
2.4.3.6	Actividad Tipo 6.....	299
2.5	Ordinalidad.....	300
2.5.1	Nivel Bajo.....	300
2.5.1.1	Actividad Tipo 1.....	300
2.5.1.2	Actividad Tipo 2.....	301
2.5.1.3	Actividad Tipo 3.....	302
2.5.1.4	Actividad Tipo 4.....	303
2.5.2	Nivel Medio.....	303
2.5.2.1	Actividad Tipo 1.....	303
2.5.2.2	Actividad Tipo 2.....	304
2.5.2.3	Actividad Tipo 3.....	305
2.5.2.4	Actividad Tipo 4.....	305
2.5.3	Nivel Alto.....	306
2.5.3.1	Actividad Tipo 1.....	306
2.5.3.2	Actividad Tipo 2.....	307
2.5.3.3	Actividad Tipo 3.....	307
2.5.3.4	Actividad Tipo 4.....	308
2.6	Problemas.....	309
2.6.1	Nivel Bajo.....	309
2.6.1.1	Actividad Tipo 1.....	309
2.6.1.2	Actividad Tipo 2.....	311
2.6.2	Nivel Medio.....	311
2.6.2.1	Actividad Tipo 1.....	311
2.6.2.2	Actividad Tipo 2.....	312
2.6.3	Nivel Alto.....	313
2.6.3.1	Actividad Tipo 1.....	313
2.6.3.2	Actividad Tipo 2.....	314

2.7	Algoritmo	315
2.7.1	Nivel Medio	315
2.7.1.1	Actividad Tipo 3.....	315
2.7.2	Nivel Alto.....	316
2.7.2.1	Actividad Tipo 3.....	316
Apéndice B		317
1	Introducción.....	317
1.1	Objetivos	317
2	Instalación	318
2.1	Requerimientos mínimos de instalación.....	322
3	Inserción de nuevos alumnos	322
3.1	Insertar nuevos alumnos.....	323
3.2	Moverse por las fichas de alumnos	324
3.3	Modificar alumnos	324
3.4	Eliminar alumnos	325
4	Uso de la aplicación.....	325
4.1	Inicio de la aplicación.....	325
4.2	Agentes Pedagógicos	327
4.3	Dominio de la Aplicación	327
4.4	Resultados	331
5	Ejemplos.....	332
5.1	Tipo de Actividad Pinchar	332
5.1.1	Clasificación	332
5.1.2	Relaciones de Orden	333
5.1.3	Correspondencia término a término	333
5.1.4	Cuantificadores	334
5.1.5	Contar.....	335
5.1.6	Cardinalidad.....	335
5.1.7	Ordinalidad	337
5.2	Tipo de Actividad Pinchar y Mover	338
5.2.1	Clasificación	338
5.2.2	Relaciones de Orden	338
5.2.3	Cuantificadores	339
5.2.4	Ordinalidad	340

Apéndice C	341
1. Introducción	341
1.1 Objetivos	341
2 Instalación	342
2.1 Requerimientos mínimos de instalación	342
3 Uso de la aplicación	343
3.1 Inicio de la aplicación	343
3.2 Agentes Pedagógicos	343
3.3 Generación de Actividades	344
3.3.1 Pinchar	346
3.3.2 Arrastrar Auto (Pinchar y Mover)	355
3.3.3 Arrastrar	359
3.3.4 Parejas (Unir colecciones)	360
3.3.5 Ordenar Auto	362
Apéndice D	365
Apéndice E	367
Apéndice F	375
Apéndice G	377
Apéndice H	379
1 Protocolo de Interacción FIPA Propose	379
2 Protocolo de Interacción FIPA Request	379
3 Protocolo de Interacción FIPA Request When	380
4 Protocolo de Interacción FIPA Query	380
5 Protocolo de Interacción FIPA Contract Net	381
6 Protocolo de Interacción FIPA Iterated Contract Net	382
7 Protocolo de Interacción FIPA English Auction	382
8 Protocolo de Interacción FIPA Dutch Auction	383
9 Protocolo de Interacción FIPA Brokering	384
10 Protocolo de Interacción FIPA Recruiting	385
11 Protocolo de Interacción FIPA Subscribe	386
Apéndice I	387
Bibliografía	389

Introducción

La presente Tesis Doctoral se enmarca en una línea de Investigación desarrollada por miembros del Departamento de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadores de la Universidad de La Laguna, en colaboración con miembros del Departamento de Análisis Matemático (área de Didáctica de las Matemáticas) de la Universidad de La Laguna. Ha sido subvencionada por una beca y un contrato en el marco del estatuto del personal investigador en formación, del Programa de Formación de Profesorado Universitario (F.P.U.) con número de referencia AP2002-3850.

En los últimos años la presencia de las nuevas tecnologías en la sociedad es cada vez mayor ya que proporciona gran cantidad de medios para acceder, transmitir e interpretar la información. Se ha demostrado que el uso del ordenador y el software educativo en el aula favorecen la motivación y la integración en la sociedad del alumnado de los distintos niveles educativos. Pero dicho software educativo debería considerar las características evolutivas y de aprendizaje del alumno proponiendo ejercicios y prácticas adecuadas a su desarrollo. Sin embargo, una gran mayoría de los programas existentes en el mercado no se adaptan a las características o la evolución del alumno. Para conseguir dicho fin esta Tesis Doctoral consiste en el Diseño e Implementación de un Sistema Tutorial Inteligente (ITS) para la enseñanza que combina el uso de metodologías Borrosa y Multiagente para la realización de un Planificador Instruccional que se adapte a las características de los estudiantes. Aunque es generalizable a cualquier dominio se ha realizado como aplicación concreta una herramienta de contenido educativo que sirva de apoyo al aprendizaje y refuerzo de los conceptos lógicos de reconocimiento del número, suma y resta.

En la primera fase de este trabajo se realiza una revisión de algunos programas de software educativo (tanto comerciales como libres) que comprenden juegos, herramientas para el aprendizaje de las matemáticas, etc. para los niveles educativos de infantil, primaria y secundaria. Posteriormente gracias a la colaboración de miembros del área de Didáctica de las Matemáticas se realiza la adquisición, análisis y modelado del conocimiento pedagógico necesario para la

realización del ITS. Con este conocimiento pedagógico se desarrolla en primer lugar la Interfaz Gráfica. Ésta consiste en plantillas XML que permiten una interfaz con tecnología web. Las múltiples instanciaciones de las plantillas generarán el conjunto de actividades a presentar al estudiante.

En la segunda fase se desarrolla el Planificador Instruccional siguiendo las metodologías borrosa y multiagente. La metodología borrosa se ha utilizado teniendo en cuenta las características de incertidumbre del conocimiento pedagógico a modelar, ya que el comportamiento del experto profesor viene dado por una serie de reglas que a menudo son imprecisas. Por otro lado la utilización de un Sistema Multiagente es muy beneficiosa pues provee de un entorno interactivo de gran capacidad de reacción que permite distribuir las tareas propuestas al alumno siguiendo los criterios de los profesores y todo ellos sin disminuir el rendimiento del ITS, ya que los agentes trabajan de forma concurrente. Por último se ha realizado la verificación y la validación del Sistema Tutorial Inteligente necesaria para la credibilidad por parte del usuario de la herramienta propuesta.

La presente memoria desarrolla las dos fases que se han comentado, para lo cual ha sido dividida en cuatro capítulos.

En el Capítulo I se realiza una introducción al uso de las nuevas tecnologías en la sociedad actual, incluyéndose el concepto de e-formación y las ventajas e inconvenientes que presentan el uso del ordenador en el aula. Tras presentar estos tópicos se introducen los conceptos de Inteligencia Artificial, centrándose en la presentación de los Sistemas Tutoriales Inteligentes. Por último se hace una revisión del software educativo para la enseñanza de las matemáticas existente, incluyéndose Sistemas Expertos y Tutoriales.

En el Capítulo II se desarrolla el diseño e implementación de la Interfaz Gráfica del Sistema Tutorial Inteligente. En primer lugar se define la estructura del ITS para a continuación pasar a determinar los objetivos de aprendizaje del ITS. El dominio de enseñanza elegido para el mismo son las matemáticas y más concretamente los conceptos lógicos de reconocimiento del número, suma y resta. A continuación el capítulo se centra en diseño de la Interfaz Gráfica incluyendo aspectos motivacionales, para posteriormente definir como se ha llevado a cabo la implementación de las plantillas XML con las que se han generado las diferentes actividades incluidas en el ITS.

En el Capítulo III se diseña el Planificador Instruccional Borroso Multiagente que permitirá la adaptabilidad del ITS permitiendo que a partir de un alumno concreto se fijen los objetivos de aprendizaje para el mismo y se realice la secuencia de acciones necesaria para cumplirlos. Inicialmente se presenta una revisión de los diferentes métodos que han sido empleados en la bibliografía en el diseño de Planificadores Instruccionales, y a continuación una presentación de las metodologías Borrosa y Multiagente puesto que son los mecanismos que se han utilizado en este trabajo. Posteriormente se especifica el diseño y la implementación del Planificador Instruccional usando ambas metodologías.

Finalmente en el Capítulo IV se realiza la verificación y validación del ITS desarrollado mostrándose los resultados obtenidos con la utilización del mismo por parte de los alumnos de la Asociación Tinerfeña de Trisómicos 21 (ATT21) que están integrados en centros de infantil y primaria siguiendo el currículo de diferentes niveles educativos.

La memoria concluye con nueve apéndices. En el primero se incluye un ejemplo de todos los tipos de actividades que componen el Tutorial. Estas actividades han sido diseñadas por los expertos en Didáctica de las Matemáticas y constituyen la base para el refuerzo de los conceptos lógicos, de número, suma y resta. Se ha propuesto como metodología que los alumnos realicen en primer lugar estas actividades en papel. El siguiente paso en el proceso de aprendizaje es realizar estas actividades en el ordenador como páginas web. Finalmente el ITS permite afianzar todo lo aprendido. Esta metodología permite un acercamiento progresivo del alumno y el profesor al ordenador. El segundo apéndice es el manual del ITS que incluye los asistentes de la base de datos. El tercero presenta el Generador de Actividades que se ha implementado para facilitar la creación de actividades para el ITS. La manera de enriquecer el ITS es aumentar en número de actividades a realizar por el alumno. Para ello se requiere una búsqueda e integración de elementos multimedia (imagen, sonido, letras, etc.). Se ha implementado un asistente que facilita la implementación de los distintos tipos de actividades. La tarea del profesor es diseñar la actividad, buscar los elementos multimedia necesarios y finalmente rellenar la plantilla XML con ayuda del asistente. El cuarto y el quinto apéndice hacen referencia a la programación de las plantillas de la interfaz gráfica siendo respectivamente el XML Schema de las mismas y un ejemplo de XSLT. El sexto, séptimo y octavo apéndices muestran aspectos concretos de las

especificaciones KQML y FIPA (performativas y protocolos de interacción de agentes inteligentes). En el último apéndice se presentan simulaciones del Sistema Multiagente.

CAPÍTULO I

EL ORDENADOR Y LA INTELIGENCIA ARTIFICIAL EN

EL AULA:

INTRODUCCIÓN HISTÓRICA Y CONCEPTUAL

1 INTRODUCCIÓN

Las Tecnologías de la Información y Comunicación (TIC), son un conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento y comunicación en forma de voz, imágenes y datos que están contenidos en señales de naturaleza acústica, óptica o electromagnética. Dichas tecnologías son una de las manifestaciones más relevantes de la globalización ya que han permitido llevar a la globalidad del mundo la comunicación, facilitando la interconexión entre las personas e instituciones a nivel mundial, y eliminando barreras espaciales y temporales. Además, se ha demostrado que la tecnología y especialmente el uso del ordenador favorece la integración educativa y social (sobre todo en alumnos con discapacidades). Es por ello que el uso de herramientas informáticas debería ser considerado por los profesores como un recurso didáctico importante y emplearse en el aula junto con libros y otros materiales más tradicionales.

2 LA SOCIEDAD DE LA INFORMACIÓN

La proliferación y usos de las tecnologías electrónicas – informática y telecomunicaciones – ha generado desde hace ya algún tiempo tales expectativas

que han determinado un nuevo paradigma social que con el nombre de Sociedad de la Información se está imponiendo en los medios políticos y sociales de los países industriales con extraordinaria fortuna mediática.

Tal Sociedad de la Información sería aquella que disponiendo de medios tecnológicos avanzados y de carácter evolutivo, los cuales soportan y facilitan el tratamiento y transmisión de la información, los usa instrumental y creativamente en los procesos intelectuales, productivos y sociales a su mejor conveniencia, con objeto de incrementar el progreso científico, la innovación tecnológica, la productividad y el crecimiento económico, y en última instancia las condiciones de vida de la gente [SI 2003].

Para que la interrelación entre la tecnología y sus usos humanos alcance los mayores logros económicos y sociales posibles, deben cumplirse dos condiciones; una necesaria y otra suficiente. La sociedad necesita estar dotada de medios tecnológicos de cuantía, extensión y coste adecuados, de suerte que, de ser posible, debieran ser accesibles a todos los individuos de la sociedad. Pero, como la tecnología que no se usa carece de utilidad económica y social, es preceptivo educar e incentivar a la gente para el uso de las nuevas tecnologías.

La Sociedad de la Información es una gran oportunidad vinculada al supuesto de que la mayoría de la gente se encuentre conectada con sus ordenadores y otros dispositivos electrónicos a las redes de telecomunicaciones, y más específicamente a Internet, para de esta manera intercambiar información y conocimiento. Esta posibilidad abarca desde la información más elemental a las aplicaciones científicas y profesionales más sofisticadas, pasando por el entretenimiento y la mera comunicación circunstancial.

Para acceder a los contenidos de esta nueva Sociedad de la Información es preceptivo estar conectados a la red, disponer de ordenadores y, sobre todo, saber utilizarlos debidamente [IRT 2001].

Para disfrutar de las oportunidades que Internet ofrece, es preciso afrontar un triple reto: aumentar la penetración social de las redes de telecomunicaciones, difundir el empleo de los ordenadores y educar a la gente en el uso de las nuevas tecnologías.

La universalización del servicio de telecomunicaciones ha venido siendo una ambición política. En la Unión Europea el servicio universal formó parte de las

obligaciones regulatorias de los gobiernos, de acuerdo con las recomendaciones de la Comisión.

Actualmente, nadie se atrevería a poner en duda la importancia de la sociedad de la información y su repercusión, no sólo en la economía, sino también en la sociedad en general. Además, pocos discutirían la necesidad de avanzar en el desarrollo de medidas del conjunto de fenómenos que abarca tal concepto. En esta línea, el *Plan de Acción e-Europe 2005* continúa con la labor desarrollada por su antecesor (*Plan e-Europe 2002*), y para ello busca el desarrollo de un entorno seguro de servicios, aplicaciones y contenidos, que esté basado en una infraestructura de banda ancha que sea fácilmente disponible para todos los agentes. Fue presentado por la Comisión en Mayo de 2002, a la vista del Consejo Europeo de Sevilla, y estableció acciones políticas (tanto para las Instituciones Europeas como para los Estados Miembros) para acelerar el desarrollo de la Sociedad de la Información en Europa.

El *Plan de acción e-Europe 2010* [EIS 2006] es el sucesor de los planes del 2002 y 2005. En él la Comisión Europea propone un nuevo marco estratégico, llamado i2010. En dicho marco se promueve una economía digital abierta y competitiva y se hace hincapié en las TIC considerándolas como impulsoras para la mejora de la calidad de vida. En este marco, por tanto, se trabajará por un enfoque integrado de las políticas de sociedad de la información y medios audiovisuales en la UE. Para ello la Comisión propone tres prioridades para las políticas europeas de sociedad de la información y medios de comunicación:

1. La construcción de un Espacio Único Europeo de la Información que promueva un mercado interior abierto y competitivo para la sociedad de la información y los medios de comunicación.
2. El refuerzo de la innovación y la inversión en la investigación sobre las TIC con el fin de fomentar el crecimiento y la creación de más empleos y de más de calidad.
3. El logro de una sociedad europea de la información basada en la inclusión que fomenta el crecimiento y el empleo de una manera coherente con el desarrollo sostenible y que da la prioridad a la mejora de los servicios públicos y de la calidad de vida.

Además la Comisión efectuará un seguimiento de la asimilación de los servicios de banda ancha, de los servicios de negocios electrónicos y administración electrónica, de la inversión en la investigación sobre TIC, de las disparidades económicas y sociales y de la alfabetización digital, mediante informes de situación.

3 E-FORMACIÓN

El aprendizaje electrónico (*e-learning*) permite complementar y reforzar la enseñanza tradicional mediante el uso de las TIC (incluyendo conexión a Internet o a alguna red de comunicaciones). Hay que tener en cuenta que el simple hecho de tener un ordenador conectado a Internet no implica por sí solo que se dé aprendizaje electrónico: para que realmente exista este aprendizaje se ha de contar con material educativo y software apropiado.

Para promover la inclusión de las TIC en la enseñanza es necesario adaptar el sistema educativo en contenidos y metodologías tanto en enseñanza obligatoria en primaria y secundaria como en los ciclos formativos superiores (medida incluida en la fase 2 del ámbito de colegios, de la Contribución Española a la Cumbre Mundial de la Sociedad de la Información (CMSI)) [Telefónica 2003]. Además para fomentar el uso de las TIC, es necesario que los colegios dispongan de al menos un ordenador conectado a Internet con fines educativos según la medida incluida en las fases 1 y 2 del ámbito de colegios para el *e-learning* de la Contribución Española a la Cumbre.

En el caso español esta incorporación de las TIC al currículo se ha dado en varias fases:

- En la primera fase, llamada fase exploratoria realizada con un número reducido de colegios se crean las infraestructuras necesarias para facilitar el acceso a las TIC y su uso (dotación de red, hardware y software) siendo necesario además el mantenimiento de las mismas. Además, para

el buen funcionamiento de este proyecto es necesaria la formación en las TIC del profesorado y la creación de un coordinador del equipo docente participante en el plan de integración de las TIC. Para comprobar los resultados obtenidos en esta fase se crea un Gabinete de seguimiento y se realiza una evaluación interna bajo la dirección del Ministerio en la que participan los formadores y los profesores de los centros. En cuanto a la evaluación externa de este proyecto hay varias realizadas por Universidades Nacionales y por la OCDE a nivel internacional [Alonso 2002].

- En la siguiente fase (fase de extensión) se aumenta el número de escuelas incorporadas al proyecto mediante un concurso público.
- Por último la fase de generalización permite ampliar el proyecto a todas las escuelas y proporcionar apoyo para la implantación del aprendizaje electrónico y el desarrollo de software educativo.

Como ya se ha comentado para que este proyecto de adaptación curricular funcione es necesario formar al profesorado mediante cursos sobre aprendizaje electrónico y hacer que los profesores se sientan motivados con el uso de herramientas TIC. Es necesario por tanto que los profesores aprendan a valorar la pedagogía mediante el uso de las TIC y vean las ventajas de la utilización de contenidos digitales y tecnologías multimedia. Hay que tener en cuenta que dichos contenidos pueden ser compartidos entre los profesores de diferentes centros, reutilizados o adaptados en función de las necesidades. El intercambio de contenidos entre los profesores puede estimular la creatividad de los docentes debido al intercambio de experiencias e ideas.

Otra línea de la e-formación consiste en la impartición de cursos de uso de las TIC al personal de las empresas, ya que esto permite que los trabajadores amplíen su formación por medios electrónicos y garantiza que los profesionales dispongan de los conocimientos necesarios para la utilización de herramientas adecuadas, que puedan facilitar su trabajo y mejorar los servicios ofrecidos por las empresas. Además este tipo de formación garantiza que los trabajadores estén familiarizados con las nuevas tecnologías a medida que estas evolucionen (considerando la rápida evolución que se da sobre todo en el caso del hardware y el software).

Por último, mencionar que las nuevas tecnologías permiten el acceso a la información y la formación e integración de desempleados, jubilados y colectivos con necesidades especiales como por ejemplo con discapacidades físicas y psíquicas [Cebrián 2000].

4 SOLUCIONES TIC EN EL AULA

Con el fin de situar esta Tesis Doctoral dentro de todo el desarrollo de la Sociedad de la Información, vamos a presentar en este apartado un retrato del uso de las TIC en la Educación.

En la figura 1-1 se representa un esquema que clasifica las principales soluciones TIC que apoyan a alumnos, profesores y gestores educativos en los procesos de enseñanza-aprendizaje que los relaciona. Se trata de una clasificación de referencia para conocer las posibilidades que ofrecen las nuevas tecnologías y su aplicación dependerá de las necesidades de cada caso.

Se entiende por educación digital tanto a la educación presencial como a distancia que hace uso de tecnologías digitales y que tiene como objetivo la adquisición de competencias para aprender a aprender, tanto por parte de profesores como de estudiantes, en un proceso de formación permanente. Vamos a ir presentando cada uno de los elementos necesarios en este proceso de educación digital.

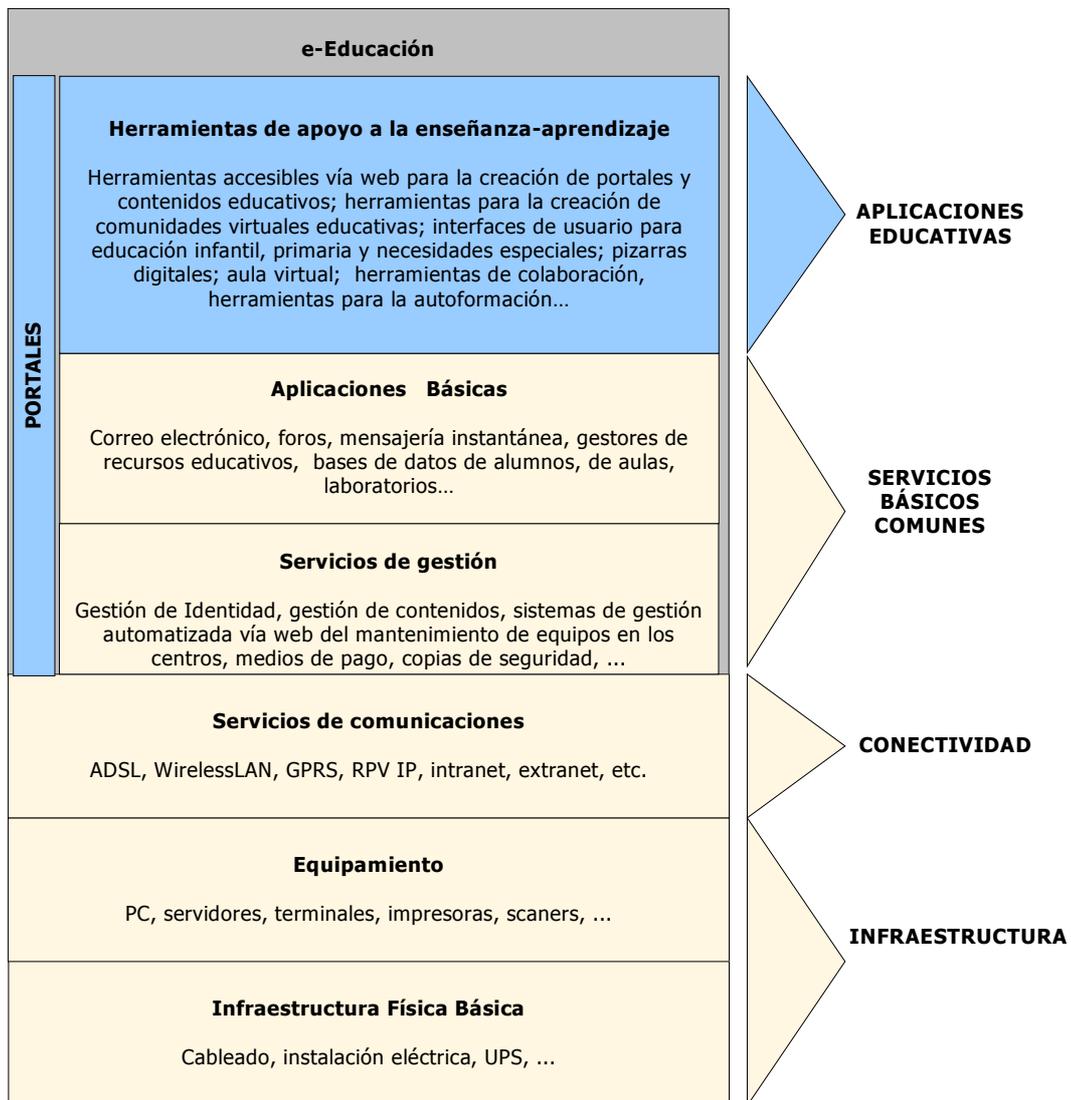


Figura 1-1.- Soluciones TIC para la educación

4.1 Aplicaciones educativas

Dentro de este apartado encontramos todas las herramientas de apoyo al aprendizaje, de apoyo a la enseñanza o de desarrollo de contenido educativo. A continuación las describimos brevemente.

4.1.1 CONTENIDOS EDUCATIVOS

El desarrollo de contenidos educativos digitales es de gran importancia para la plena incorporación de las TIC a los procesos educativos. En esta creación de contenidos no es suficiente con la mera digitalización del material educativo habitual, sino que será preciso adaptarlo específicamente al soporte utilizado. En esta labor cobra especial importancia el papel de los profesores, que en algunos casos serán los que realicen este trabajo, de ahí la importancia de facilitarles herramientas con las que puedan construirlos, y las editoriales, que serán las encargadas de generar los contenidos de carácter general.

El éxito de la incorporación de las TIC a la educación depende en gran medida de la calidad de los contenidos educativos. El desarrollo de contenidos se complementa con herramientas que facilitan la autoevaluación, y los juegos y simuladores, ideados para facilitar la comprensión de conceptos.

El objetivo fundamental de esta Tesis Doctoral es el Diseño e Implementación de un Sistema Tutorial Inteligente como una herramienta de contenido educativo que sirva de apoyo al aprendizaje (concretamente al aprendizaje de los conceptos lógicos, reconocimiento del número, suma y resta).

4.1.2 HERRAMIENTAS DE APOYO AL APRENDIZAJE

En este grupo se encuentran aquellas aplicaciones que hacen posible el establecimiento de nuevos modelos de aprendizaje o soportan modelos ya existentes. El ejemplo más representativo es el conjunto de aplicaciones para la comunicación, como el correo electrónico, el chat, los programas de mensajería instantánea, videoconferencia y los foros y herramientas de soporte a grupos virtuales.

Este tipo de herramientas puede facilitar el aprendizaje cooperativo en red, el establecimiento de comunicación entre personas que realizan el mismo tipo de investigación y cuya comunicación anteriormente resultaba complicada.

Favorecen además la comunicación no sólo entre alumnos, sino entre alumnos y profesores, agilizando las tutorías y también la comunicación entre profesores y padres.

Los procesos de interacción que se producen entre los diferentes actores del sistema educativo son el elemento esencial del proceso de aprendizaje. Normalmente las personas aprenden cuando ponen en común con profesores y otros alumnos sus dudas o aportan soluciones a las mismas.

4.1.3 HERRAMIENTAS DE APOYO A LA ENSEÑANZA

Dentro de este grupo se encuentran aquellas herramientas que facilitan la labor docente. Entre ellas destacan:

- Las herramientas que facilitan la creación de portales y contenidos educativos por parte de profesores y pedagogos. Los profesores necesitan de este tipo de herramientas para digitalizar los contenidos de sus clases y adaptarlos a Internet como un nuevo recurso que pueden utilizar en sus métodos de enseñanza.
- Las herramientas para la creación de comunidades virtuales educativas.
- Las interfaces de usuario para educación infantil, primaria y necesidades especiales. Gracias a ellas es posible adaptar el medio a las necesidades de los alumnos. Por ejemplo, en el caso de una persona con una minusvalía visual es posible representar la información con un tipo de letra mayor o incluso apoyarse en sistemas de reconocimiento y síntesis de voz.
- La pizarra digital (pizarra electrónica, kit de Internet en el aula) es un sistema tecnológico que consiste básicamente en un ordenador multimedia conectado a Internet y un videoprojector. Gracias a esta pizarra, profesores y alumnos tienen permanentemente a su disposición un sistema para visualizar y comentar de manera colectiva toda la información que puede proporcionar Internet, la televisión o cualquier otra fuente de que dispongan en cualquier formato: presentaciones multimedia y documentos digitalizados en disco (apuntes, trabajos de clase...), vídeos, documentos en papel (que pueden capturar con una simple webcam), etc.

- Aula virtual: consiste en un sistema que facilita la conexión por videoconferencia con otras aulas en las que se imparte o recibe formación. Esta conexión implica tanto la visualización del profesor, como la de todo el material que éste está presentando. Por otro lado, facilita la comunicación entre las diferentes aulas presenciales que se conectan a través de este sistema. De esta forma es posible compartir recursos entre diferentes centros, independizándose el lugar de formación de la formación en sí misma.

4.2 Servicios Básicos Comunes: aplicaciones básicas y servicios de gestión

Dentro de este grupo se encuentran aquellas herramientas que ofrecen soporte a la gestión de los recursos educativos, como por ejemplo:

- Las aplicaciones que gestionan las bases de datos de alumnos, con información académica y que facilitan el seguimiento de los mismos.
- Las aplicaciones que gestionan las BBDD de recursos de los centros, con información sobre horarios, profesores, disponibilidad de aulas, etc.
- Las aplicaciones que gestionan contenidos educativos y de apoyo. Se trata de bases de datos que almacenan contenidos educativos generados por los profesores o desarrollados por otros agentes, material complementario, bibliografías, revistas, etc.
- Las herramientas que facilitan la gestión de los equipos informáticos en los centros.

4.3 Portales

Como elemento vertical que se relaciona con todos y cada uno de los niveles descritos anteriormente, los portales de información facilitan el acceso a las

herramientas y contenidos desde una interface web, ya sea desde una intranet o desde Internet.

4.4 Conectividad e Infraestructura

La disposición de las infraestructuras de comunicaciones e informática es el primer paso que debe dar la comunidad educativa para poder acceder al resto de soluciones TIC. Los elementos que forman parte de la misma son:

- Conexiones de banda ancha para cada uno de los centros educativos, gracias a las cuales es posible conectar a los diferentes centros entre sí y a todos ellos con Internet. La tecnología de acceso utilizada en cada caso dependerá de las necesidades de dichos centros, pero en cualquier caso deberá ser compatible con la tecnología IP. Estas conexiones facilitan el acceso a los sistemas y herramientas de soporte a la educación, tanto desde el interior de los centros como desde su exterior.
- Redes de datos para la interconexión de los equipos informáticos. Esto permite compartir los recursos de los centros incluido el acceso a las intranets y a Internet. Las redes de datos pueden ofrecerse sobre diferentes tecnologías. En este sentido, la irrupción de la tecnología de redes inalámbricas puede ser de gran ayuda, ya que facilitan la disposición de conectividad desde todos los puntos de un determinado centro (aulas, despachos, bibliotecas, salas de profesores, etc.) sin necesidad de cablear los edificios.
- Equipos informáticos. Aquí se incluyen tanto los equipos de los centros, de los profesores, como de los alumnos, además de aquellos que son necesarios para el almacenamiento de las diferentes herramientas y contenidos educativos. En el caso de los alumnos y profesores hay que destacar también la importancia que tiene el que éstos puedan disponer, además de equipos, de facilidades de conexión de banda ancha para el acceso a todas las soluciones desde sus propios hogares.

5 VENTAJAS E INCONVENIENTES DEL USO DEL ORDENADOR EN EL AULA

Las ventajas e inconvenientes del uso del ordenador en el aula (y de las TIC en general) se pueden calificar desde varios puntos de vista que se mostrarán a continuación.

5.1 Perspectiva del aprendizaje y de los estudiantes

En primer lugar desde la perspectiva del aprendizaje y de los estudiantes con el uso del ordenador en el aula hay que tener en cuenta el interés, la motivación y el alto grado de interacción que provoca el uso de recursos TIC en los alumnos. Esto favorece la iniciativa (ya que hay que tomar decisiones ante cada actividad mostrada por el ordenador) y permite que los estudiantes conozcan sus errores, lo cual permite que en ocasiones puedan volverlo a intentar o se les explique la solución correcta al problema. También hay que tener en cuenta que se puede usar el ordenador para trabajar en grupo (aunque hay que evitar que estos sean numerosos, ya que esto puede provocar que sólo algunos alumnos trabajen y el resto solo miren), favoreciendo el aprendizaje cooperativo. Aunque no todo son ventajas, el mismo material que se usa para motivar a los alumnos también puede provocar distracciones y que en ocasiones los alumnos se dediquen a jugar en vez de a trabajar o que pierdan el tiempo. Además si se permite la libre interacción del estudiante con el software educativo, hay que tener en cuenta el hecho de que puede que dicha interacción no se realice correctamente o que la estrategia de aprendizaje sea muy rígida. Esto puede producir aprendizajes incompletos o el mismo puede ser poco flexible. También hay que indicar que el ámbito de las personas con necesidades especiales es uno de los campos donde el uso del ordenador en general, proporciona mayores ventajas. Ya que muchas deficiencias físicas o psíquicas limitan las posibilidades de comunicación y el acceso a la

información y en muchos de estos casos el ordenador (con software o periféricos especiales) puede ayudar a resolver estas limitaciones.

5.2 Perspectiva del profesorado

Desde el punto de vista del profesorado, los materiales didácticos interactivos permiten que los profesores puedan individualizar el trabajo de sus alumnos ya que el ordenador puede adaptarse a sus conocimientos previos y a su ritmo de trabajo. El uso del ordenador en el aula resulta muy útil también para realizar actividades complementarias y de recuperación. Si el programa permite llevar un historial del trabajo de los alumnos, los profesores pueden evaluar los resultados obtenidos por los estudiantes realizando un seguimiento y control de los mismos. Y si además el software almacena un informe detallado de los errores cometidos por los alumnos, esta información puede permitirle al profesor ver cuáles son los fallos que más cometen los alumnos o cuáles son los temas que más dificultad les causan. El principal problema es que a veces el profesorado no dispone de conocimientos de informática y por tanto puede sentir rechazo hacia la incorporación de las nuevas tecnologías en el aula (o tener la sensación de que se pretende reemplazarlo por un ordenador). También el uso de estos métodos puede requerir una mayor dedicación por parte del profesor, no sólo para ponerse al día con las nuevas tecnologías sino por el hecho de que pueda tener que desarrollar el mismo las actividades que hay que presentar al alumno, buscar información en Internet, realizar en ocasiones *chats* o tutorías virtuales con los alumnos, etc.

5.3 Perspectiva de los Centros Educativos

Por último desde la perspectiva de los centros educativos, las ventajas que ofrecen el uso de las TIC en la enseñanza consisten en que además de mejorar la eficiencia educativa permiten que a través de Internet (mediante páginas Web por ejemplo), los centros se puedan dar a conocer y mostrar sus logros mientras que posibilitan que los profesores puedan compartir recursos educativos como materiales informáticos de uso público, páginas Web de interés educativo, materiales realizados por los profesores y los estudiantes, etc. Los principales

problemas en cambio son los costes de formación del profesorado en las TIC y la necesidad de tener zonas convenientemente equipadas para desarrollar dichas actividades. Además es necesario el mantenimiento de los ordenadores e inversiones en nuevos equipos al cabo de cierto tiempo ya que el mundo de la informática está en continua evolución.

6 SISTEMA TUTORIAL INTELIGENTE: PERSPECTIVA HISTÓRICA

En la década de los 50 el uso de los ordenadores para la enseñanza empezó a cobrar importancia. En esta época surgen los primeros sistemas informáticos para la enseñanza que se conocen con el nombre de *Computer Assisted Instruction* (CAI) [Carbonell 1970]. Los programas CAI tradicionales aprovechaban la experiencia tutorial de los expertos maestros y directamente reflejaban esta habilidad en su comportamiento. Esta circunstancia hacía que el enfoque de los sistemas CAI fuera potencialmente poderoso, pero, al mismo tiempo, presentaba un grave problema ya que pocos maestros (aunque sean expertos en su dominio) pueden anticipar todas las reglas, decisiones y definir *todos* los errores que un estudiante puede cometer. Por ello es prácticamente imposible realizar una aplicación para la enseñanza que contenga *todas* las decisiones imaginables. Otro problema de los sistemas CAI es que debido a la complejidad de reglas que los forman son difíciles de ampliar o modificar.

Dependiendo del papel que juega el ordenador en el aprendizaje los sistemas CAI se pueden clasificar en dos grandes tipos: por un lado están los que crean un ambiente de aprendizaje (hay cambios en función de las respuestas del alumno) y por otro las aplicaciones en las que la estrategia de aprendizaje es estricta. En este último caso la estrategia de enseñanza no se ajusta a las necesidades específicas de cada estudiante.

A partir de los sistemas CAI surgen varias líneas diferentes de sistemas de enseñanza por ordenador. A lo largo de la década siguiente, y a mediados de los

70, se desarrolla el uso de sistemas para el aprendizaje individual basados en la instrucción programada. Estos tienen como objetivo que el alumno adquiera de forma autónoma e individual conocimientos y habilidades, los cuales son establecidos previamente gracias a la ayuda de textos programados. Con ello se pretendía probar que el uso del ordenador podía dar tan buenos resultados como los materiales educativos en los que se basa la enseñanza tradicional [Tyler, 1975]. Otra línea que se desarrolla a partir de ese momento es la basada en que los conocimientos deben construirse, no reproducirse. Los alumnos deben participar por tanto activamente en la construcción de las estructuras del conocimiento. Todo lo que se aprende depende del conocimiento previo y de cómo la nueva información es interpretada por el alumno [Reigeluth 1987].

Además, debido a los problemas mencionados anteriormente de los sistemas CAI (complejidad de diseño de la aplicación debido a la gran cantidad de información a representar y difícil modificación para la mejora e introducción de nuevas utilidades) a comienzos de los años 70 surge una propuesta para mejorar los sistemas CAI utilizando técnicas de Inteligencia Artificial, que estaban en auge en ese momento. A este respecto fue Carbonell con su artículo "*AI in CAI: An Artificial Intelligence Approach to Computer Aided Instruction*" y el desarrollo del sistema SCHOLAR que es un sistema para la enseñanza de la geografía de América del Sur, quien sentó las bases para el desarrollo de los llamados ICAI (Intelligent CAI) que se puede considerar como el punto de partida de los Sistemas Tutoriales Inteligentes (ITS) [Sleeman 1982].

Estos sistemas surgen por tanto después de los tradicionales tutoriales estáticos como un medio para proporcionar una mayor flexibilidad en la estrategia de aprendizaje y lograr una mejor interacción con el usuario. Con los sistemas tutoriales inteligentes se pretende capturar el conocimiento de los expertos, crear interacciones con los usuarios de forma dinámica, y así poder tomar decisiones, aunque éstas no hayan sido previstas por los expertos.

Para lograr este objetivo, estos sistemas tienen que ser capaces de resolver problemas en su dominio de aplicación como si fuera un experto humano. El sistema debe mantener un modelo del conocimiento del usuario para poder actuar en consecuencia al comportamiento de éste, siendo recomendable también desarrollar una interfaz amigable con el usuario que sea fácil de usar y que use un lenguaje natural [Anderson 1990]

La principal ventaja de los ITS frente a los sistemas tutoriales tradicionales es que son más flexibles tanto en el tratamiento del dominio de enseñanza como en la adaptación al alumno. En los sistemas tradicionales que contienen una gran cantidad de reglas y por tanto de información, el alumno se puede perder no siendo guiado de forma adecuada por el tutorial. Además puede ocurrir que como los tutoriales tradicionales no se adaptan al alumno, si éste ya tiene conocimientos del tema se aburra si se le obliga a seguir de forma inflexible una secuencia de actividades con conceptos que ya tiene dominados. También puede ocurrir el caso contrario, que el tutorial avance demasiado rápido para un alumno que no tiene bien asimilados los conocimientos básicos. Los ITS ofrecen como solución la inclusión de varios módulos que contengan información sobre el alumno y el dominio. De esta forma, gracias a la flexibilidad y la adaptación, el tutorial puede ofrecer soluciones a un grupo amplio de usuarios, beneficiándose también del uso de medios audiovisuales (video, audio, animación, etc) que motivan al alumno.

De forma general se puede considerar que los sistemas tutoriales inteligentes están compuestos por los siguientes cuatro módulos: módulo de dominio, modelo del estudiante, módulo tutor y módulo interfaz (figura 2-2) [Burns 1988, Cuevas 1996].

- **Módulo de dominio:** Este módulo contiene el conocimiento acerca del área de estudio. Aquí se encuentra, por tanto, el conocimiento específico y detallado del dominio de aplicación, obtenido de los expertos humanos. Se describen todos los conceptos relacionados con los objetivos de aprendizaje y la forma de proceder en la enseñanza de los mismos. Esto es, para cada alumno se especifican tipos de *tareas* que tiene que realizar de acuerdo a los objetivos que se le han fijado previamente teniendo en cuenta el historial de dicho alumno. Estas tareas, que pueden ser de *motivación*, *presentación*, *evaluación* o *refuerzo* de un concepto serán presentadas al alumno a través del módulo interfaz.

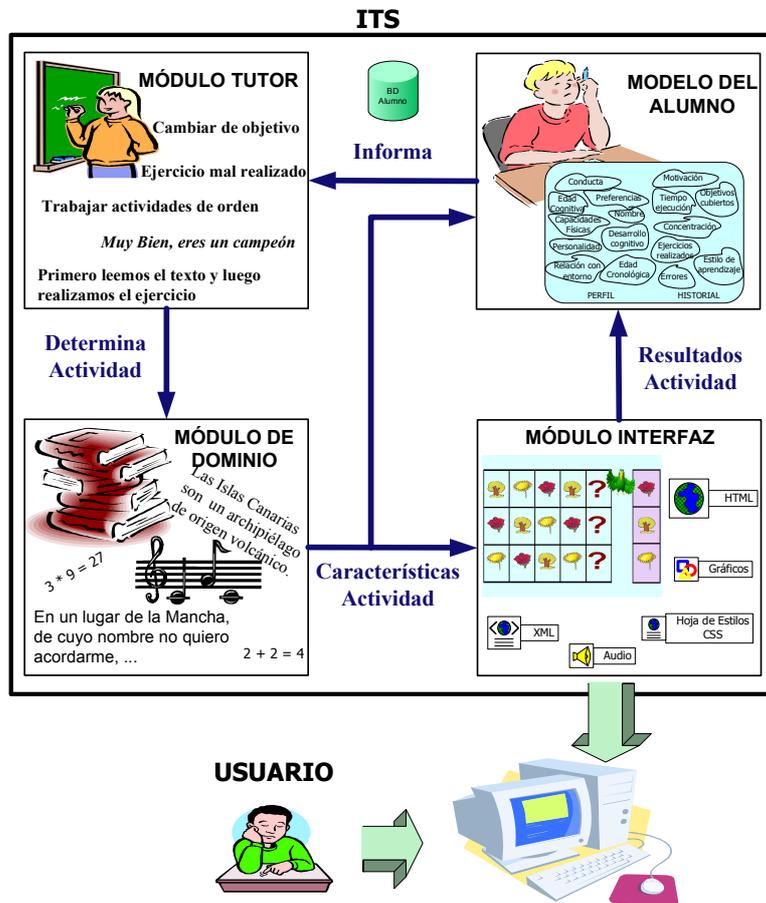


Figura 2-2.- Arquitectura de un Sistema Tutorial Inteligente

- Modelo del estudiante:** En este módulo se guarda el progreso del estudiante que interactúa en el sistema experto. Este módulo contiene todos los datos e información del estudiante, la cual se puede utilizar para elegir el siguiente tema de enseñanza y la metodología o estrategias adecuadas. Básicamente se puede descomponer el modelo del alumno en dos componentes fundamentales: el perfil y el historial. El perfil nos indica aquellas características personales del alumno que están relacionadas con su desarrollo motor, cognitivo y psicosocial. Las variables que se consideran en el perfil son: edad cronológica, edad cognitiva, capacidades físicas (motricidad fina, contacto ocular, capacidad auditiva), desarrollo cognitivo (memoria a corto plazo, atención, capacidad de formar conceptos, capacidad de agrupar objetos en categorías significativas, desarrollo del lenguaje y vocabulario,

comprensión de palabras, capacidad de reacción e iniciativa), relación con el entorno (conexión con el entorno, nivel de adaptación, participación en actividades grupales, obtención de información del entorno), personalidad (miedo al fracaso, dinámica -hiperactivo o pasivo-), patrones de conducta (conducta repetitiva, conducta restringida, actividades estereotipadas), preferencias (colores, intereses personales). Mientras que el historial reúne las variables que nos informan sobre los conceptos previos que conoce el alumno, así como la trayectoria seguida en estos aprendizajes (repeticiones de los ejercicios, tareas no realizadas, ritmo de evolución entre actividades, estilo de aprendizaje -proceso cognitivo y motivacional- ...).

- **Módulo tutor o planificador instruccional:** Se caracteriza por controlar el sistema. Este módulo determina las estrategias de enseñanza de manera que el sistema pueda adaptar y mejorar las estrategias tutoriales en función del alumno. Se tiene que encargar por tanto de detectar el nivel del estudiante, seleccionar la siguiente actividad a trabajar por el estudiante, seleccionar ejemplos, corregir errores, etc.
- **Módulo interfaz:** Es el que se encarga de las presentaciones de los temas a los alumnos. La importancia de la interfaz radica en que puede hacer la interacción del usuario con el sistema sea más o menos comprensible dependiendo de cómo esté diseñada la interfaz, y así afectar el nivel de aceptación que el estudiante tenga del ITS.

Puesto que en esta Tesis Doctoral se describirá el diseño e implementación de un ITS utilizado para el refuerzo de los conceptos lógicos de número, suma y resta, para un currículo español de infantil se van a comentar los requisitos necesarios para el desarrollo de un software para este tipo de currículo y los distintos tipos de software existentes en el mercado para la enseñanza de las matemáticas (epígrafes 7 y 8 respectivamente).

7 SOFTWARE PARA EDUCACIÓN INFANTIL

En los últimos años debido al aumento del uso de las Nuevas Tecnologías de la Información y la Comunicación, se ha empezado a introducir estas tecnologías también en los primeros años de educación infantil. En la actualidad cada vez son más los niños que disponen de un ordenador en su casa y son pocos los alumnos que al menos no hayan visto o utilizado el teclado del ordenador. Es por eso que la sociedad actual demanda el uso de los medios informáticos en el aula como un recurso educativo adicional a los tradicionales.

Hay que tener en cuenta que en la etapa de educación infantil se sienta la base del futuro aprendizaje del niño y además el niño adquiere hábitos de conducta y convivencia con los demás. El uso del ordenador puede contribuir a la consecución de estos logros ya que se enseña a los niños a resolver problemas a través de software educativo y a trabajar en grupo. Además hay que tener en cuenta que con esta edad el niño a la vez que aprende se divierte con el ordenador lo que favorece la motivación del niño a la hora de utilizar este medio.

Los programas educativos tienen que tener varias características primordiales cuando van dirigidos a la enseñanza en el ciclo de educación infantil:

- Deben ser atractivos (usando por ejemplo animación, video, imágenes llamativas, etc.) para que el niño se sienta atraído por este método de enseñanza.
- Explicar de forma clara al niño cuál es la actividad que tiene que realizar y permitir repeticiones dicha explicación en caso de que sea necesario.
- Tienen que permitir que los niños tomen sus propias decisiones aunque éstas sean equivocadas y en este caso darle a conocer el error y permitir nuevas interacciones con la actividad (darle nuevas oportunidades al niño de realizar la actividad).

- En el caso de que el niño sea incapaz de resolver la actividad (siempre se equivoca al intentarlo varias veces) se le debería permitir pasar a otra actividad (para que no se desmotive al ver cómo siempre falla y no consigue avanzar en el programa) pero explicándole en dónde se ha equivocado y enseñándole la correcta resolución del ejercicio.
- Las explicaciones, demostraciones, correcciones, etc., deben darse de forma verbal (ya que algunos niños a esa edad pueden no saber leer) por lo que el volumen y la claridad de la grabación son factores importantes a tener en cuenta.
- Puesto que los destinatarios de ese software son niños pequeños que no tienen en principio demasiado dominio del ordenador se deben evitar en los programas abusar de menús y barras de botones ya que esto aumenta la complejidad del programa (y probablemente impediría que el niño pudiera manejar solo el programa siendo necesaria la ayuda del profesor).
- Por otro lado se debe fomentar el uso del ratón (puesto que como se comentó anteriormente algunos niños a esta edad pueden no saber leer ni escribir, no es conveniente abusar del uso del teclado y evitar interacciones complejas en las actividades o ir aumentando progresivamente la dificultad de éstas (por ejemplo hacer clic en un elemento es una tarea fácil, pero arrastrarlo hasta una determinada posición puede ser una tarea muy complicada para un niño la primera vez que usa un ordenador).
- Fomentar tanto el trabajo individual como el trabajo en grupo, de forma que en el primer caso se adapte al ritmo de aprendizaje de cada alumno y en el segundo los alumnos puedan cooperar para solucionar un problema de forma que se favorezca el diálogo, la comunicación, el intercambio de ideas, etc.
- También hay que tener en cuenta que muchos de estos programas se pueden adaptar para su uso en educación especial de forma que se puede utilizar también este medio para contribuir a la mejora en el aprendizaje de niños con discapacidades físicas o psíquicas.

- Asimismo se deben evitar en estos programas educativos cambios bruscos de nivel de aprendizaje puesto que esto puede provocar que los alumnos se desmoralicen si después de actividades "fáciles" en un nivel se pasa a otras imposibles de realizar.
- Como se mencionó anteriormente, en el caso de que el niño se equivoque se debe corregir el error, pero sin que ello provoque la pérdida de motivación (por ejemplo no decirle al niño que lo ha hecho muy mal).

Además existen varios problemas a la hora de incorporar el uso del ordenador en el aula en educación infantil (y también en otras etapas de la enseñanza) como es la escasez de recursos informáticos que hay en muchos centros, el hecho de que sólo unos pocos profesores se interesen por la incorporación de las nuevas tecnologías en la enseñanza (incluso hay algunos que tienen una actitud de rechazo ante la incorporación de cualquier método que no sea el tradicional), etc.

8 SOFTWARE EDUCATIVO PARA LAS MATEMÁTICAS

Como el objetivo de este trabajo es el diseño e implementación de un ITS para el refuerzo en la enseñanza del concepto de número y las operaciones aritméticas de la suma y la resta, el primer paso es realizar un estudio del software existente en este tópico.

Se ha realizado una revisión de algunos programas de software educativo (tanto comerciales como libres) que comprenden juegos, herramientas para el aprendizaje de las matemáticas, etc. para los niveles de educación infantil, primaria y secundaria. Hemos hecho una recopilación de una gran variedad de programas existentes en el mercado que permiten el aprendizaje de los conceptos de lógica, que enseñan a reconocer los números, a sumar, restar, multiplicar, dividir, etc., pero sin adaptarse a las características o la evolución del alumno (como mucho, en algunos se puede elegir el nivel en el que se quiere trabajar). Estos se pueden ver

en forma de tabla resumen en el epígrafe 8.1: Revisión de software educativo en matemáticas.

Hemos realizado también un estudio más exhaustivo sobre sistemas expertos y tutoriales que tenían la característica de la adaptabilidad. Algunos de los cuales son:

- El sistema experto EULER es entrenador y tutor al mismo tiempo y esta construido con técnicas de Inteligencia Artificial. Se basa en la enseñanza del Cálculo. Fue desarrollado por el grupo de Software Educativo del Departamento de Cibernética de la Facultad de Matemáticas de la Universidad de La Habana.
- PREMATIC es un tutorial hipermedia destinado a estudiantes de nivel medio (preuniversitario o bachillerato) que contiene los conceptos, definiciones, fórmulas, teoremas y métodos de resolución de problemas que se incluyen en los cursos de matemáticas de esos niveles, haciendo énfasis en las operaciones aritméticas con números reales, el trabajo con variables, funciones, sistemas de ecuaciones, trigonometría y geometría. La idea es que sirva como guía en el repaso y sistematización de los conocimientos y para ello cuenta con más de 700 ejercicios precedidos de ejemplos.
- El paquete ESTADIS es también un tutorial hipermedia dirigido a usuarios de nivel medio y superior con un enfoque principalmente práctico y una estrategia pedagógica elaborada a partir de las dificultades de los estudiantes. Abarca los contenidos básicos de Estadística Descriptiva y no Paramétrica haciendo énfasis en la selección de las herramientas estadísticas adecuadas para la solución de problemas y la interpretación correcta de sus resultados.
- CALCULUS es un libro electrónico hipermedia para la enseñanza - aprendizaje orientado a estudiantes universitarios y de nivel medio-superior que optan por cursos avanzados que requieran conocer y aplicar los conceptos esenciales tratados en los cursos de Cálculo Diferencial e Integral en una variable y de Matemática Numérica. Los contenidos que incuye son: Límite y continuidad, Calculo Diferencial, Cálculo Integral y

un apéndice: Explorador de cónicas. El capítulo dedicado a la Matemática Numérica aborda la solución aproximada de ecuaciones no lineales, la solución de sistemas de ecuaciones lineales, la interpolación y aproximación, la diferenciación e integración numéricas y la resolución aproximada de ecuaciones diferenciales ordinarias. Contiene más de 1000 ejemplos y ejercicios y permite la autoevaluación empleando un tutor inteligente. Además combina y muestra el empleo del asistente matemático DERIVE.

- Apolonio1+ es un sistema tutorial inteligente cuyo objetivo es desarrollar la habilidad para la solución de problemas en estudiantes universitarios. Está integrado por cuatro componentes: Currículo, Planificador, Tutor e Interfaz, basándose en el modelo teórico del grupo Heron de la Universidad de Québec en Montreal. El objetivo de este proyecto fue la disminución del tiempo y coste de aprendizaje y el aumento de la efectividad del mismo, ya que se entrena a una persona particular aprovechando sus conocimientos previos en el tema a tratar.
- GEOLOG-WIN/GEOLOG2000 es un sistema dinámico de geometría que integra un sistema tutorial basado en el conocimiento para solucionar ejercicios mediante construcción, cálculo y prueba.
- PSS-Tutor desarrollado por la Universidad de Québec en Montreal es un sistema tutorial inteligente para la enseñanza de estrategias de alto nivel para la resolución de problemas matemáticos. Se establece un diálogo con el estudiante con el fin de averiguar qué dificultades tiene para resolver el problema y a partir de esta interacción el tutorial decidirá de que forma se adopta la resolución del problema.
- Advanced Geometry Tutor es un tutorial inteligente para geometría. Su objetivo principal es enseñarle a los alumnos diversas estrategias y teoremas para la resolución de problemas geométricos. Para probar los teoremas usa principalmente dos métodos de resolución: el encadenamiento hacia delante (FC) y el encadenamiento hacia detrás (BC). En cuanto a la adaptación, la fórmula utilizada es la siguiente: cuando el estudiante realiza correctamente un paso en la resolución del problema, se aumenta el nivel de dificultad y de forma inversa, si falla se baja el nivel. Existen tres niveles de dificultad: Showtell, en el que

tutorial le explica al estudiante qué tiene que hacer y le muestra cómo se hace; Tell en el que se le dice al estudiante que tiene que hacer, pero tiene que ser él quien realice el paso; y por último Prompt en el que el tutorial simplemente incita al estudiante a realizar el siguiente paso.

- MATH VIZUAL es un sistema tutorial inteligente en el ámbito de la aritmética y el álgebra cuyas características principales son: que el inicio depende del nivel en el que se encuentre el alumno, los progresos se hacen en función de los resultados que obtenga el mismo, se da corrección a las respuestas equivocadas, se hacen preguntas, se orienta al alumno para que descubra las respuestas por sí mismo y se incorporan mecanismos que permiten al sistema "hablar", lo cual le otorga ventajas para que sea utilizado, por ejemplo, por discapacitados.
- Tutormat es un sistema tutorial matemático flexible, en el que los conceptos que en él se explican pueden ser fácilmente extendidos o modificados individualmente por cada profesor o por el conjunto de profesores de un determinado centro escolar. A partir de la información generada por los profesores genera tests, prácticas y/o exámenes en función del grado de conocimiento del alumno y los evalúa on-line inmediatamente o en forma de evaluación continua (esto significa que si se falla antes de dar las respuestas correctas se proporciona una ayuda para resolverlas).
- Matemanía I es un programa para niños de educación primaria que los inicia en el reconocimiento de las cantidades y las operaciones básicas, concretamente en: la representación numérica, la comparación de cantidades, ordinales y cardinales, cálculo mental, sumas y restas y resolución de problemas. El niño realizará los ejercicios en la forma que previamente haya definido el profesor mediante el programa de configuración, adecuando así su utilización a cada caso concreto.
- Cálculo intuitivo es una introducción a la aritmética para niños de educación infantil y primaria donde el usuario puede realizar ejercicios de: identificación de cantidades, identificación de cifras, comparación de cantidades, resolución de sumas y resolución de restas. Desarrolla la capacidad lógica y de razonamiento del niño y permite adaptar el programa a las características del mismo.

- Ms. Linqvist es un sistema tutorial inteligente desarrollado por la Universidad Carnegie Mellon que ayuda a los estudiantes a la comprensión y resolución de problemas de álgebra. Este programa es capaz de realizar una conversación con el estudiante y de adaptarse al mismo en función de los errores. Las estrategias usadas son: usar y trabajar ejemplos, explicar en lenguaje natural antes de hacerlo en lenguaje formal y usar métodos de descomposición y sustitución.

Se puede observar una tabla resumen de sistemas en el epígrafe 8.2. Estos sistemas se caracterizan por ser diseñados para grupos de estudiantes homogéneos en su perfil (edad, motivaciones, etc.) y en su grado de conocimiento. La mayoría de estos tutoriales están orientados a los niveles de Bachillerato y Universitario y sólo uno se corresponde con el nivel de infantil en el que nosotros estamos trabajando. Ninguno de ellos está orientado a alumnos con dificultades de aprendizaje y aunque algunos de ellos tienen en cuenta las dificultades de los alumnos para la resolución de los ejercicios, nos encontramos con que no se valora la motivación al estudiante para la resolución del ejercicio. En cuanto a la adaptación es variada dependiendo del tutorial. Podemos encontrar tutoriales que no se adaptan a las características de los estudiantes (por ejemplo Ms. Linqvist) y otros que varían en función de los conocimientos previos de los alumnos o de los resultados (por ejemplo Math Vizual).

En esta Tesis Doctoral se propone un Sistema Tutorial Inteligente Adaptivo para el refuerzo y seguimiento escolar en el currículum español de enseñanza infantil y primaria, de los objetivos de reconocimiento del número, suma y resta. El ITS que se propone se caracterizará por adaptar tanto los objetivos como el método de enseñanza.

8.1 Revisión de Software Educativo en Matemáticas

NOMBRE	DISTRIBUIDORA	NIVEL	DESCRIPCIÓN
101 Ejercicios	Emme Interactive	Infantil	Ejercicios para el reconocimiento de los colores, los sonidos, las formas, el aprendizaje de las cifras y de las letras.
102 Ejercicios	Emme Interactive	Primaria	Actividades variadas para que los niños aprendan a contar, sumar, leer, pensar, memorizar, emparejar y escuchar.
Adiboo: Lectura y Cálculo	Coktel Educative	Infantil	Lecciones, herramientas y vídeos sobre lengua y matemáticas.
Adibú: Yo Leo. Yo Calculo	Coktel Educative	Primaria	Actividades y juegos para familiarizarse con las sumas y las restas, escribir, ordenar y comparar los números.
Aritmética, Probabilidad y Estadística	Vermic S.A.	Secundaria	Programa que permite de forma fácil el aprendizaje de los principales conceptos de aritmética, probabilidad y estadística de enseñanza secundaria.
Álgebra	Vermic S.A.	Secundaria	Programa para el aprendizaje del álgebra.
Aprende con Arnoldo y sus Amigos Preescolar Matemáticas	Vermic S.A.	Infantil	Programa que enseña a los niños los números del 1 al 10, comparaciones, la recta numérica, sumas, restas, etc.
Aprende con Pepo	Edicinco	Infantil y Primaria	Juego para aprender nociones de seriaciones de objetos y el orden natural del tiempo.

Aprende con Zipi y Zape	Zeta Multimedia	Infantil y Primaria	Programa para el aprendizaje de lengua, matemáticas y ciencias.
Aritmética	Vermic S.A.	Secundaria	Programa para el aprendizaje de la aritmética de secundaria.
Arki	Consultora Informática S.R.L.	Infantil y Especial	Realización de actividades de clasificación, cuantificación, seriación y equivalencias de objetos.
Basquet	Vermic S.A.	Secundaria	Juego en el que usando el baloncesto se plantean preguntas de diversos temas como Historia, Civismo, Geografía, Lengua, Matemáticas, C. Naturales, Ed. Artística, Ed. Física, etc.
Batalla en el espacio	Swissoft	Primaria	Enseñanza de las operaciones de multiplicación y división.
Cálculo: Iniciación a la suma y resta	MEC Centro de Desarrollo Curricular	Infantil y Primaria	Realización de operaciones de suma y resta.
Cálculo: Los 9 primeros números	MEC Centro de Desarrollo Curricular	Infantil y Primaria	Realización de actividades de reconocimiento, relaciones y ordenación con números del 1 al 9.
Calculosaurios	Edicinco	Infantil y Primaria	Programa con ejercicios matemáticos ambientados en la prehistoria.
Calcum	Edicinco S.A.	Primaria y Secundaria	Programa para el aprendizaje de la resolución de sumas, restas, multiplicaciones, divisiones, polinomios, ecuaciones, cálculo fraccional, etc.
Cantidades	Junta de Andalucía	Infantil y Primaria	Actividades de ordenación de diferentes cantidades. Posibilidad de elegir el número de variables.

Castillo de Drácula	Vermic S.A.	Primaria y Secundaria	Juego en el que hay que ir resolviendo correctamente las preguntas de matemáticas que se plantean (sumas, restas, multiplicaciones, divisiones, ordinales, series, etc.) para destruir el Castillo de Drácula.
Circo de Don Pepón	MEC Centro de Desarrollo Curricular	Infantil y Primaria	Realización de seriaciones en función del tamaño, el color o la forma.
Clasificaciones	MEC Centro de Desarrollo Curricular	Infantil y Primaria	Programa con dos grupos de actividades: clasificación de objetos en función de uso y de pertenencia o no, a conjuntos de objetos.
Clic	Generalitat de Cataluña y MEC	Infantil y Primaria	Programa informático que permite la creación de actividades para múltiples áreas de conocimiento (entre ellas matemáticas).
Compra	Junta de Andalucía	Infantil y Primaria	Operaciones matemáticas para que los niños aprendan a realizar diferentes tipos de compras.
Conejo Lector	TLC Multimedia	Infantil y Primaria	Programa para que los niños aprendan conceptos de matemáticas, lenguaje, creatividad y habilidades de lógica.
Contar y Agrupar	Zeta Multimedia	Infantil y Primaria	Programa para que los niños aprendan el concepto de número.
Correspondencias	MEC Centro de Desarrollo Curricular	Infantil y Primaria	Este programa consiste en completar series según tamaño, color o forma.
Desafíos matemáticos	Swissoft	Primaria	Actividades de correspondencia de conjuntos y ordenaciones de números.

Descubre con Adibuchú en el mar	Coktel Educative	Infantil	Juego ambientado en la playa para reconocer las formas y los colores, contar de 0 a 5 y ampliar el vocabulario sobre el mar.
Edebits Primaria	Edebé Digital	Primaria	Actividades de Lengua, Matemáticas y Conocimiento del Medio.
El Conejo Matemático	The Learning Company	Primaria	Programa con cuatro actividades diferentes (contar, ordenar, test de memoria y operaciones de suma y resta) para practicar los conceptos de numeración y sumas y restas con números del 0 al 99.
El Mago de Oz	Zeta Multimedia	Primaria	Juego para aprender cálculo y resolución de problemas, reconocimiento de colores, formas y figuras.
Expedición a Martemática	MSD Informática	Primaria y Secundaria	Programa para que el alumno mediante un juego de aventura espacial aprenda las operaciones de suma, resta, multiplicación, división, equivalencia y seriación.
Fantasia	Vermic S.A.	Infantil y Primaria	Programa para desarrollar la capacidad para la solución de problemas, la concentración y el manejo de los números y las letras.
Frankie	Knowledge Adventure	Infantil y Primaria	Programa que enseña el reconocimiento de letras y números, a deletrear, a contar, a sumar y restar, etc.
Geometría	Vermic S.A.	Secundaria	Programa para el aprendizaje de la geometría de secundaria.

Juega con las Matemáticas	Zeta Multimedia	Primaria y Secundaria	Programa educativo que usando civilizaciones históricas propone ejercicios de resolución de problemas de geometría, aritmética, medidas y fracciones.
Juega con los Teletubbies	Zeta Multimedia	Infantil	Juego para reconocer formas, posiciones, medidas y cantidades, permitiendo el desarrollo de la imaginación y la creatividad.
KBruch	Edubuntu	Primaria	Programa para calcular fracciones.
Kig	Edubuntu	Secundaria	Herramienta que permite crear construcciones geométricas de gran precisión.
KmPlot	Edubuntu	Primaria y Secundaria	Aplicación para dibujar funciones regulares, funciones paramétricas, visualizar ecuaciones, etc.
KPercentage	Edubuntu	Primaria	Aplicación matemática para calcular porcentajes.
KTurtle	Edubuntu	Primaria	KTurtle es un entorno de programación educativa, utilizando Logo como lenguaje de programación. Enseña a los niños los conceptos básicos de programación, matemáticas y geometría.
La Casa de las Matemáticas de Millie	T&R Multimedia	Infantil y Primaria	Contiene 6 actividades para el aprendizaje de números, formas y secuencias.
La maldición de Kalkul-atú	Edicinco	Primaria	Juego para aprender matemáticas, ambientado en una aventura en el antiguo Egipto.

Laberinto de Arnoldo 8 Materias	Vermic	Primaria y Secundaria	Programa que permite aprender las materias de Historia, Civismo, Geografía, Lengua, Matemáticas, Ciencias Naturales, Ed. Artística y Ed. Física
Los Pequeños Exploradores	The Learning Company	Primaria	Es un juego en el que los niños tienen que solucionar problemas de matemática, lógica, geografía, ciencias, lectura y destrezas lingüísticas al mismo tiempo que intentan rescatar una persona secuestrada y enfrentarse a monstruos.
M&M's Las fórmulas pérdidas	Zeta Multimedia	Primaria	Programa para realizar actividades de sumar, restar, multiplicar y dividir.
Maratón 7 Maravillas	Vermic S.A.	Primaria y Secundaria	Juego en el que el niño realiza una carrera recorriendo las 7 maravillas del mundo antiguo, compitiendo con otros personajes a los que podrá ganar resolviendo problemas de matemáticas e historia.
Matemáticas con Pipo	Cibal Multimedia	Infantil y Primaria	Juego para que los niños aprendan la serie de números, a contar, a ordenar y a resolver problemas variados.
Mates Blaster Junior	Anaya Multimedia	Infantil	Programa-juego para que los niños empiecen a trabajar con números, series y operaciones elementales.
Mates Blaster: El misterio del ladrón de cerebros	Anaya Multimedia	Secundaria	Este programa utiliza un juego como medio para la enseñanza de operaciones de álgebra, decimales, números primos, problemas, etc.

Mates Blaster: El Secreto de la Ciudad Perdida	Anaya Multimedia	Infantil, Primaria y Secundaria	Este programa plantea los problemas de matemáticas como un juego de naves espaciales. Permite la edición de actividades
Mates Blaster: En busca de Positrón	Anaya	Primaria	Juego para aprender de forma sencilla y eficaz matemáticas.
Mi primer cd-rom	Clásicos informáticos	Infantil	Programa para el aprendizaje interactivo y guiado que refuerza el pensamiento lógico, el cálculo básico, la memoria visual, etc.
Mía Matemática- Justo a tiempo	Zeta Multimedia - Kutoka Interactive	Primaria	Programa para aprender jugando los conceptos matemáticos de lógica y cálculo.
Minimonstruos	Zeta Multimedia	Infantil y Primaria	Juego para aprender el reconocimiento de números, a contar, estimar y calcular, realizar sucesiones de números, resolver problemas de lógica, sumas, restas, etc.
Mis primeros pasos con Pipo	Micronet	Infantil	Juego que estimula la memoria visual, verbal, numérica y auditiva, enseñando entre otras materias: lenguaje, cálculo, música, el cuerpo humano, etc.
Mix-Match	Edicinco S.A.	Infantil y Primaria	Enseña a distinguir formas, letras y números
Números y Bloques	Vermic S.A.	Infantil y Primaria	Programa que mediante un juego permite a los niños a aprender a contar, sumar, restar, multiplicar y dividir.
Ocho juegos de cálculo mental	Javier Rey	Primaria	Programa que plantea actividades de cálculo mental

Olimpiadas de Arnoldo Matemáticas	Vermic S.A.	Primaria y Secundaria	Programa que mediante el uso de disciplinas olímpicas permite practicar los números naturales y fraccionarios, los sistemas de medición y problemas de probabilidad y estadística.
Pepe el maquinista	Informática Alcorta	Infantil y Primaria	Programa sobre conocimiento del entorno y conceptos básicos del lenguaje y lógico-matemáticos.
Primer	MEC Centro de Desarrollo Curricular	Primaria y Secundaria	Programa para trabajar la prioridad en las operaciones matemáticas
Proceso	Junta de Andalucía	Infantil y Primaria	Permite al niño aprender a trabajar con secuencias temporales.
Sumar y Restar	Zeta Multimedia	Primaria	Programa que presenta series de actividades para que los niños aprendan los conceptos de la suma y la resta.
Sumas y Restas	Edicinco S.A.	Infantil y Primaria	Este programa introduce al usuario en los conceptos de cálculo de la suma y la resta.
Taller de juegos de Aladdin	Disney Arts	Infantil y Primaria	Juegos para aprender a dibujar, leer, contar, escribir y mejorar la concentración.
Trampolín	Knowledge Adventure	Primaria	Módulos interactivos que enseñan al niño estructura, lectura y cálculo básico, entre otros temas.
Trampolín educación infantil	Knowledge Adventure	Infantil	Juego para educación infantil que incluye el aprendizaje de letras y números, vocabulario, lectura música, creatividad, secuenciar y ordenar, selección visual, aptitud con el ordenador, etc.

TuxMath	Edubuntu		Juego basado en el arcade "Missile Comando" que permite al niño defender ciudades resolviendo problemas aritméticos.
Tweenies ¡Listos para jugar!	Zeta Multimedia	Infantil	Juego en el que los niños pueden aprender números y aritmética elemental, formas y medidas.
Unidades de Medida	Edicinco S.A.	Primaria	Aprendizaje de los conceptos de longitud, capacidad, masa y tiempo.
Viva Matemáticas	System	Infantil y Primaria	Programa para el aprendizaje de las matemáticas mediante la realización de ejercicios prácticos.
Win-Abc	MEC-Pntic	Infantil, Primaria y Secundaria	Programa para aprender lectoescritura y cálculo.
Zoológico	Vermic S.A.	Infantil	Juego en el que mediante el recorrido de un zoológico se enseña a los niños a conocer los números, los colores, las letras, las formas, a aprender a contar, etc.

8.2 Revisión de Sistemas Expertos y Tutoriales

NOMBRE	NIVEL	ADAPTACIÓN	CONTENIDO
EULER			Sistema experto entrenador y tutor para la enseñanza del Cálculo.
PREMATIC	Bachillerato		Tutorial hipermedia sobre operaciones aritméticas con números reales, el trabajo con variables, funciones, sistemas de ecuaciones, trigonometría y geometría.
ESTADIS	Bachillerato y Universitario		Tutorial hipermedia con contenidos básicos de Estadística Descriptiva y no Paramétrica.
CALCULUS	Bachillerato y Universitario		Libro electrónico hipermedia para la enseñanza de Límites y continuidades, Calculo Diferencial, Cálculo Integral y un apéndice: Explorador de cónicas.
APOLONIO+	Universitario	En función de conocimientos previos	Sistema Tutorial Inteligente cuyo objetivo es desarrollar la habilidad para la solución de problemas.
GEOLOG-WIN/ GEOLOG2000			Sistema dinámico de geometría que integra un sistema tutorial basado en el conocimiento.
PSS-TUTOR		En función de las dificultades que tenga el alumno para resolver el problema	Sistema Tutorial Inteligente para la enseñanza de estrategias de alto nivel para la resolución de problemas matemáticos.
ADVANCED GEOMETRY TUTOR		En función de las respuestas del estudiante.	Sistema Tutorial Inteligente para Geometría.

MATH VIZUAL		En función del nivel en el que se encuentre el alumno al inicio y los resultados.	Sistema Tutorial Inteligente en el ámbito de la aritmética y el álgebra.
TUTORMAT			Sistema Tutorial Matemático flexible, en el que los conceptos que en él se explican pueden ser fácilmente extendidos o modificados individualmente por cada profesor.
MATEMANIA I	Primaria	El niño realizará los ejercicios en la forma que previamente haya definido el profesor mediante el programa de configuración.	Tutorial que inicia en el reconocimiento de las cantidades y las operaciones básicas, concretamente en: la representación numérica, la comparación de cantidades, ordinales y cardinales, cálculo mental, sumas y restas y resolución de problemas.
CALCULO INTUITIVO	Infantil y Primaria	En función de las características del niño.	Tutorial donde el usuario puede realizar ejercicios de identificación de cantidades, identificación de cifras, comparación de cantidades, resolución de sumas y resolución de restas.
MS. LINQUIST		En función de los errores.	Sistema Tutorial Inteligente que ayuda a los estudiantes a la comprensión y resolución de problemas de álgebra.

CAPÍTULO II

ITS PARA EL REFUERZO DE LOS CONCEPTOS LÓGICOS DE NÚMERO, SUMA Y RESTA

1 INTRODUCCIÓN

El objetivo de esta Tesis Doctoral es presentar el diseño e implementación de un Sistema Tutorial Inteligente (ITS) para el refuerzo y seguimiento escolar en el curriculum español de enseñanza infantil y primaria. El tutorial seguirá un proceso de enseñanza individualizada que consiste en determinar, a partir de unas características predeterminadas de los alumnos, cuáles son los objetivos de aprendizaje. Para ello se deben elaborar una serie de actividades a realizar por el alumno que le permita adquirir las habilidades marcadas. El conjunto de actividades, por tanto, no será estándar para todos los alumnos, sino que dependerá de las características de cada uno de ellos.

El ITS está compuesto de una estructura modular que hace factible la reutilización en otros dominios de aplicación. En la figura 2-1 se muestran los componentes principales del ITS [Burns 1991].

Aunque para la implementación del ITS hemos decidido usar las matemáticas como dominio de aplicación, éste se ha diseñado de forma que sea posible su generalización a cualquier otro dominio. Dos aspectos son fundamentales para permitir dicha generalización: el enfoque centrado en base de datos (BD), en donde toda la información estática y dinámica del sistema está almacenada en una BD; así como el diseño de un sistema borroso y multiagente estructurado en la implementación del Planificador Instruccional.

La forma de proceder del ITS implementado está dirigida por el Gestor que es un programa implementado en C++ y que tiene las siguientes funcionalidades:

1. El Gestor consulta en el modelo del estudiante tanto su personalidad como su evolución en el proceso de aprendizaje del objetivo (EAO). Esto es necesario para adaptar el contenido y lo que se presenta a cada alumno.
2. El Gestor informa de estos parámetros al Planificador Instruccional y le solicita el grado de dificultad, el objetivo y el tipo de la siguiente actividad con el que trabajará el alumno.
3. A continuación el Gestor busca en la BD actividades con el nivel de dificultad deseada.
4. Estas actividades se le pasan a la Interfaz Multimedia que se encarga de presentarle al alumno la actividad que tiene que realizar y devuelve la interacción realizada al gestor para que registre la evolución en la BD.

En este capítulo se presentará en primer lugar el dominio al que se ha aplicado el ITS. Luego se explicarán la Base de Datos, la Interfaz Gráfica y por último el Generador de Actividades. Debido a su importancia en esta Tesis, al Planificador Instruccional se le dedicará un capítulo completo (Capítulo III).

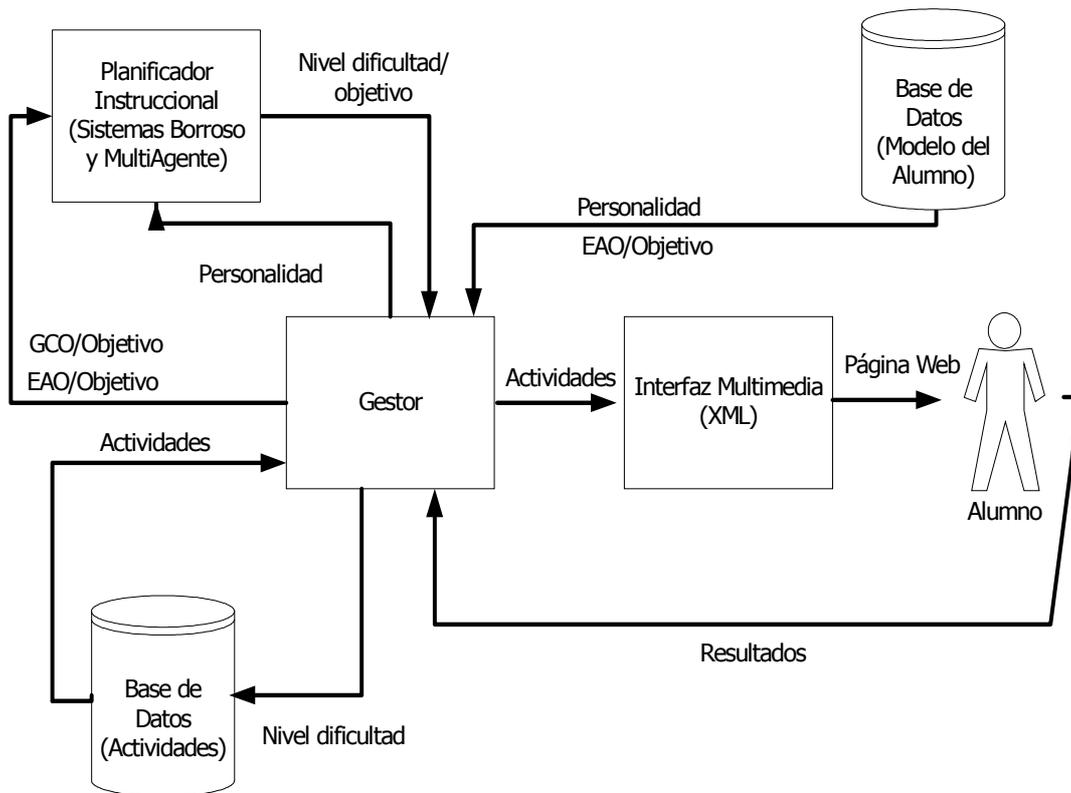


Figura 2-1. Arquitectura del ITS para el refuerzo de los conceptos lógicos de número, suma y resta.

2 DOMINIO

El primer paso en el diseño del ITS es especificar el objetivo de enseñanza, que en nuestro caso es el concepto lógico de número y los objetivos de suma y resta en el currículo español de infantil y primaria. Tal como hemos indicado, aunque se haya elegido este dominio, el diseño del tutorial se ha realizado de forma que se pueda utilizar en cualquier dominio de aplicación.

El temario que se va a incorporar al ITS se ha dividido en objetivos, algunos de los cuales se trabajan de forma simultánea mientras que otros son prerequisites de sus sucesores. Es por ello que el ITS se estructura en fases, que se caracterizan por un conjunto de objetivos que se trabajan simultáneamente. Cuando se cubre

una fase, se pasa a la siguiente y en el caso de que no se avance en una fase, se permanece en ella o si los resultados no son los deseados se puede regresar a la predecesora [Aguilar 2003a].

Se establecen cuatro fases (Figura 2-2):

1. Lógica.
2. Concepto de número, suma y resta con números de un dígito.
3. Concepto de número, suma y resta sin llevar con números de dos dígitos.
4. Suma y Resta llevando con números de dos dígitos.

Estas fases se tienen que ir cubriendo sucesivamente y progresar/retroceder en las mismas en función de los resultados de la ejecución de las actividades por el alumno.

La fase 1 (Lógica) consta de 4 objetivos a cubrir, que son realizados en paralelo. Cuando el alumno realiza las actividades correspondientes a cada objetivo con un porcentaje adecuado de aciertos, puede pasar a la fase 2, caracterizada por otra serie de objetivos. Del mismo modo se puede progresar hacia las siguientes fases.

Las actividades a realizar por el alumno, correspondientes a la fase 1, han sido encuadradas en dos niveles de dificultad, mientras que las actividades correspondientes a los objetivos de las fases 2, 3 y 4 se han organizado en tres niveles de dificultad.

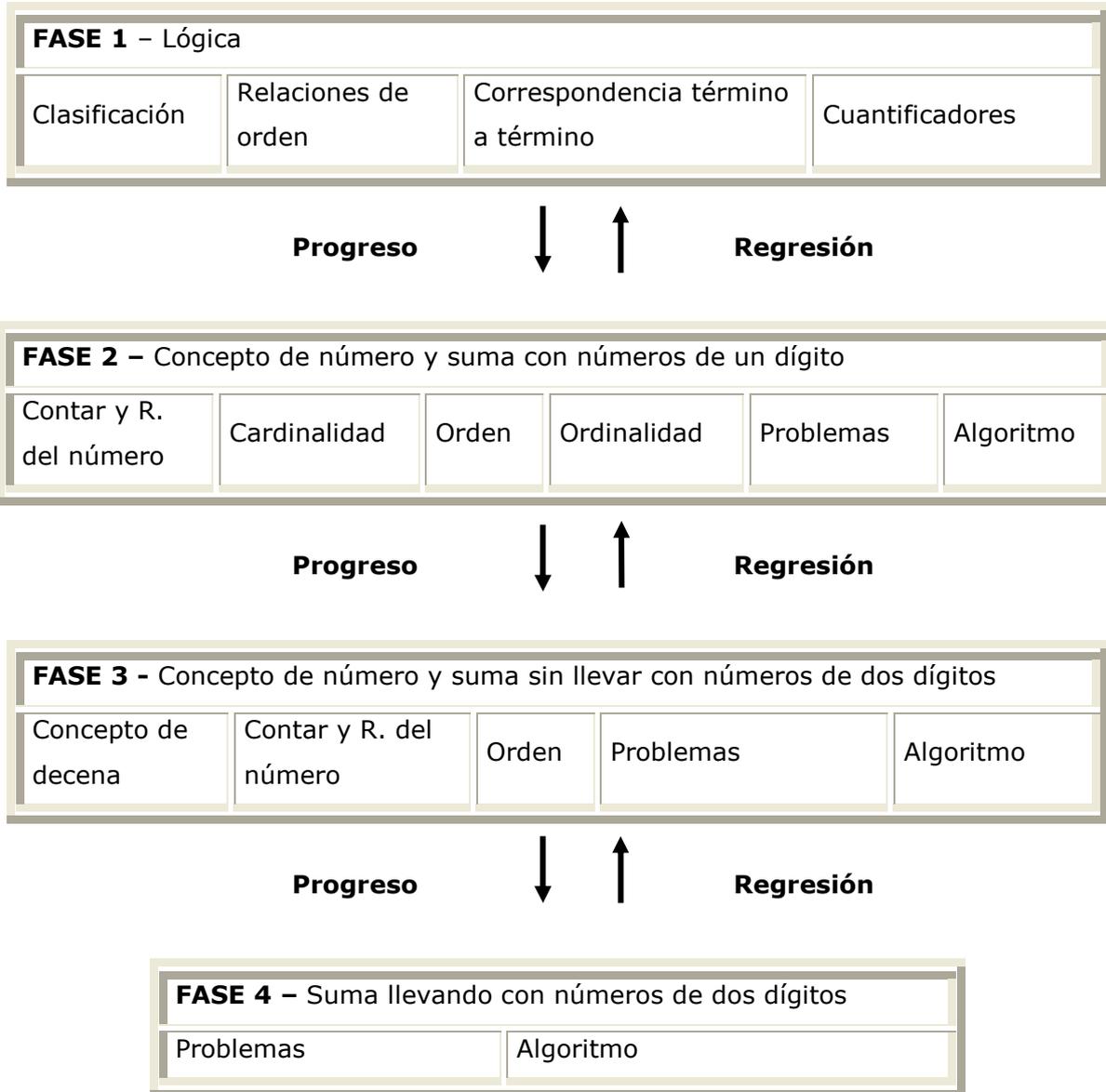


Figura 2-2. Estructuración en fases de los subobjetivos a cubrir en la enseñanza de la suma.

3 BASE DE DATOS

La base de datos del modelo del estudiante tiene la información relativa a las características de cada estudiante. La base de datos de conocimiento pedagógico contiene las actividades que se muestran a los alumnos para la consecución de cada objetivo. Se ha implementado con una sola base de datos relacional que consta de diez tablas. La descripción de éstas se puede ver en las tablas 2-1 a 2-10.

FASE: Tabla en la que se almacenan las fases en las que se divide el ITS	
Nombre del Campo	Descripción del Campo
IdFase	Identificador de la Fase. El identificador es un valor único para cada Fase, por lo que no puede haber dos que tengan el mismo identificador.
Nombre	Nombre de la Fase.

Tabla 2-1. Tabla FASE de la Base de Datos.

OBJETIVO: Tabla en la que se almacenan los diferentes objetivos de aprendizaje que componen el ITS.	
Nombre del Campo	Descripción del Campo
IdObj	Identificador del Objetivo. El identificador es un valor único para cada objetivo, por lo que no puede haber dos que tengan el mismo identificador.
Nombre	Nombre del Objetivo.

Tabla 2-2. Tabla OBJETIVO de la Base de Datos.

NIVEL OBJETIVO: Clasificación de los niveles en los que se divide un Objetivo.	
Nombre del Campo	Descripción del Campo
IdObj	Identificador del Objetivo. El identificador es un valor único para cada Objetivo, por lo que no puede haber dos que tengan el mismo identificador.
NivelObj	Es el nivel de dificultad del Objetivo. La fase 1 consta de dos niveles de dificultad, mientras que los objetivos de las fases 2, 3 y 4 se han organizado en tres niveles de dificultad.

Tabla 2-3. Tabla NIVEL OBJETIVO de la Base de Datos.

ACTIVIDAD: Tabla en la que se almacenan todas las actividades del ITS.	
Nombre del Campo	Descripción del Campo
IdAct	Identificador de la Actividad. El identificador es un valor único para cada actividad, por lo que no puede haber dos que tengan el mismo identificador.
Ruta	Dirección de almacenamiento en disco con extensión .html que contiene la actividad.
TipoAct	Identificador de las diferentes clases de actividades en la que se puede dividir un Objetivo. La descripción completa de todos los tipos de actividades que se han creado se encuentra en el Apéndice A.
Importante	Determina si una Actividad tiene prioridad o no dentro del Objetivo (si es prioritaria aumentan sus probabilidades de ser mostrada al alumno).

Tabla 2-4. Tabla ACTIVIDAD de la Base de Datos.

ALUMNO: Tabla en la que se almacenan los datos personales de los alumnos del ITS	
Nombre del Campo	Descripción del Campo
IdAl	Identificador del Alumno. El identificador es un valor único para cada alumno, por lo que no puede haber dos que tengan el mismo identificador.
Nombre	Nombre del Alumno
Apellidos	Apellidos del Alumno
Edad	Edad del Alumno
NivelEd	Nivel Educativo al que pertenece el alumno. Por ejemplo: Infantil, Primaria, Secundaria, etc.
TipoAl	Identificador que determina las singularidades del alumno. Puede ser: tipo 1 (alumnos con miedo al fracaso); tipo 2 (alumnos motivados); tipo 3 (alumnos). Estos tipos de alumnos se explicarán con más detalle en el epígrafe 4.2.
Inicio	Determina si es la primera vez que el alumno entra en el ITS o ya tiene historial.

Tabla 2-5. Tabla ALUMNO de la Base de Datos.

Clasificación Objetivos: Clasificación de las objetivos asociados a cada fase.	
Nombre del Campo	Descripción del Campo
IdFase	Identificador de la Fase.
IdObj	Identificador del Objetivo.

Tabla 2-6. Tabla Clasificación Objetivos de la Base de Datos.

Clasificación Actividades: Clasificación de las actividades asociadas a cada objetivo.	
Nombre del Campo	Descripción del Campo
IdAct	Identificador de la Actividad
IdObj	Identificador del Objetivo.
nivelObj	Nivel de dificultad del Objetivo.

Tabla 2-7. Tabla Clasificación Actividades de la Base de Datos.

Historial Borroso: Historial de los resultados del cálculo del nivel en el que tendrá que trabajar el alumno.	
Nombre del Campo	Descripción del Campo
IdAl	Identificador del Alumno
IdObj	Identificador del Objetivo.
nivelObj	Nivel de dificultad del Objetivo.
ResultFuzzy	Resultado del Sistema Borroso (Capítulo III)
Fecha	Fecha en la que se ha realizado el cálculo del siguiente nivel de dificultad en el que tendrá que trabajar el alumno.
Hora	Hora en la que se ha realizado el cálculo del siguiente nivel de dificultad en el que tendrá que trabajar el alumno.

Tabla 2-8. Tabla Historial Borroso de la Base de Datos.

Historial Objetivos: Historial de los objetivos en los que ha trabajado el alumno y de los resultados que ha obtenido.	
Nombre del Campo	Descripción del Campo
IdAl	Identificador del Alumno
IdObj	Identificador del Objetivo.
nivelObj	Nivel de dificultad del Objetivo.
GCO	Porcentaje de Aciertos del alumno en las actividades del objetivo.
PosObj	Orden en el que se ha mostrado el Objetivo.

Tabla 2-9. Tabla Historial Objetivos de la Base de Datos.

Historial Actividades: Historial de todas las actividades en las que ha trabajado el alumno y de los resultados que ha obtenido.	
Nombre del Campo	Descripción del Campo
IdAl	Identificador del Alumno
IdAct	Identificador de la Actividad
Aciertos	Número de aciertos del alumno en la actividad.
Total	Número de cuestiones totales de las que contaba la actividad.
Tiempo	Tiempo que ha tardado el alumno en realizar la actividad (en milisegundos)
Fecha	Fecha en la que el alumno ha realizado la actividad.
Hora	Hora en la que el alumno ha realizado la actividad.

Tabla 2-10. Tabla Historial Actividades de la Base de Datos.

A continuación se muestra el diagrama entidad-relación de las tablas que han sido definidas previamente (figura 2-3).

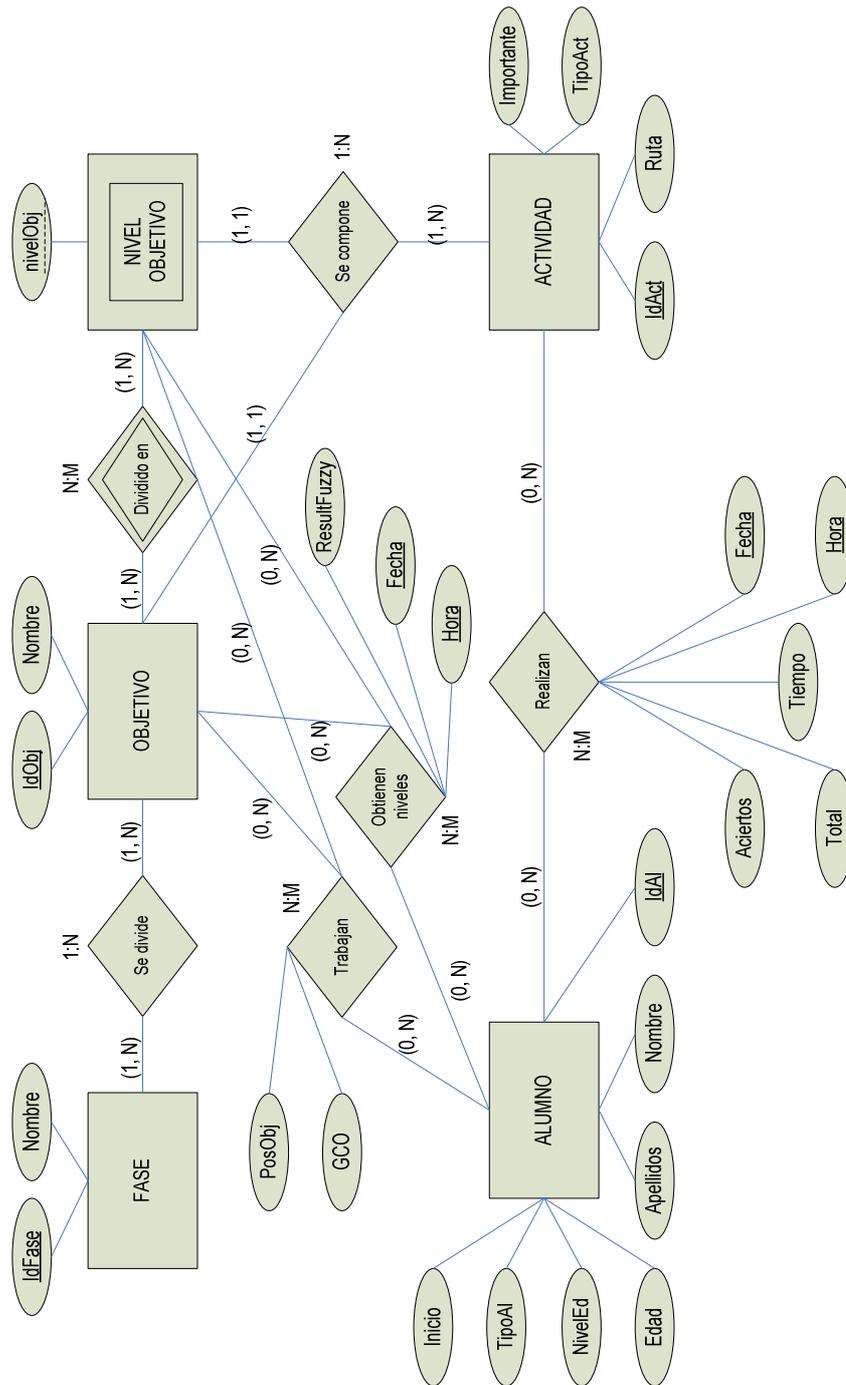


Figura 2-3. Diagrama Entidad Relación de la BD.

El Modelo Entidad-Relación es un concepto de modelado para bases de datos propuesto por [Chen 1988], mediante el cual se pretende ver gráficamente los objetos que pertenecen a la base de datos como entidades que tienen unos atributos y se vinculan mediante relaciones. Una entidad es cualquier objeto discreto sobre el que se tiene información. Se representa mediante un rectángulo etiquetado en su interior con un nombre. Una relación describe interdependencia entre entidades y se representa mediante un rombo etiquetado con un verbo. Dicho rombo debe unirse mediante líneas con las entidades que relaciona (es decir, con los rectángulos). Los atributos son propiedades relevantes propias de una entidad o de una relación que se representan mediante un círculo o elipse etiquetado con un nombre. Cuando un atributo es identificativo de la entidad se subraya dicha etiqueta.

Las entidades que participan en las relaciones pueden ser de dos tipos: Una entidad débil es aquella que no puede existir sin participar en la relación, es decir, aquella que no puede ser unívocamente identificada solamente por sus atributos. Una entidad fuerte es aquella que si puede ser identificada unívocamente. En los casos en que se requiera, se puede dar que una entidad fuerte ceda algunos de sus atributos a una entidad débil para que, esta última, se pueda identificar. Las entidades débiles se representan mediante un rectángulo con línea doble. Las relaciones también pueden ser débiles o fuertes.

Cada entidad interviene en una relación con una determinada cardinalidad. La cardinalidad (número de instancias o elementos de una entidad que pueden asociarse a un elemento de la otra entidad relacionada) se representa mediante una pareja de datos, en minúsculas, de la forma (cardinalidad mínima, cardinalidad máxima), asociada a cada uno de las entidades que intervienen en la relación. Los valores que pueden tomar las cardinalidades mínima y máxima son:

- "0" si la entidad no está obligada a participar en la relación.
- "1" si la entidad está obligada a participar en la relación y, además, cada instancia solamente participa una vez.
- "N" o "M" si la entidad no está obligada a participar en la relación y cada instancia puede participar cualquier número de veces.

El tipo de relación se define tomando los máximos de las cardinalidades que intervienen en la relación. Hay cuatro tipos posibles:

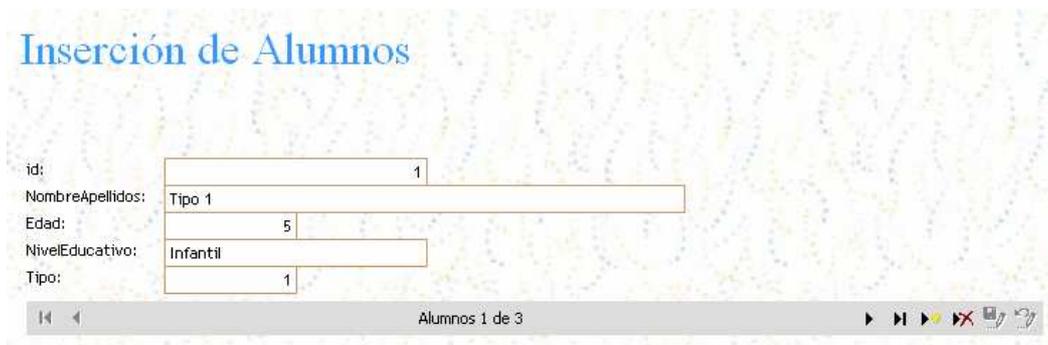
- Relaciones de una a una (1:1): una instancia de la entidad A se relaciona con una y solamente una de la entidad B (1:1).
- Relaciones de una a muchas (1:N) o de muchas a una (N:1): cada instancia de la entidad A se relaciona con varias instancias de la entidad B o viceversa.
- Relaciones de muchas a muchas (N:M): cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B.

En la base de datos que se ha diseñado para el ITS como se puede ver en la figura 2-3, las entidades son las tablas FASE, OBJETIVO, NIVEL OBJETIVO, ACTIVIDAD y ALUMNO (tablas 2-1 a 2-5). Las relaciones que se establecen entre ellas se corresponden con las tablas Clasificación Objetivos, Clasificación Actividades, Historial Borroso, Historial Objetivos e Historial Actividades que se han definido previamente (tablas 2-6 a 2-10). Los atributos y cardinalidad de cada una de ellas son las que aparecen en dicha figura. Cada FASE se divide en 1 o varios OBJETIVOS (1:N), y a su vez cada OBJETIVO se divide en 1 o varios NIVELES de OBJETIVO (1:N). Cada NIVEL OBJETIVO se compone de una o varias ACTIVIDADES (1:N). El Tutorial puede ser usado por varios ALUMNOS que trabajan simultáneamente varios OBJETIVOS (N:M), en los que van obteniendo diferentes NIVELES de dificultad del OBJETIVO para cada uno de ellos (N:M). Por último cualquier ALUMNO puede realizar cualquier ACTIVIDAD de las que se encuentran almacenadas en la base de datos (N:M).

La Base de Datos incluye además dos páginas en .html: un asistente para la inserción de alumnos en el tutorial y una página de visualización de los resultados. Con esto se pretende facilitar el uso del ITS a los profesores ya que permiten interactuar con la base de datos sin que sea necesario tener conocimiento de su funcionamiento. A continuación se mostrará una breve descripción del asistente para la inserción de alumnos en el tutorial y de la página de visualización de los resultados, encontrándose una información más detallada en el Apéndice B.

El asistente para la inserción de nuevos alumnos permite ver los alumnos que ya están inscritos en la base de datos de la aplicación, insertar nuevos alumnos, modificar sus datos personales o eliminarlos del ITS. Los campos a rellenar se ven en la figura 2-4 y son los siguientes (el campo id ya aparece relleno por defecto y no es necesario modificarlo):

- Nombre y Apellidos: Aquí se inserta el nombre y los apellidos del alumno, y son los datos que van a aparecer en el listado de alumnos de la pantalla de inicio (figura 2-5).
- Edad: La edad del alumno, debe de ser un valor numérico.
- Nivel Educativo: Hace referencia al nivel educativo en el que se encuentra el alumno. Por ejemplo: Infantil, Primaria, etc.
- Tipo de alumno: Puede contener los valores 1, 2 y 3. Un alumno tipo 1 es un alumno con miedo al fracaso, un tipo 2 es un alumno motivado y un tipo 3 es un alumno hiperactivo (estos tipos de alumnos se explicarán con más detalle en el epígrafe 4.2).



The screenshot shows a window titled "Inserción de Alumnos" with a light blue background. The form contains the following fields:

id:	1
NombreApellidos:	Tipo 1
Edad:	5
NivelEducativo:	Infantil
Tipo:	1

At the bottom of the window, there is a status bar with navigation icons on the left, the text "Alumnos 1 de 3" in the center, and more navigation icons on the right.

Figura 2-4. Pantalla de Inserción de Alumnos

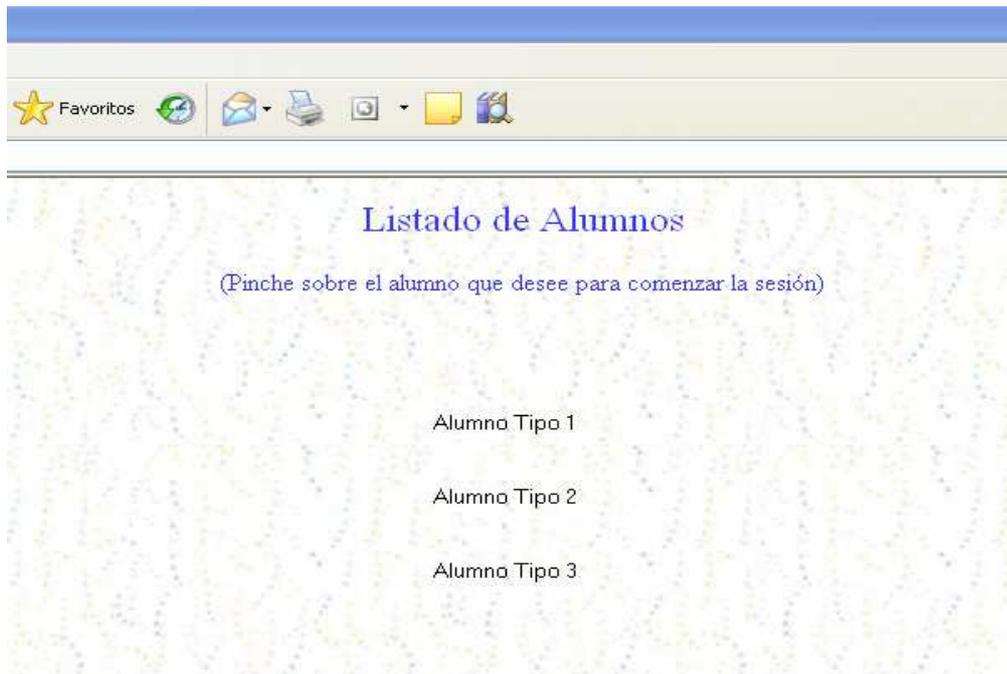


Figura 2-5. Pantalla de Inicio del ITS

La página de visualización de resultados muestra los datos personales de cada alumno y los resultados que ha obtenido en la ejecución del tutorial. Como se puede observar en la figura 2-6, el campo Fase indica en que fase se encuentra trabajando actualmente el alumno (como se ha mencionado anteriormente hay varias fases posibles). Para cada fase se muestran todos los objetivos que se trabajan en ella, indicando para cada uno, en que nivel de dificultad se encuentra el alumno. Para la fase 1 los niveles son 1 = Poco y 2 = Alto y para las restantes fases son 1 = Poco, 2 = Medio, 3 = Alto. También se muestra que porcentaje de aciertos tiene el alumno en las actividades realizadas de ese nivel.

Resultados de los alumnos

Id:	<input type="text" value="1"/>
NombreApellidos:	<input type="text" value="Alumno Tipo 1"/>
Edad:	<input type="text" value="5"/>
NivelEducativo:	<input type="text" value="Infantil"/>
Tipo:	<input type="text" value="1"/>
Fase:	<input type="text" value="2"/>

Fase 1			
Clasificación	nivel:	<input type="text" value="2"/>	Aciertos: <input type="text" value="85,00%"/>
Relaciones de Orden	nivel:	<input type="text" value="2"/>	Aciertos: <input type="text" value="80,00%"/>
Correspondencia	nivel:	<input type="text" value="2"/>	Aciertos: <input type="text" value="90,00%"/>
Cuantificadores	nivel:	<input type="text" value="2"/>	Aciertos: <input type="text" value="100,00%"/>

Fase 2			
Contar y Rec del Num	nivel:	<input type="text" value="1"/>	Aciertos: <input type="text" value="75,00%"/>
Cardinalidad	nivel:	<input type="text" value="1"/>	Aciertos: <input type="text" value="63,00%"/>
Orden	nivel:	<input type="text" value="1"/>	Aciertos: <input type="text" value="21,00%"/>
Ordinalidad	nivel:	<input type="text" value="2"/>	Aciertos: <input type="text" value="85,00%"/>
Problemas	nivel:	<input type="text" value="2"/>	Aciertos: <input type="text" value="90,00%"/>
Algoritmo	nivel:	<input type="text" value="1"/>	Aciertos: <input type="text" value="50,00%"/>

⏪ ⏩ AlumnosWiz 1 de 1 ⏪ ⏩

Figura 2-6. Pantalla de Resultados del ITS

4 INTERFAZ GRÁFICA

La interfaz gráfica controla el flujo de comunicación entre el ITS y el alumno. Es responsable de atraer y retener la atención del estudiante, y de crear un ambiente adecuado para completar el proceso de aprendizaje. Para lograr esto se deben generar pantallas que integren textos, ilustraciones, figuras, gráficas, preguntas, vídeos, audios, animaciones y diagramas; y todo ello, de una manera concreta para cada alumno. En esta Tesis Doctoral hemos decidido utilizar

tecnología web de forma que la actividad final que se le presenta al alumno se puede visualizar en un navegador y no sea necesario la instalación ni el aprender a manejar un software adicional.

Puesto que nos interesa que el ITS implementado para el refuerzo de la suma sea generalizable a cualquier dominio de aplicación, hemos creado una plantilla, es decir, una estructura básica que se pueda instanciar en tantos casos y ejercicios distintos como deseemos para interaccionar con el alumno. Las múltiples instancias de esta plantilla generarán el conjunto de actividades a presentar al estudiante. El gestor, por tanto, se encargará de comunicarle a la interfaz multimedia las especificidades de la siguiente actividad a presentar y el módulo interfaz le mostrará la actividad al estudiante. La plantilla se ha definido en XML, utilizándose Javascript para definir las interacciones [Aguilar 2003b].

En este apartado se describe en primer lugar las interacciones que se han definido en las plantillas para pasar a comentar posteriormente los aspectos motivacionales y los agentes pedagógicos que se han incluido en el ITS. Se realiza a continuación una introducción a los lenguajes de marcas y a JavaScript puesto que han sido los lenguajes elegidos para la implementación de la interfaz gráfica del ITS. Por último se muestra como han sido desarrolladas las plantillas que componen dicha interfaz.

4.1 Posibles interacciones

La respuesta del alumno a las actividades presentadas viene dada por las siguientes interacciones:

- 1. Pinchar en un objeto*
- 2. Pinchar y Mover*

La interacción de *Pinchar en un objeto* se usa para generar actividades en las que es necesario pinchar en uno, o más elementos, para que la actividad se considere correcta. Por ejemplo: "*En la pecera hay tres peces amarillos y uno rosa.*"

¿Cuántos peces hay en total?”. Para solucionar el ejercicio hay que pinchar en el número correspondiente, que en este caso es el cuatro (figura 2-7).



Figura 2-7.- Ejemplo de actividad de *Pinchar*.

La interacción de *Pinchar* y *Mover* es una variación de la interacción de *Pinchar* y *Arrastrar*, ya que se comprobó que ésta les resultaba difícil a los niños. Consiste en pinchar en un objeto, mover el objeto hasta el lugar deseado de la pantalla (sin tener pulsado el ratón como en el caso de arrastrar) y volver a pinchar para colocarlo en la ubicación deseada de la pantalla. Por ejemplo, una actividad de clasificación tiene como objetivo agrupar en cada uno de los recuadros de la parte inferior de la pantalla, aquellas imágenes que sean iguales a las indicadas en estos recuadros. En el caso mostrado en la figura 2-8, los plátanos se colocarían en la parte inferior izquierda de la pantalla y los tomates, en la inferior derecha. Una vez que se han colocado todos, hay que pulsar en el botón *Aceptar* para que se realice la validación del ejercicio. Si es correcto, se pasa al siguiente ejercicio, y si no, se vuelven a colocar los objetos en su posición inicial y se da una nueva oportunidad para empezar el ejercicio.

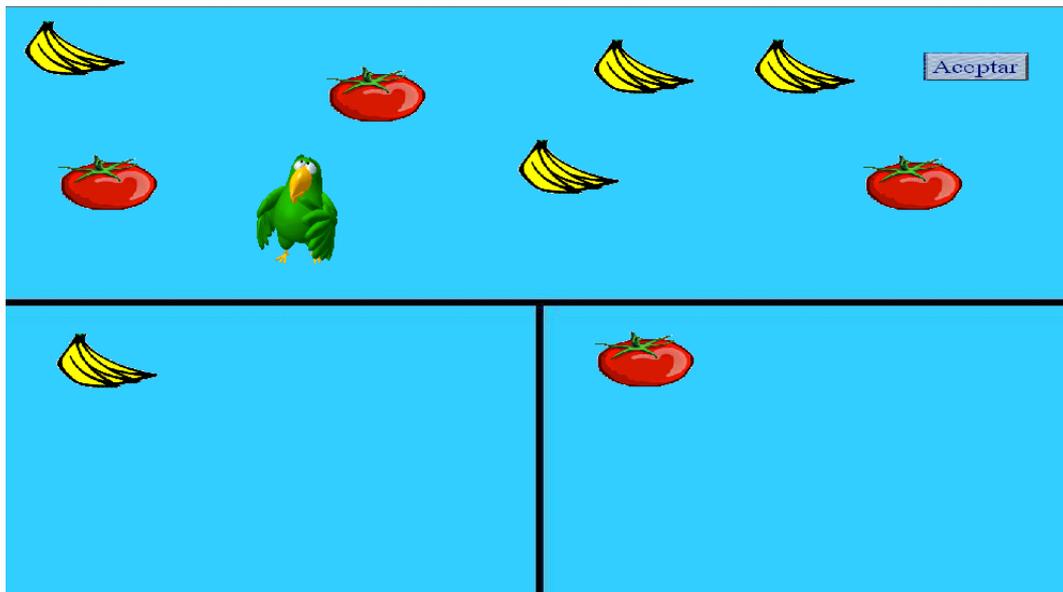


Figura 2-8.- Ejemplo de actividad de Pinchar.

Con estas interacciones se construyen una serie de plantillas para la generación de actividades que se explicarán posteriormente. Por ejemplo en la plantilla de *Ordenar* se hace uso de la interacción pinchar y mover y en la de *Hacer Parejas* se usa la interacción pinchar.

4.2 Aspectos Motivacionales

Para captar y mantener la atención y la motivación del alumno durante la ejecución del ITS, utilizamos los siguientes métodos:

1. Se distingue la personalidad de los tres tipos de alumnos lo cual permite modelizar planes de trabajo diferentes para cada uno de ellos:
 - Alumnos con miedo al fracaso que pueden presentar una resistencia a implicarse en tareas que perciben complicadas. Sobre todo los niños más pequeños pueden manifestar conductas estereotipadas para evadirse de una tarea que perciben demasiado difícil [Bevington 1999]. El objetivo es tratar de plantearle una oferta

atractiva, graduando muy bien los pasos y reforzando sus logros. Así, el éxito en la tarea les da seguridad e incrementa su motivación, proporcionando una mejor disposición a abordar tareas más complicadas.

- Alumnos hiperactivos: son alumnos que tienen dificultades para centrar la atención ante un estímulo. Esto hace que su atención se desplace lentamente de unos aspectos del estímulo a otros, precisando más tiempo para reaccionar ante los estímulos. Además tienen dificultades para retener sus respuestas hasta haber analizado la situación, lo que conduce a manifestar una conducta impulsiva [Hallowell 1996]. Otra dificultad de atención proviene por su conducta dispersa, centran la atención en los aspectos menos relevantes de la situación, olvidando los más significativos.
 - Alumnos motivados con ganas de trabajar para los que las estrategias didácticas estándares son adecuadas a sus procesos de aprendizaje (no se engloba en ninguno de los grupos anteriores) y tienen interés por las cosas nuevas.
2. Incorporamos agentes pedagógicos, a presentar durante la explicación de la actividad a realizar en cada objetivo. Este agente es escogido de una galería de agentes preestablecida, en función de la edad cognitiva y la madurativa (ver apartado 4.3).
 3. Dotamos al sistema de diferentes estímulos alternativos para presentar una misma actividad de modo que se evite la repetición de presentaciones idénticas.

4.3 Agentes Pedagógicos

Dentro del uso de la multimedia con fines educativos, una de las líneas sobre las que se está investigando estos últimos años es la incorporación de agentes pedagógicos al proceso de aprendizaje. Podemos definir los agentes pedagógicos como "caracteres naturales en pantalla, que proveen de consejos contextualizados y de realimentación durante todo el proceso de aprendizaje" [Lester 1997].

En un estudio realizado en 1996 [Reeves 1996] con el objetivo de conocer el efecto de los agentes sobre el aprendizaje se demostró que su presencia constituye un elemento motivador que produce un incremento en el rendimiento de aprendizaje. Siguiendo esta misma línea, en un estudio realizado en el año 2000 [Mayer 2000] se exponen una serie de resultados que refuerzan la idea de que la eficacia en el aprendizaje aumenta cuando existe algún agente pedagógico que interactúa con el alumno y dirige el proceso de aprendizaje.

Los agentes pedagógicos se encargan de interactuar con los alumnos y explicarles la actividad que tienen que realizar, por lo que interactúan y cooperan con el alumno de una manera natural para que una actividad sea ejecutada. De esta manera, se encargan de presentar el problema, guiar la ejecución de la actividad y presentar estímulos positivos y/o negativos (feedbacks) de acuerdo a los resultados obtenidos por el alumno en la resolución de la actividad. A través de la utilización de estos agentes, la motivación del alumno se ve incrementada, ya que lo ven como una interacción social.

Los agentes han sido escogidos de entre una galería de agentes preestablecida, en función de la edad cognitiva y la madurativa. También hay que tener en cuenta que todos los agentes no son adecuados para todas las situaciones de aprendizaje ni para todos los contenidos a enseñar. Las capacidades físicas del agente (cómo se desplaza por la pantalla, cómo gesticula) y las funciones que puede ejecutar (cómo indica un objeto o muestra un concepto) deben ser coherentes con el contenido a enseñar y el modo de presentación de la información. Por ejemplo podemos considerar que un agente pedagógico que represente a un perro no sería el adecuado a la hora de explicar una tarea donde se necesita que el agente pedagógico se desplace en la pantalla entre un árbol y el suelo, para explicar los conceptos de arriba y abajo. Sería más lógico entonces usar un agente que pudiera volar (un pájaro, un genio, etc.).

Los agentes pedagógicos que se han elegido (aunque siempre con la posibilidad de inclusión de cualquier nuevo agente) se pueden observar en la figura 2-9.



Figura 2-9. Agentes Pedagógicos

A la hora de incluir a los agentes en las actividades, inicialmente se utilizó tecnología TTS (Text to Speech, o lo que es lo mismo, la traducción del texto a voz sintetizada) ya que ésta hacía posible que los agentes reprodujeran textos introducidos en el ordenador. Estos textos son distintas variables que soportan las instrucciones, los refuerzos verbales tanto positivos como negativos y variables personales del alumno como su nombre. Así mediante esta transformación de texto a voz sintetizada se puede crear un lenguaje personalizado para un determinado alumno. Pero una vez realizada una prueba con los niños de la Asociación Tinerfeña de Trisómicos 21 (ATT21) se comprobó que el agente hablaba demasiado rápido y que los alumnos no le entendían bien. Como consecuencia de esto, se decidieron grabar las voces correspondientes a los agentes y reproducirlas en formato wav (Waveform Audio File, formato de audio digital sin compresión de datos) durante la ejecución del tutorial.

Los agentes pedagógicos pueden desplazarse por la pantalla y realizar varios gestos y acciones. Éstas pueden ser por ejemplo:

- Greet: Saludar cuando comienza la sesión.
- Explain: Explicar el ejercicio.
- Think: Pensar mientras el alumno está resolviendo la actividad.
- Congratulate: Felicitar al alumno cuando resuelve el ejercicio correctamente (positive feedback).
- Decline: Corregir al alumno cuando se equivoca (negative feedback).
- GestureAt: Para señalar algún elemento u objeto de la pantalla.
- Etc.

Además de la voz y los gestos, otros factores importantes son las emociones que podemos incorporarles a estos agentes, por lo que están programados para comportarse según las acciones cometidas por los usuarios. Por ejemplo, si el alumno se equivoca, el agente se comporta con tristeza o desconcierto y le pide al usuario una nueva interacción. En el caso contrario (respuesta correcta), los agentes saltan de alegría, aplauden, les dan algún premio y le muestran una nueva actividad [Towns 1998]. Cada agente posee una personalidad, por lo que las animaciones asociadas a estas emociones difieren en cada personaje en particular.

4.4 Introducción a los Lenguajes de Marcas y JavaScript

Para el desarrollo de la Interfaz Gráfica se ha decidido usar XML que es un lenguaje de marcas. Dichos lenguajes permiten codificar un documento de manera que además del texto se pueden incorporar etiquetas (o marcas) que contienen información adicional sobre la estructura de dicho texto o sobre su presentación [Sperberg 1999].

Existen tres clases de lenguajes de marcas:

- Marcado de presentación, que indica el formato del texto.
- Marcado de procedimientos, que se enfoca hacia la presentación del texto pero que también es visible para el usuario que edite el texto. El programa software en el que se quiera visualizar el documento tiene que interpretar el código en el orden en el que aparezca.
- Marcado descriptivo, que utiliza etiquetas para describir las partes del texto pero que no especifica como hay que representar dicho texto o en que orden hay que hacerlo.

Existen diferentes lenguajes de marcas que se van a describir brevemente a continuación.

4.4.1 SGML

El SGML (Standard Generalized Markup Language) es un lenguaje de marcas que se caracteriza porque hace más énfasis en el marcado descriptivo que en el de procedimientos. Además se ha diseñado para ser independiente del sistema en el que se represente la codificación, lo que permite que pueda ser procesado en ordenadores con diferente hardware y utilizando diferente software sin que se pierda información.

Este estándar introduce la definición de las DTD que se explicarán en el apartado siguiente (4.4.2).

El principal inconveniente de SGML es que es complicado de utilizar ya que fue diseñado para sistemas grandes y es excesivo para las necesidades de un usuario estándar. En la figura 2-10 se puede ver un pequeño ejemplo de cómo se definirían los alumnos que forman parte de una asignatura (incluyendo la nota que han sacado en ella) en SGML.

```
<asignatura>
  <nombre>Física
  <alumno>
    <nombreapellidos>Juan Martín
    <nota>7.5
  </alumno>
    <nombreapellidos>María González
    <nota>9.0
</asignatura>
```

Figura 2-10. Ejemplo de SGML.

4.4.2 DTD (DOCUMENT TYPE DEFINITION)

Las DTDs se crearon para la descripción de la estructura y sintaxis de un documento SGML. Su función es la de describir el formato de los datos texto para que todos los documentos que se asocien a una misma DTD tengan el mismo formato. La estructura del documento se determina mediante etiquetas. En la DTD se describen:

- Elementos: indican las etiquetas permitidas y su contenido.
- Estructura: indica el orden de las etiquetas.
- Anidamiento: define la anidación entre etiquetas (cuales van dentro de otras).

En la figura 2-11 está definida la DTD para el ejemplo definido en SGML del apartado 4.4.1.

<!ELEMENT	asignatura	- -	(nombre?, alumno+)>
<!ELEMENT	nombre	- 0	(#PCDATA)>
<!ELEMENT	alumno	- 0	(nombreapellidos, nota+)>
<!ELEMENT	nombreapellidos	- 0	(#PCDATA)>
<!ELEMENT	nota	- 0	(#PCDATA)>

Figura 2-11. Ejemplo de DTD.

4.4.3 HTML (LENGUAJE DE MARCAS DE HIPERTEXTO)

HTML es un lenguaje de marcas derivado del SGML que se usa principalmente en la construcción de páginas web. A diferencia del lenguaje del que procede el conjunto de etiquetas que se pueden emplear en un documento HTML es finito por lo que su uso es más sencillo y está más generalizado. Aún así presenta una serie de inconvenientes como son que no tiene etiquetas identificativas del idioma, es necesaria la sustitución a mano de los vínculos en caso de modificación y su validación (comprobación de que el documento ha creado correctamente) es relativa. Esto último significa que aunque un documento HTML se pueda validar en cuanto al correcto uso de las etiquetas no se puede hacer lo mismo para la presentación de contenido. En la figura 2-12 se puede observar el mismo ejemplo del apartado 4.4.1, pero en HTML.

```
<html>
  <head>
    <title>Física</title>
  </head>
  <body>
    <p>Juan Martín<b> 7.5 </b></p>
    <p>María González<b> 9.0 </b></p>
  </body>
</html>
```

Figura 2-12. Ejemplo de HTML.

4.4.4 XML (LENGUAJE DE MARCAS EXTENSIBLE)

El lenguaje de marcas XML se comenzó a desarrollar en 1996 por el World Wide Web Consortium. Con él se intentaba crear un lenguaje de marcas que fuera intermedio entre SGML y HTML, es decir que fuese más fácil que SGML pero más rígido que HTML. Los objetivos que se pretendían cumplir con este lenguaje son que [Sturm 2000]:

- Se utilice directamente en Internet.
- Pueda ser utilizado por un gran número de aplicaciones.
- Sea compatible con SGML.
- Sea fácil escribir con él programas de procesamiento del lenguaje.
- No tenga demasiadas características opcionales.
- Sea lo más legible posible.
- Tenga un diseño formal y a la vez conciso.
- La creación de documentos con él sea fácil.

El lenguaje XML tiene las siguientes ventajas:

- Puede incluir caracteres de múltiples idiomas.
- Puede utilizarse una DTD o un XML Schema (se explicará en el apartado 4.4.4.2) para estructurar el documento XML.
- Mediante métodos de vinculación se pueden construir documentos XML por composición de otros documentos.
- Se puede usar para contener datos.

En la figura 2-13 se puede ver el ejemplo del apartado 4.4.1 en código XML.

Por todas estas ventajas, se ha decidido implementar las plantillas para la interfaz del Tutorial en el lenguaje XML.

```
<?xml versión="1.0" ?>
<asignatura nombre="Física">
  <alumno>
    <nombreapellidos>Juan Martín</nombreapellidos>
    <nota>7.5</nota>
  </alumno>
  <alumno>
    <nombreapellidos>María González</nombreapellidos>
    <nota>9.0</nota>
  </alumno>
</asignatura>
```

Figura 2-13. Ejemplo de XML.

4.4.4.1 XSL y XSLT

Dado que el lenguaje XML está pensado para el almacenamiento y la transmisión de datos y no para su visualización se han creado las hojas de estilo XSL (eXtensible Stylesheet Lenguaje). Además estas hojas de estilo permiten elegir que datos se muestran y cómo.

En las XSL se puede definir un estilo para cada elemento del documento XML con la ventaja de que si se quiere cambiar algún tipo de elemento no hay que hacerlo uno a uno en el documento, si no que cambiando únicamente la hoja de estilo se cambia la apariencia de todo el documento.

La hoja de estilo XSL sólo es una herramienta de visualización de documentos XML. Para además poder transformar dicho documento en cualquier otro basado en texto (por ejemplo HTML) es necesario usar un lenguaje XSLT (XSL Transformations).

En la figura 2-14 se ha definido un XSL para el ejemplo XML del apartado 4.4.4.

```
<?xml version="1.0" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
<xsl:template match="/">
  <xsl:template match="asignatura">
    <html><head><title><xsl:value-of select="nombre"/></title></head>
    <body>
      <xsl:template match="asignatura/alumno">
        <p><xsl:value-of select="nombreakapellidos"/>
        <b><xsl:value-of select="nota"/></b></p>
      </body>
    </html>
  </xsl:template>
</xsl:template>
</xsl:stylesheet>
```

Figura 2-14. Ejemplo de XSL.

4.4.4.2 XML Schemas

Los XML Schemas Definition (XSD, que se corresponden con ficheros de extensión .xsd) fijan una serie de reglas que describen el contenido del documento XML. Esto permite estructurar la información del mismo, de forma que cuando un documento XML cumple con las reglas de un XML Schema se dice que es válido o que está validado. El proceso de comprobar si un documento XML es válido respecto a un XML Schema lo que se denomina validación del documento.

En un XML Schema se definen:

- Los elementos y atributos que pueden aparecer en el documento XML.
- Qué elementos son hijos de un elemento padre, cual es el orden en el que deben aparecer en el documento y en que cantidad.
- Tipo de datos (numérico, texto, etc.) para los elementos y los atributos.
- Valores por defectos que pueden tomar los elementos así como valores constantes si los hay.

Un XML Schema tiene las siguientes ventajas frente al uso de una DTD. Estas son las siguientes:

- Los XML Schema son más fáciles de ampliar y modificar que las DTDs.
- Los XML Schema tienen más tipos de datos que las DTDs (string, date, boolean, ...), ya que en éstas sólo pueden usarse las palabras claves como CDATA, NUMBER, ...
- Los XML Schema están escritos en XML por lo que no hay que aprender ningún otro lenguaje mientras que las DTDs tienen el suyo propio.
- Los XML Schema permiten la definición de nuevos tipos de datos por parte del usuario (las DTDs no).
- Los XML Schema permiten definir elementos de igual nombre siempre que vayan precedidos por prefijos diferentes.

La figura 2-15 muestra un ejemplo de XML Schema para el ejemplo XML del apartado 4.4.4.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="asignatura">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="nombre" type="xs:string"/>
      <xs:element name="alumno">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="nombreapellidos" type="xs:string"/>
            <xs:element name="nota" type="xs:decimal"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Figura 2-15. Ejemplo de XML Schema.

4.4.5 JAVASCRIPT

JavaScript es un lenguaje interpretado (no necesita ser compilado) usado en páginas web [Java 2006]. Fue creado por Brendan Eich para la empresa Netscape Communications. Sun Microsystems y Netscape le dieron este nombre en 1995 ya que antes su desarrollador lo había llamado Mocha y posteriormente LiveScript.

Aunque su sintaxis es similar al lenguaje Java no es un lenguaje orientado a objetos ya que las clases no pueden derivar unas de otras (no existe herencia). Los navegadores interpretan el código JavaScript que se encuentra integrado dentro de las páginas web. JavaScript se ejecuta en modo cliente (en la máquina del usuario) ya que sus sentencias se van descargando al mismo tiempo que el HTML.

Javascript se puede incluir en cualquier documento HTML o en todo aquel que termine traducándose en HTML en el navegador del cliente, por ejemplo XML, PHP, ASP, SVG, etc. Permite controlar los eventos que se producen en el sistema por la interacción del usuario.

En la figura 2-16 se puede ver un sencillo ejemplo de uso de Javascript en una página HTML. En ella se escribe "Hola Mundo" por pantalla.

```
<HTML>
<HEAD><TITLE>Ejemplo</TITLE></HEAD>
<BODY>
<H1>Ejemplo en JavaScript</H1>
<script type="text/javascript">
    document.write("Hola Mundo");
</script>
</BODY>
</HTML>
```

Figura 2-16. Ejemplo de JavaScript.

4.5 Definición de la Plantilla XML

La plantilla XML que se ha implementado para instanciar las distintas actividades, está basada en tres elementos fundamentales:

1. *Imagen* donde se describen todas las imágenes que aparecerán en la página web.
2. *Agente* que describe el agente pedagógico que interactúa con el alumno en la explicación y refuerzo de la tarea, y
3. *Web* que indica la siguiente actividad a presentar al alumno una vez finalizada el trabajo actual.

Para ello se ha definido el XML Schema que se ve representado en la figura 2-17. Un ejemplo de lo que significan los términos *Imagen*, *Agente* y *Web* se puede observar en la figura 2-18. Iremos desarrollando dichos términos a lo largo de este epígrafe.

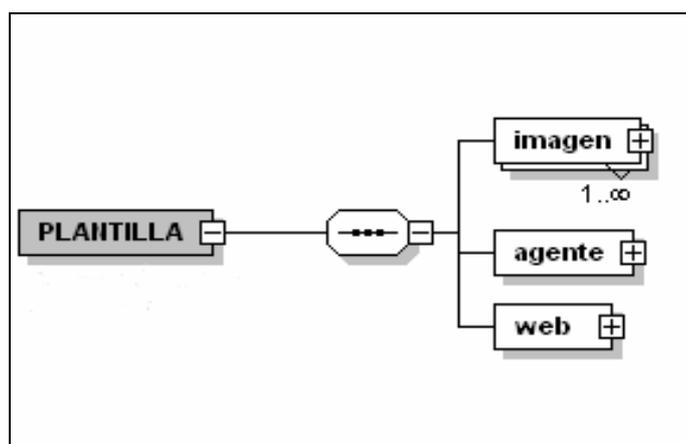


Figura 2-17.- Los tres elementos básicos de la plantilla XML que genera la interfaz multimedia del ITS.

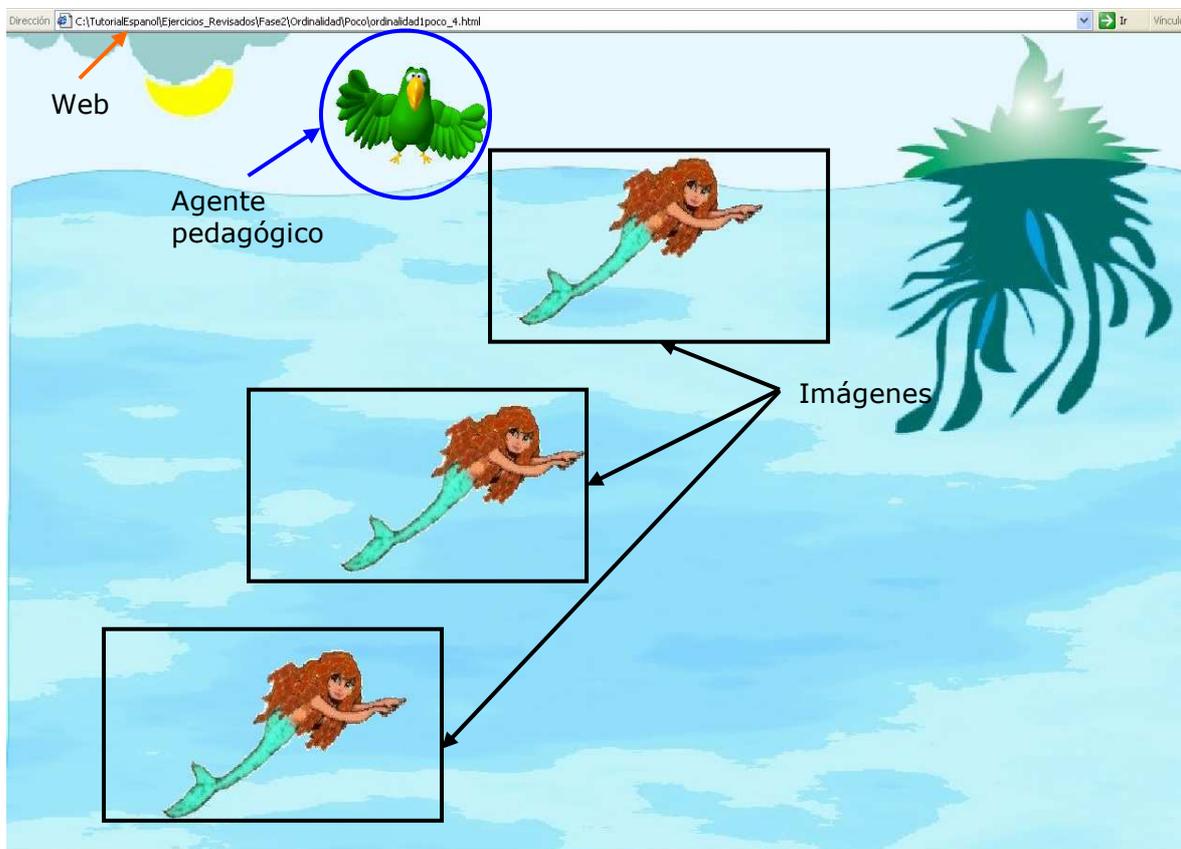


Figura 2-18. Ejemplo de Imágenes, Agente Pedagógico y Web en una actividad.

El XML Schema del elemento *Imagen* está compuesto de los siguientes atributos que se pueden observar en la figura 2-19:

- *Id*: Es el nombre o identificador de la imagen (por ejemplo: fondo, imagen1, imagen2, etc). El identificador es un valor único para cada imagen, por lo que no puede haber dos imágenes que tengan el mismo identificador.
- *Posición Inicial Ix*: Coordenada en el eje X de la pantalla donde se sitúa la imagen inicialmente. Son coordenadas relativas respecto a las dimensiones de la pantalla. El valor debe ser un número entre 0 (parte izquierda de la pantalla) y 100 (parte derecha de la pantalla).

- *Posición Inicial Iy*: Igual que la anterior, pero en el eje Y.
- *Posición Final Ax1*: Este campo indica la posición inicial del área en el que la imagen se considera correcta en el eje X. El valor debe ser un número entre 0 (parte izquierda de la pantalla) y 100 (parte derecha de la pantalla)
- *Posición Final Ax2*: En este campo hay que indicar el ancho del área correcta en porcentaje relativo respecto al tamaño de la pantalla. El valor debe ser un número entre 0 y 100.
- *Posición Final Ay1*: Igual que Ax1, pero en el eje Y.
- *Posición Final Ay2*: Igual que Ax2, pero hace referencia al alto del área.
- *Tamaño Alto*: Altura de la imagen en porcentaje relativo respecto al tamaño de la pantalla. El valor debe ser un número entre 0 y 100.
- *Tamaño Ancho*: Anchura de la imagen en porcentaje relativo respecto al tamaño de la pantalla. El valor debe ser un número entre 0 y 100.
- *Ruta*: Nombre del fichero (incluyendo extensión) que contiene la imagen.
- *Fijo*: Indica si la imagen es solución o no. En las actividades de pinchar en un objeto, el campo fijo debe estar activado cuando la imagen no sea solución al problema. Es decir, que cuando la imagen sea solución de la actividad no se tendrá activado el valor fijo y habrá que activarlo en cualquier otro caso (cuando una imagen sea fondo siempre se debe activar también el valor fijo).
- *Fondo*: Indica que la imagen es el fondo de la actividad. Puede haber uno o varios fondos. Por lo tanto, el campo fondo debe estar activado cuando la imagen sea el fondo de la actividad.
- *Pareja*: En el que hay que escribir el Id de la imagen con la que queremos emparejar la actual.
- *Orden*: Indica qué posición tiene que ocupar el objeto en una serie ordenada.
- *Cantidad*: Indica que cantidad de objetos se tienen que colocar dentro de un área.

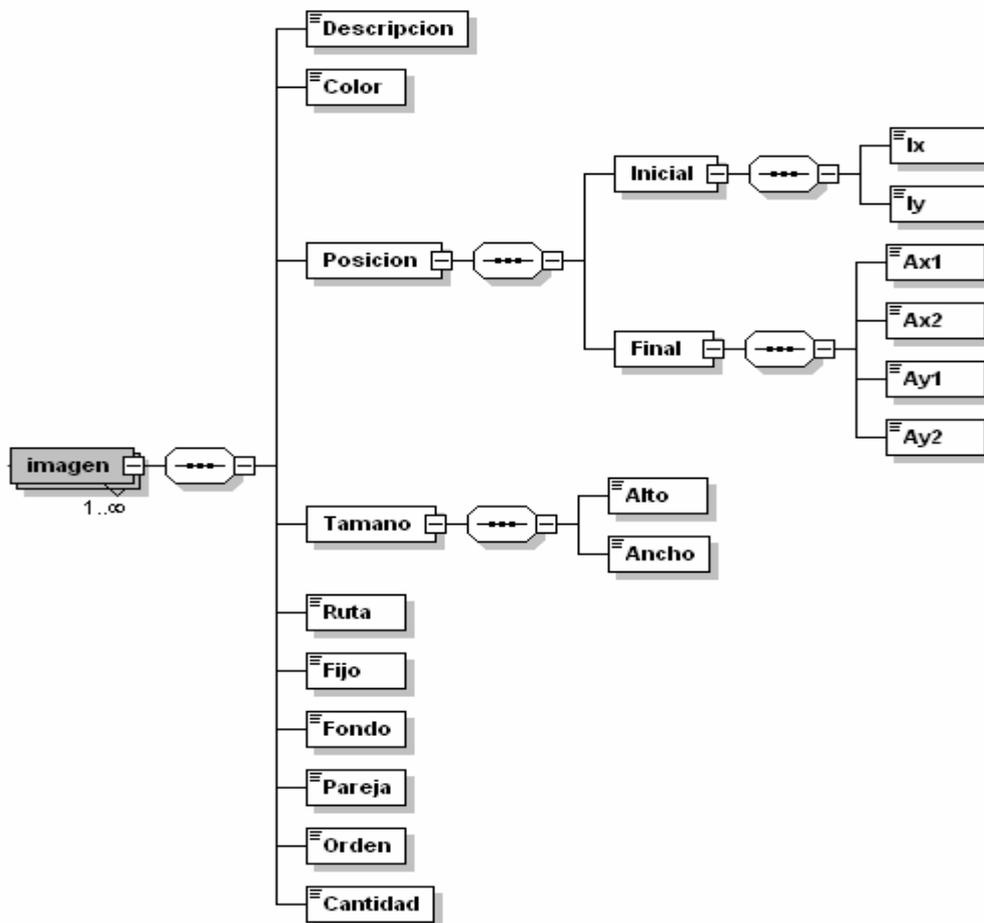


Figura 2-19.- Elemento imagen.

El siguiente elemento es *Agente* cuyos atributos son (figura 2-20):

- *Nombre*: Indica el agente pedagógico que se quiere utilizar en la actividad (con extensión .acs).
- *Explicación Id*: Es el identificador de la acción del agente (por ejemplo: accion1, accion2, explicacion1, saludo1, etc). El identificador es un valor único para cada acción por lo que no puede haber dos que tengan el mismo identificador.
- *Explicación MoveTo PosiciónX*: Este campo indica la posición del agente en el eje X. El valor debe ser un numero entre 0 (parte izquierda de la pantalla) y 100 (parte derecha de la pantalla).

- *Explicación MoveTo PosiciónY*: Igual que campo anterior, pero en el Eje Y en lugar de en el X.
- *Play*: Indica la interacción a realizar por el agente. Por ejemplo: *Greet* (el agente saluda cuando comienza la actividad), *Explain* (explicación del ejercicio), *Think* (el agente se pone a pensar mientras espera que el alumno resuelva el ejercicio), etc.
- *Explicación Speak*: Es el mensaje que reproducirá el agente durante la acción elegida. Como se está usando tecnología *wav*, hay que escribir la ruta del fichero que contiene el audio que puede ser absoluta o relativa.
- *Felicitación*: Es el texto que le dice el agente al alumno cuando resuelve correctamente el ejercicio. Igual que el caso anterior hay que escribir la ruta al fichero de audio.
- *Corrección*: Es el texto con el que el agente pedagógico corrige al niño cuando realiza mal el ejercicio (hay que escribir la ruta del audio igual que en los elementos anteriores).

Por último el elemento *Web* sólo contiene el elemento *Siguiente* (figura 2-20) que se usa para indicar la siguiente actividad que se va a mostrar al alumno cuando se resuelva la que se está creando, en el caso de que se quiera hacer una secuenciación de actividades. Se debe escribir la ruta completa de la siguiente actividad (o relativa si están todas en el mismo directorio) teniendo que incluirse la extensión *.html*. Si no se quiere encadenar con otra actividad sino que se ejecute de forma independiente se tiene que rellenar el campo con el texto *Ninguna*.

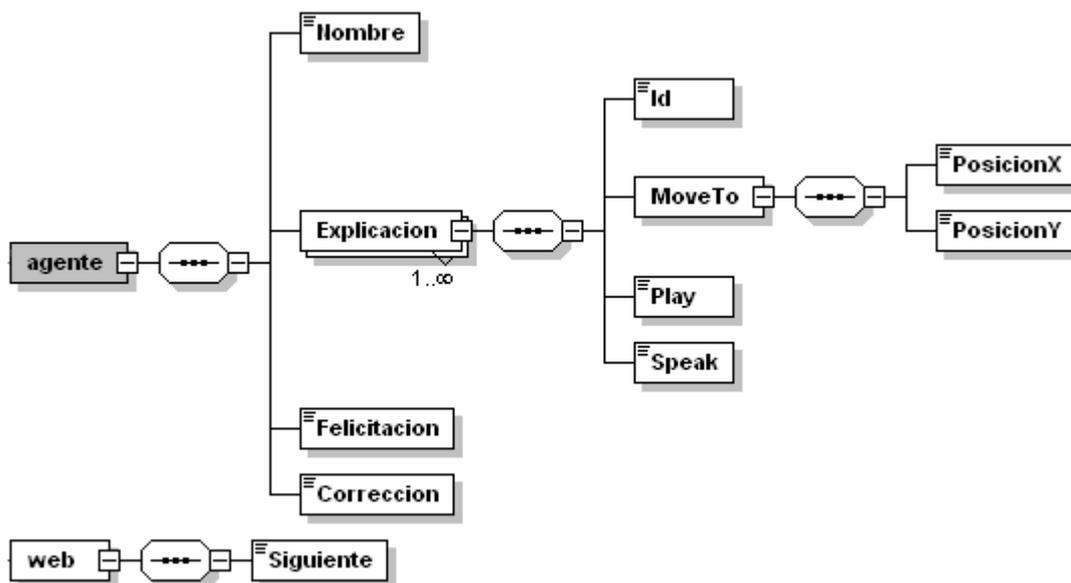


Figura 2-20.- Elementos agente y web.

Aunque como se ha visto anteriormente sólo se ha definido un XML Schema no todos los campos se usan para cada interacción. Es decir, dependiendo del tipo de interacción o de ejercicio, se utilizarán unos atributos u otros. Por ejemplo, en el caso del elemento imagen, el atributo pareja sólo se usa en actividades que necesiten emparejar elementos. El atributo Posición Final solamente se usa en la interacción de pinchar y mover puesto que es el campo que hemos definido para indicar el área de pantalla correspondiente a la solución correcta del ejercicio. En el Apéndice D se puede ver el código del XML Schema del Tutorial.

En la instanciación de la plantilla, se utiliza un XSLT que incluye JavaScript para definir las interacciones que puede utilizar el alumno [Aguilar 2005a]. El pseudocódigo de ese programa se puede observar en la figura 2-21. En primer lugar se inicializan las imágenes y el agente. Para colocar cada una de las imágenes definidas en el XML en la pantalla se calculan las dimensiones de la misma, se escalan las imágenes y finalmente se muestran poniéndolas en capas superpuestas empezando por el fondo de la imagen. Al mostrarse el agente, éste le explica al niño la actividad que tiene que realizar y se queda esperando la respuesta del alumno. Cuando se produce la interacción (pinchar o pinchar y mover) se comprueba si la respuesta del alumno es correcta o no. Si es correcta el agente

felicita al alumno y se le muestra otra actividad en el caso de estarse trabajando una secuencia de actividades o se finaliza (si era la última actividad o se estaban usando las actividades independientemente). En el caso de que la respuesta del alumno no fuera correcta se le indica su error y se le da otra oportunidad para resolver el ejercicio.

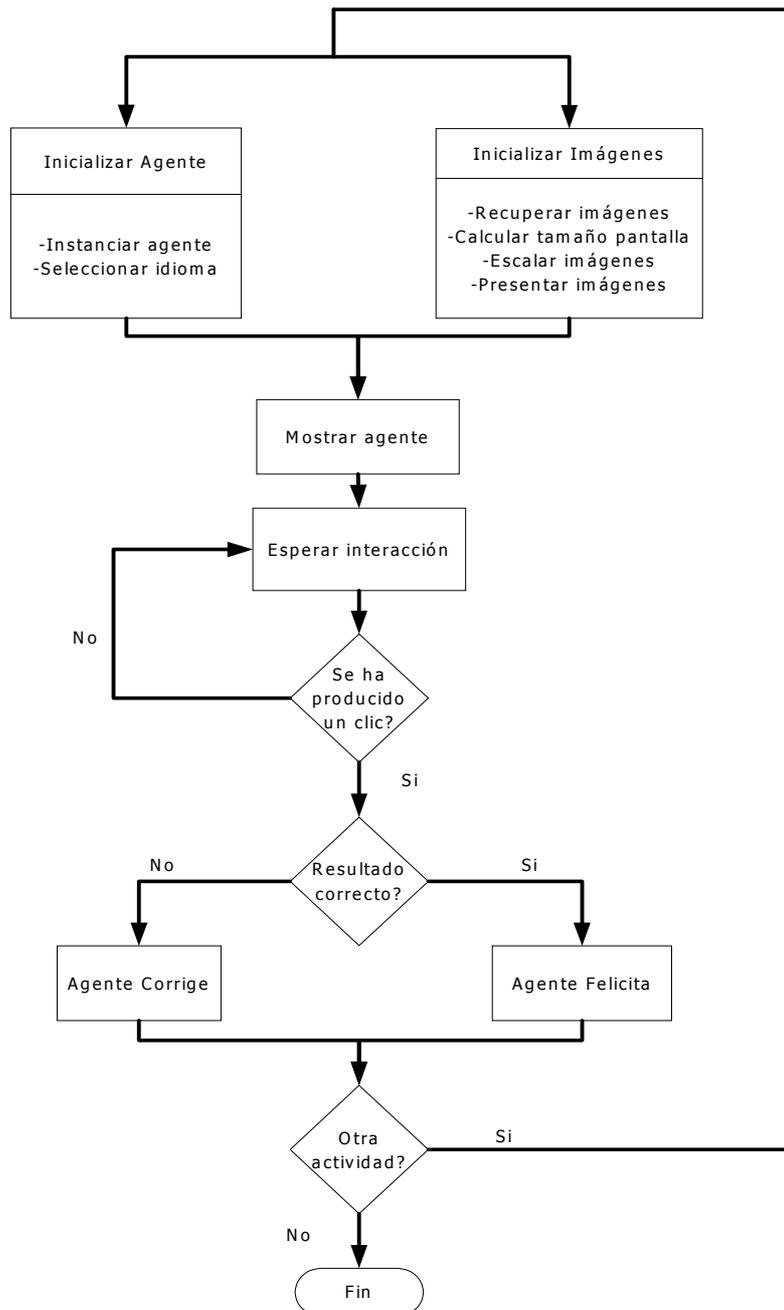


Figura 2-21.- Pseudocódigo de la transformación utilizada en la XSLT.

En el Apéndice E se incluye como ejemplo el XSLT de una plantilla de tipo Pinchar. Las actividades del Tutorial han sido convertidas a html para mejorar el rendimiento del programa evitando los tiempos de interpretación del código.

4.5.1 EJEMPLO DE ACTIVIDAD DE PINCHAR

Como ejemplo vamos a ver una actividad del tipo Pinchar correspondiente al objetivo de clasificación. La descripción completa de todos los tipos de actividades que se han creado se puede ver en el Apéndice A.

En primer lugar hay que introducir los datos de entrada correspondientes al elemento imagen de la plantilla XML que en este caso son los que se pueden ver en la figura 2-22. Cada una de las filas de la tabla imagen se corresponde con una imagen que aparecerá en la página web final.

Imagen (6)	Descripcion	Posicion	Tamano	Ruta	Fijo	Fondo
1	fondo	Posicion Inicial lx: 0 ly: 0 Final	Tamano Alto: 100 Ancho: 100	fondocielo1.gif	true	true
2	gato2	Posicion Inicial lx: 65 ly: 70 Final	Tamano Alto: 25 Ancho: 20	gato2.gif	true	false
3	gato8	Posicion Inicial lx: 40 ly: 40 Final	Tamano Alto: 25 Ancho: 20	gato8.gif	true	false
4	gato9	Posicion Inicial lx: 75 ly: 40 Final	Tamano Alto: 25 Ancho: 15	gato9.gif	true	false
5	perro7	Posicion Inicial lx: 10 ly: 30 Final	Tamano Alto: 35 Ancho: 15	perro7.gif	false	false
6	gato10	Posicion Inicial lx: 25 ly: 70 Final	Tamano Alto: 25 Ancho: 15	gato10.gif	true	false

Figura 2-22.- Datos del elemento imagen para una plantilla de tipo pinchar.

Vamos a ir viendo lo que significan cada uno de los atributos para este ejemplo concreto. Esta es una actividad de clasificación en la que hay que pinchar la imagen

que no pertenece al conjunto. Se van a mostrar cuatro gatos y un perro sobre una pradera, teniendo el alumno que pinchar en el perro para solucionar correctamente el ejercicio. Los atributos que no se usan en este ejemplo (color, pareja, etc.) se han ocultado para facilitar la visualización y se explicarán en ejemplos posteriores.

En el caso de la estructura figuran:

- La *Descripción* de la imagen que se correspondería con el nombre con el que vamos identificar a ésta (fondo, gato2, perro7, etc.).
- El atributo *Posición* que consta de dos valores: *Inicial* y *Final*. El *Inicial* nos indica en qué posición de la pantalla va a estar situada la imagen. Como ya se indicó anteriormente *Posición Final* no se usa en este ejercicio ya que éste usa interacción del tipo pinchar en un objeto.
- *Tamaño* es otro atributo doble que consta a su vez de los campos *Ancho* y *Alto*. Estos definen el tamaño con el que se va a dimensionar la imagen. Tanto la posición como las dimensiones se dan en valores relativos, puesto que nos interesa que el ejercicio se pueda visualizar correctamente con cualquier resolución de pantalla. Por ejemplo, como se puede observar en la figura 7 el fondo de la actividad comienza en la *Posición Inicial* $I_x=0$, $I_y=0$ y ocupa toda la pantalla ya que *Alto* y *Ancho* tienen como valor 100. En cambio la imagen gato2 está situada en la *Posición Inicial* $I_x=65$, $I_y=72$ y su tamaño es de el 25% de la pantalla en altura y el 20% en anchura.
- Las *Rutas* de las imágenes en este caso son: fondo1.gif, gato2.gif, gato10.gif, etc.
- El atributo *Fijo* en esta interacción tiene el valor *true* por defecto en todas las imágenes excepto en aquella que se considera correcta que en este caso es el perro7 que está a *false*.
- El atributo *Fondo* en cambio está a *false* en todas las imágenes menos en la primera que es el fondo de la actividad.

Introducimos ahora los datos correspondientes a los elementos *Agente* y *Web* (Figura 2-23).

▲ agente	⊗ Nombre	peedy.acs
	▲ Explicacion	
	⊗ Id	1
	▲ MoveTo	
	⊗ PosicionX	10
	⊗ PosicionY	20
	⊗ Play	Explain
	⊗ Speak	clasificacion1.wav
	⊗ Felicitacion	congratulate.wav
	⊗ Correccion	decline.wav
▲ web	⊗ Siguiente	Ninguna

Figura 2-23.- Datos de los elementos agente y web para una plantilla de tipo pinchar.

En cuanto al *Agente* nos encontramos que los datos de entrada que se deben rellenar en la plantilla XML son:

- El *Nombre* del agente (con extensión .acs).
- Después encontramos un atributo múltiple correspondiente a la explicación del ejercicio. En el se definen las distintas posiciones de pantalla a las que debería desplazarse el agente (*MoveTo*), la acción a realizar (*Play*) y el texto que se usará para explicar el ejercicio (*Speak*).
- La *Felicitación* es el texto a decir por el agente, en el caso de que el ejercicio se resuelva correctamente (realimentación positiva).
- Y *Corrección*, es el caso contrario, la realimentación negativa.

Como por el momento no queremos encadenar esta actividad con ninguna otra en el campo *Siguiente* del elemento *Web* aparece el valor por defecto que es: *Ninguna*.

Una vez que ya se han definido todos los datos de entrada, se le pasan al XSLT correspondiente a la actividad de pinchar, que es la que contiene el código JavaScript que usaremos para generar el fichero *.html* resultante. El resultado completo de este ejemplo concreto sería el que se puede observar en la figura 2-24.



Figura 2-24.- Actividad resultante de tipo pinchar.

4.5.2 EJEMPLO DE ACTIVIDAD DE PINCHAR Y MOVER

El objetivo de estas actividades es que dada una serie de objetos colocados en posiciones variables de la pantalla, éstos se tengan que colocar en el lugar que se indique.

En las actividades de *Pinchar* y *Mover* hay un botón de *Aceptar* en la parte superior derecha de la pantalla. Una vez que se han movido los objetos y se han colocado en el lugar deseado, hay que presionar dicho botón para que la actividad sea evaluada.

Para explicar la plantilla de este tipo de actividad sólo vamos a comentar en el ejemplo los campos nuevos o los que cambian con respecto a las plantillas de *Pinchar*. Esta actividad de clasificación consiste en que dadas una serie de imágenes que se encuentran en la parte superior de la pantalla, hay que agrupar las que sean herramientas en el recuadro de la parte inferior izquierda de la pantalla, y los muebles, en la parte inferior derecha (figura 2-26). El elemento imagen de esta actividad se puede ver en la figura 2-25. Se puede observar que la

imagen silla2, que es un mueble, tiene como área correcta para colocarla $Ax1 = 50$, $Ax2 = 50$, $Ay1 = 50$, $Ay2 = 50$ que es precisamente la parte inferior derecha de la pantalla (recordar que $Ax1$ y $Ay1$ representan la posición inicial del área correcta en coordenadas relativas y $Ax2$ y $Ay2$ el ancho y el alto respectivamente que ocupa dicha zona también en coordenadas relativas). Por el contrario llave, que es una herramienta, tiene como área correcta la parte inferior izquierda de la pantalla representada por $Ax1 = 0$, $Ax2 = 50$, $Ay1 = 50$, $Ay2 = 50$.

Para esta plantilla el campo *Fijo* tiene que estar activado (a *true*) para los objetos que no se pueden mover por la pantalla.

La actividad resultante se ve en la figura 2-26.

imagen (12)					
Descripción	Posicion	Tamano	Ruta	Fijo	Fondo
1 herramientas	Posicion Inicial Final	Tamano Alto 50 Ancho 50	fondoherramientas.gif	true	true
2 casa	Posicion Inicial Final	Tamano Alto 50 Ancho 50	fondocasa.gif	true	true
3 textoherramientas	Posicion Inicial Final	Tamano Alto 10 Ancho 30	textoherramientas.gif	true	true
4 textomuebles	Posicion Inicial Final	Tamano Alto 10 Ancho 20	textomuebles.gif	true	true
5 fondo	Posicion Inicial Final	Tamano Alto 100 Ancho 100	fondoazul.gif	true	true
6 silla2	Posicion Inicial Final Ax1 50 Ax2 50 Ay1 50 Ay2 50	Tamano Alto 20 Ancho 10	silla2.gif	false	false
7 silla	Posicion Inicial Final	Tamano Alto 20 Ancho 10	silla.gif	false	false
8 sillon	Posicion Inicial Final	Tamano Alto 20 Ancho 20	sillon.gif	false	false
9 llave	Posicion Inicial Final Ax1 0 Ax2 50 Ay1 50 Ay2 50	Tamano Alto 20 Ancho 10	llave.gif	false	false
10 paleta	Posicion Inicial Final	Tamano Alto 15 Ancho 15	paleta.gif	false	false
11 clavos	Posicion Inicial Final	Tamano Alto 15 Ancho 10	clavos.gif	false	false
12 cincel	Posicion Inicial Final	Tamano Alto 15 Ancho 10	cincel.gif	false	false

Figura 2-25.- Datos del elemento imagen para una plantilla de tipo pinchar y mover.



Figura 2-26.- Actividad resultante de tipo pinchar y mover.

4.5.3 EJEMPLO DE ACTIVIDAD DE HACER PAREJAS (UNIR COLECCIONES)

Esta plantilla es como las de tipo *Pinchar*, pero se ha modificado para que se puedan emparejar objetos. Por ello hay que rellenar el campo llamado *Pareja* en el que hay que escribir el *Id* de la imagen con la que queremos emparejar la actual. Y puesto que cuando se relacionan dos imágenes de forma correcta aparecerá un recuadro alrededor de ellas se usa el campo *Color* para elegir el color de dichos recuadros. El ejemplo que vamos a ver es una actividad de correspondencia en la que hay que relacionar dos conjuntos de objetos con una relación de igualdad, concretamente guantes de colores. Se van a colocar las imágenes en dos columnas con la misma imagen repetida en cada una de ellas, pero en distintas posiciones. El alumno tiene que emparejar las de la columna izquierda con las de la derecha. La plantilla XML rellena se puede ver en la figura 2-27. Por ejemplo se puede observar que el guante31 tiene como pareja al guante32 y que cuando se emparejan correctamente se recuadran en naranja, el guante41 está emparejado con el guante42 y su recuadro es amarillo y así sucesivamente. En la figura 2-28 se muestra la actividad con algunas parejas resueltas correctamente.

imagen (9)							
Descripcion	Color	Posicion	Tamano	Ruta	Fijo	Fondo	Pareja
1 fondo	azul	Posicion Inicial lx 0 ly 0 Final	Tamano Alto 100 Ancho 100	fondo11.jpg	false	true	Ninguna
2 guante31	naranja	Posicion Inicial lx 25 ly 7 Final	Tamano Alto 17 Ancho 12	guante3.gif	false	false	guante32
3 guante41	amarillo	Posicion Inicial lx 25 ly 29 Final	Tamano Alto 17 Ancho 12	guante4.gif	false	false	guante42
4 guante51	rosa	Posicion Inicial lx 25 ly 51 Final	Tamano Alto 17 Ancho 12	guante5.gif	false	false	guante52
5 guante61	fucsia	Posicion Inicial lx 25 ly 73 Final	Tamano Alto 17 Ancho 12	guante6.gif	false	false	guante62
6 guante42	amarillo	Posicion Inicial lx 65 ly 7 Final	Tamano Alto 17 Ancho 12	guante4.gif	true	false	guante41
7 guante62	fucsia	Posicion Inicial lx 65 ly 29 Final	Tamano Alto 17 Ancho 12	guante6.gif	true	false	guante61
8 guante32	naranja	Posicion Inicial lx 65 ly 51 Final	Tamano Alto 17 Ancho 12	guante3.gif	true	false	guante31
9 guante52	rosa	Posicion Inicial lx 65 ly 73 Final	Tamano Alto 17 Ancho 12	guante5.gif	true	false	guante51

Figura 2-27.- Datos del elemento imagen para una plantilla de tipo hacer parejas.

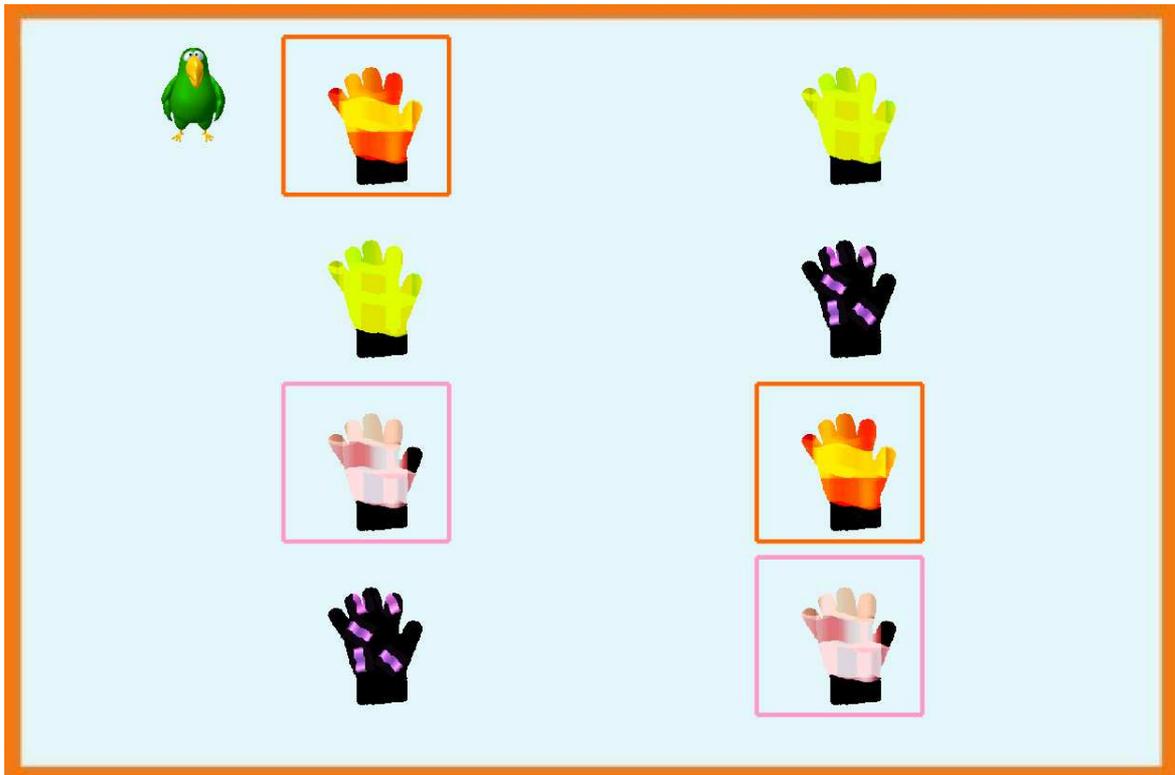


Figura 2-28.- Actividad resultante de tipo hacer parejas.

4.5.4 EJEMPLO DE ACTIVIDAD DE TIPO ORDEN

Con esta plantilla se pueden ordenar colecciones de imágenes. Es una modificación de la plantilla de *Pinchar y Mover* que tiene como objetivo, dada una serie de imágenes que se colocan en la parte superior de la pantalla, colocarlas en el orden que les corresponda en el rectángulo de la parte inferior (de menor a mayor, de mayor a menor, etc.). Además del fondo aparece otra imagen que consta de varias divisiones que nos permitirán ordenar las imágenes, teniendo ya calculado de antemano en qué área hay que situar esos objetos para que el resultado sea el esperado.

El campo *Orden* como ya se indicó anteriormente indica qué posición en el rectángulo tiene que ocupar el objeto. Todas las imágenes de fondo tienen que tener el valor 0. Se puede observar en la figura 2-29 la imagen división2 que es un rectángulo con 6 divisiones en la que el lapizt1 tiene que situarse en la posición 1, el lapizt2 en la 2 y así sucesivamente. Igual que en el caso de la plantilla de *Pinchar y Mover* el campo *Fijo* tiene que estar a *true* para los objetos que no se puedan

mover por la pantalla y *false* para los que sí. Así el lapizt2 y el lapizt6 no se pueden mover ya que aparecen directamente colocados en el recuadro correcto para ayudar al niño a solucionar el ejercicio. La actividad resultante se ve en la figura 2-30.

imagen (8)						
Descripción	Posicion	Tamaño	Ruta	Fijo	Fondo	Orden
1 fondo	▲ Posicion ▲ Inicial lx 0 ly 0 ▼ Final	▲ Tamaño Alto 100 Ancho 100	fondoliso3.jpg	true	true	0
2 division2	▲ Posicion ▲ Inicial lx 2 ly 50 ▼ Final	▲ Tamaño Alto 50 Ancho 97	division2.gif	true	true	0
3 lapizt1	▲ Posicion ▲ Inicial lx 40 ly 10 ▼ Final	▲ Tamaño Alto 15 Ancho 5	lapizt.gif	false	false	1
4 lapizt2	▲ Posicion ▲ Inicial lx 24 ly 75.5 ▼ Final	▲ Tamaño Alto 20 Ancho 5	lapizt.gif	true	false	2
5 lapizt3	▲ Posicion ▲ Inicial lx 71 ly 10 ▼ Final	▲ Tamaño Alto 25 Ancho 5	lapizt.gif	false	false	3
6 lapizt4	▲ Posicion ▲ Inicial lx 24 ly 10 ▼ Final	▲ Tamaño Alto 30 Ancho 5	lapizt.gif	false	false	4
7 lapizt5	▲ Posicion ▲ Inicial lx 56 ly 10 ▼ Final	▲ Tamaño Alto 35 Ancho 5	lapizt.gif	false	false	5
8 lapizt6	▲ Posicion ▲ Inicial lx 87 ly 55 ▼ Final	▲ Tamaño Alto 40 Ancho 5	lapizt.gif	true	false	6

Figura 2-29.- Datos del elemento imagen para una plantilla de tipo orden.

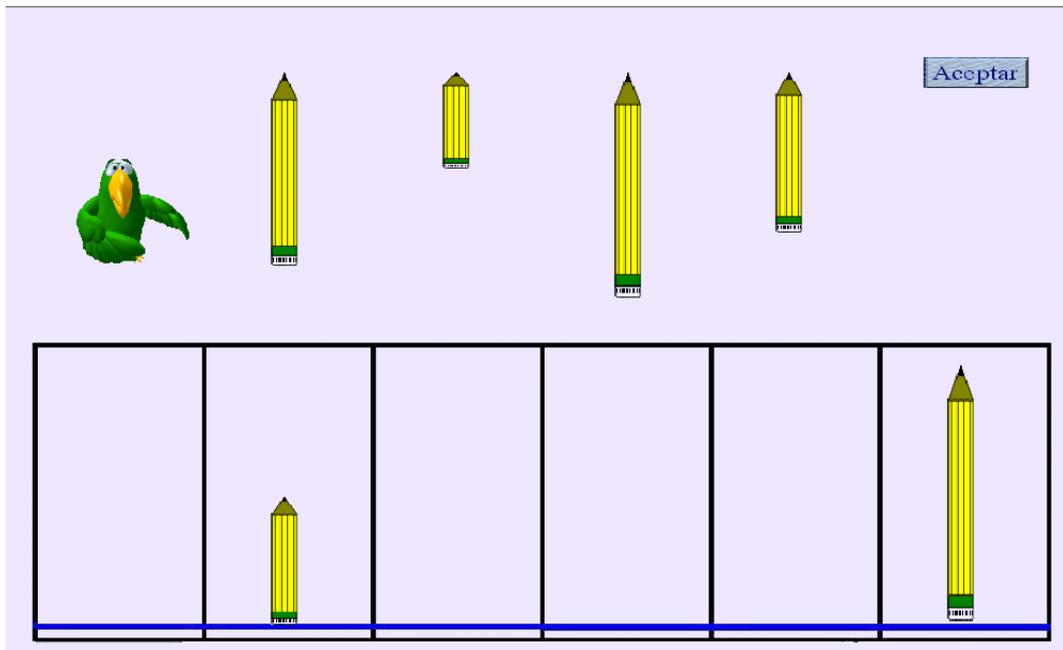


Figura 2-30.- Actividad resultante de tipo orden.

4.5.5 EJEMPLO DE TIPO DE ACTIVIDAD DE CANTIDAD

Esta plantilla también es una modificación de la de tipo Pinchar y Mover pero consiste en colocar una serie de imágenes encima de otra. Por ejemplo en la siguiente actividad se le pide al alumno que coloque siete peces en la pecera (figura 2-31). Por ello en la plantilla la imagen pecera, que es sobre la que el alumno va a tener que colocar los peces, se indica en el atributo Cantidad que el número para que la actividad sea correcta es 7. La actividad resultante se puede ver en la figura 2-32.

imagen (9)							
Descripción	Posicion	Tamano	Ruta	Fijo	Fondo	Cantidad	
1 fondo	▲ Posicion ▲ Inicial lx 0 ly 0 ▼ Final	▲ Tamano Alto 100 Ancho 100	fondo2.jpg	true	true	0	
2 pecera	▲ Posicion ▲ Inicial lx 20 ly 5 ▼ Final	▲ Tamano Alto 70 Ancho 50	pecera5.gif	true	true	7	
3 pez1	▲ Posicion ▲ Inicial lx 7 ly 70 ▼ Final	▲ Tamano Alto 10 Ancho 10	pez1.gif	false	false	0	
4 pez2	▲ Posicion ▲ Inicial lx 10 ly 87 ▼ Final	▲ Tamano Alto 10 Ancho 10	pez5.gif	false	false	0	
5 pez15	▲ Posicion ▲ Inicial lx 20 ly 80 ▼ Final	▲ Tamano Alto 10 Ancho 10	pez15.gif	false	false	0	
6 pez12	▲ Posicion ▲ Inicial lx 40 ly 80 ▼ Final	▲ Tamano Alto 10 Ancho 10	pez12.gif	false	false	0	
7 pez4	▲ Posicion ▲ Inicial lx 65 ly 85 ▼ Final	▲ Tamano Alto 10 Ancho 10	pez8.gif	false	false	0	
8 pez3	▲ Posicion ▲ Inicial lx 85 ly 85 ▼ Final	▲ Tamano Alto 10 Ancho 10	pez10.gif	false	false	0	
9 pez7	▲ Posicion ▲ Inicial lx 88 ly 70 ▼ Final	▲ Tamano Alto 10 Ancho 10	pez16.gif	false	false	0	

Figura 2-31.- Datos del elementos imagen para una plantilla de tipo cantidad.



Figura 2-32.- Actividad resultante de tipo cantidad.

5 GENERADOR DE ACTIVIDADES

Para facilitar la creación de plantillas por parte de los profesores y que no sea necesario que tengan que aprender XML se ha desarrollado un Generador de Plantillas. El objetivo del Generador es por tanto que los profesores puedan diseñar sus propios ejemplos.

El Generador ha sido diseñado utilizando tecnología Apache Cocoon que es un sistema de publicación web basado en XML/XSL. Está siendo desarrollado por el equipo Apache XML [Apco 2005] y cuenta con un desarrollo en Java que permite que se pueda ejecutar sobre cualquier servidor o localmente en la máquina del usuario. Cocoon es Open Source (Código abierto) y es un producto gratuito. Permite diferenciar el procesamiento del documento para tenerlo en diferentes formatos y cuenta con un sistema de caché que mejora el rendimiento.

Se han incluido varios tipos de interacciones en las plantillas, para favorecer la versatilidad de los tipos de actividades que se puedan crear. Estas son:

1. *Pinchar*
2. *Arrastrar Auto*
3. *Arrastrar*
4. *Parejas*
5. *Ordenar*

A continuación se explicará un ejemplo de cómo generar una actividad del tipo *Pinchar*. En el Apéndice C está incluido el manual de uso del Generador de Actividades donde se pueden encontrar ejemplos de todos los demás tipos implementados.

5.1 Pinchar

Son actividades en las que es necesario pinchar en uno, o más elementos, para que la actividad se considere correcta. La pantalla que se ve cuando se selecciona este tipo de actividad es la de la figura 2-33.

Plantillas de Pinchar

Imágenes:

Select	Id	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ancho	Alto	Ruta	Fijo	Fondo
<input type="button" value="Añadir Imagen"/>		<input type="button" value="Borrar imagen seleccionada"/>						

Agente:

Accion * ?

Select	Id	Posicion Eje X	Posicion Eje Y	Accion	Texto Explicacion
<input type="button" value="Añadir Explicación Agente"/>		<input type="button" value="Borrar Explicación seleccionada"/>			

Felicitation * ?

Correccion * ?

Siguiente Actividad:

Ruta * ?

Figura 2-33.- Actividad de Pinchar

Para insertar imágenes en nuestra actividad es necesario pinchar en el botón *Añadir Imagen* (figura 2-34).

Imágenes:

Select	Id	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ancho	Alto	Ruta	Fijo	Fondo
<input type="checkbox"/>	Anadir Imagen							
<input type="checkbox"/>	Borrar imagen seleccionada							

Figura 2-34. Acción de añadir imágenes.

Esto nos permite abrir un formulario para incorporar una imagen a nuestra actividad. Se pueden incorporar tantas imágenes como se deseen, sólo teniendo en cuenta que éstas van a ir en capas superpuestas desde el fondo de la pantalla. Esto significa que la primera imagen que deberíamos crear sería la del fondo, puesto que si lo hacemos al revés y por ejemplo ponemos en último lugar el fondo, taparía todas las demás imágenes. Los campos que tenemos que rellenar para crear una imagen son los que se han visto en la definición de plantillas de tipo Pinchar (epígrafe 4.5.1).

Para ver un ejemplo insertamos seis imágenes: un fondo, cuatro gatos y un perro, siendo la primera el fondo de la actividad y las siguientes posibles soluciones al ejercicio. En este ejemplo se va a determinar que la resolución correcta del ejercicio sea pinchar en la figura que representa al perro (figura 2-35).

Imágenes:

Select	Id	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ancho	Alto	Ruta	Fijo	Fondo
<input type="checkbox"/>	fondo	0	0	100	100	fondocielo1.gif	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	gato1	65	70	25	20	gato1.gif	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	gato2	40	40	25	20	gato2.gif	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	gato3	75	40	25	15	gato3.gif	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	gato4	25	70	25	15	gato4.gif	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	perro1	10	30	35	15	perro1.gif	<input type="checkbox"/>	<input type="checkbox"/>

Anadir Imagen Borrar imagen seleccionada

Figura 2-35. Ejemplo de cómo rellenar los campos Imagen para una actividad de tipo Pinchar

A continuación, hay que rellenar el campo agente, que se corresponde con el agente pedagógico que queremos que explique e interactúe con el niño en la

actividad. Para elegir el agente se debe pinchar en el menú desplegable que está debajo del texto. Una vez que hemos seleccionado alguno de estos agentes podemos definir las acciones que queremos que realicen en nuestra actividad. Igual que en el caso de las imágenes, podemos insertar las explicaciones de los agentes pinchando en el botón *Añadir Explicación Agente*. Los campos que aparecen para rellenar son comunes a todos los tipos de plantillas y fueron definidos en el epígrafe 3.5.1.1. En la figura 2-36 podemos ver un ejemplo de cómo se rellenan los campos para el agente.

Agente:

Accion * ?

Select	Id	Posicion Eje X	Posicion Eje Y	Accion	Texto Explicacion
<input type="checkbox"/>	<input type="text" value="accion1"/> * ?	<input type="text" value="10"/> * ?	<input type="text" value="20"/> * ?	<input type="text" value="Saludar"/> * ?	<input type="text" value="Hola. Yo soy Peedy"/> * ?
<input type="checkbox"/>	<input type="text" value="accion2"/> * ?	<input type="text" value="10"/> * ?	<input type="text" value="20"/> * ?	<input type="text" value="Explicar"/> * ?	<input type="text" value="Señala el perro"/> * ?

Felicitation * ?

Correccion * ?

Figura 2-36. Ejemplo de cómo rellenar el campo Agente para una actividad de tipo Pinchar

Por último tenemos la parte que hemos llamado *Siguiente Actividad*. Como muestra el nombre se trata de indicar la siguiente actividad que se va a mostrar al alumno cuando se resuelva la que se está creando, en el caso de que se quiera hacer una secuenciación de actividades. Se debe escribir la ruta de la siguiente actividad teniendo que incluirse la extensión *.html*. Por defecto aparece el texto *Ninguna* en el campo, lo que indica que la actividad se ejecuta de forma independiente (figura 2-37).

Siguiente Actividad:

Ruta * ?

Figura 2-37. Campo Siguiente Actividad relleno con el valor por defecto.

Una vez que hemos rellenado todos los campos, hay que pulsar en el botón *Enviar consulta*, para ver como ha quedado la actividad (figura 2-33). La actividad resultante para los valores que hemos definido se ve en la figura 2-38.

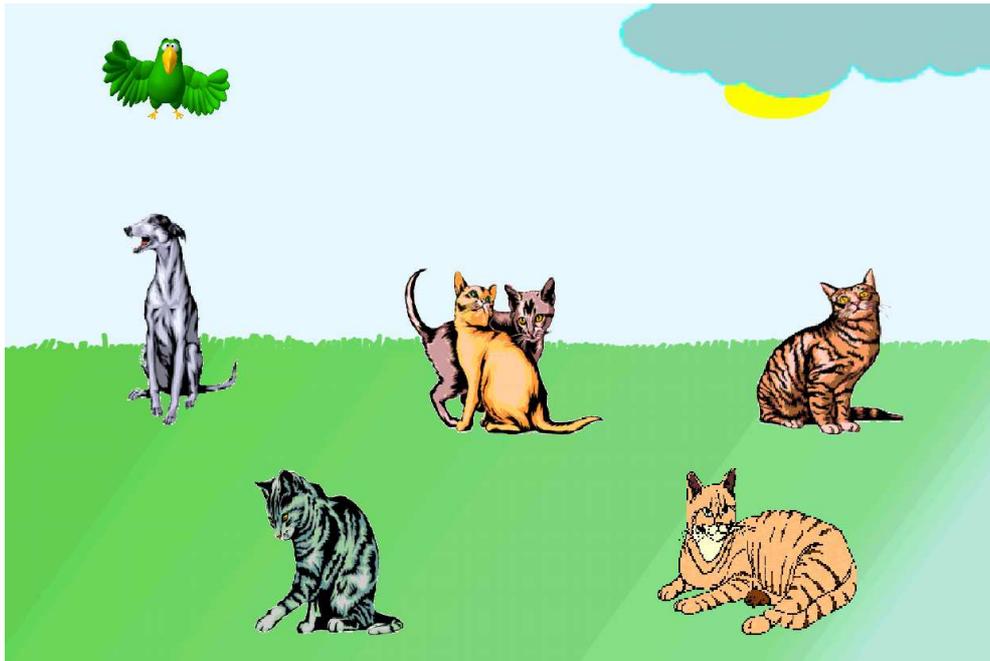


Figura 2-38. Ejemplo de actividad de Tipo Pinchar.

Si una vez que se ha visto el resultado de la actividad se quiere hacer algún cambio, se puede volver a la pantalla anterior pulsando en el botón *Atrás* de la barra de herramientas del navegador. Se puede modificar cualquiera de los campos, tanto de las imágenes como del agente. Si por el contrario en lugar de modificar, se quiere eliminar una imagen o una acción ya insertada habría que activar en el campo *Select* correspondiente a esa imagen o acción, y pinchar en el botón *Borrar Imagen Seleccionada* o *Borrar Explicación Seleccionada*. Por ejemplo, en la figura 2-39 se muestra el borrado de la imagen gato4.

Imágenes:

Select	Id	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ancho	Alto	Ruta	Fijo	Fondo
<input type="checkbox"/>	fondo	0	0	100	100	fondocielo1.gif	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	gato1	65	70	25	20	gato1.gif	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	gato2	40	40	25	20	gato2.gif	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	gato3	75	40	25	15	gato3.gif	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	gato4	25	70	25	15	gato4.gif	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	perro1	10	30	35	15	perro1.gif	<input type="checkbox"/>	<input type="checkbox"/>

Figura 2-39. Borrado de la imagen gato4.

Una vez que se crea la actividad, se puede guardar, haciendo uso del menú Archivo y eligiendo Guardar como. La opción correcta para el tipo de fichero es la que aparece por defecto: Página Web, sólo HTML (*.htm, *.html).

CAPÍTULO III

EL PLANIFICADOR INSTRUCCIONAL

1 INTRODUCCIÓN

El proceso de enseñanza individualizada consiste en determinar a partir de las características de cada alumno cuáles son los objetivos de aprendizaje. A continuación se deben elaborar una serie de actividades a realizar por el alumno que le permitan conocer los conceptos o adquirir las habilidades fijadas en los objetivos. El conjunto de actividades a realizar para alcanzar un mismo objetivo no será estándar para todos los alumnos, sino que dependerá de las características de cada uno de ellos. Esto es, a partir de un alumno concreto se fijan los objetivos de aprendizaje y se considera una secuencia de acciones para lograr esos objetivos. El componente del ITS que se encarga de realizar esta tarea es el planificador instruccional.

Los métodos utilizados para el diseño e implementación de planificadores instruccionales se basan en diferentes técnicas de IA que permiten representar el conocimiento experto del profesor en el proceso de enseñanza aprendizaje. Entre la bibliografía se pueden encontrar distintas técnicas: Razonamiento basado en casos, Procesos de decisión de Markov, Neuro-fuzzy, Redes bayesianas, Arquitectura de Pizarra, Reglas y Belief vectors. A continuación pasamos a describir cada uno de ellos.

1.1 Razonamiento basado en casos

Es un método para resolver problemas a partir de experiencias precedentes (casos), adaptando soluciones antiguas para resolver problemas nuevos o

recuperando casos anteriores para decidir aspectos de la situación actual. Generalmente en el razonamiento basado en casos se siguen los siguientes pasos:

1. Identificar el problema actual reconociendo las características más relevantes del nuevo problema.
2. Encontrar un caso similar al nuevo utilizando las características encontradas en paso anterior.
3. Utilizar el caso encontrado para sugerir una solución al problema actual (la solución propuesta se puede basar en uno o más casos similares al actual)
4. Evaluar la solución propuesta y determinar si de verdad es apropiada para el problema actual.
5. Actualizar el sistema almacenando el nuevo caso con su solución para permitir que el sistema incremente su conocimiento del problema.

Uno de los problemas del razonamiento basado en casos es que para poder iniciar el proceso de razonamiento, es necesario disponer de un conjunto de casos precedentes o librería de casos. Un inconveniente de esta técnica es que al elegir un caso almacenado como solución al problema podría no recuperarse el caso más apropiado.

El razonamiento basado en casos aplicado a la planificación se conoce como planificación basada en casos (case-based planning). La planificación basada en casos se caracteriza por construir los planes conforme a los conocimientos que tiene el planificador sobre los planes empleados en experiencias pasadas (casos). La construcción de un plan consiste en la búsqueda de un plan que, estando almacenado en la memoria del planificador, satisfaga todos los objetivos. Puesto que un profesor experimentado puede hacer un plan educacional para un estudiante en particular basándose en el carácter del estudiante y en los resultados obtenidos (y almacenar esos datos en la biblioteca de casos), esta técnica se basa en el hecho de que se pueden reutilizar (y en ocasiones combinar) esos casos pasados para resolver nuevos problemas. [Warendorf 1997, Elorriaga 1999, Remolina 2002].

1.2 Procesos de decisión de Markov

Los problemas de decisión involucran un conjunto de decisiones cuyo resultado no se conoce hasta el final. Se considera que se tiene una serie de estados y ciertas acciones que pueden llevar a modificar el estado. Los resultados de las acciones no son exactos sino que presentan cierta incertidumbre que se traduce en términos de probabilidad.

Inicialmente para realizar el planificador el profesor tiene que asumir que el estudiante tiene cierta cantidad de conocimiento y que la meta de la planificación instruccional es llegar a un estado determinado (o varios). La planificación instruccional por tanto se considera como la repetición del proceso de toma de decisiones, donde en cada estado el profesor debe determinar una acción para provocar una transición del estado actual a un estado mejor.

En los planificadores de este tipo, un estado representa la creencia del profesor particular sobre una situación. Dicha creencia se puede obtener combinando el conocimiento del estudiante, una respuesta prevista, y cualquier otra información derivada de las acciones. Una acción es una interacción que puede ser realizada por el estudiante. Realizar acciones cambia el estado pero, puesto que la respuesta del estudiante a una acción no es determinista, una acción dada con diferentes probabilidades puede conducir a diferentes estados. Por último, un evento es una acción inesperada del estudiante (como por ejemplo hacer una pregunta) lo cuál puede también causar una transición de estado.

Las desventajas que presentan los Modelos de Markov son la difícil modelización de sistemas complejos ya que no se ve el estado en el que se encuentra el sistema y el hecho de que no tienen memoria (el tiempo en un estado no influye sobre la distribución de probabilidad, ni del estado actual ni del próximo estado) [Matsuda 2000].

1.3 Neuro-fuzzy

Los sistemas Neuro-fuzzy son sistemas basado en técnicas borrosas y en redes neuronales. La lógica borrosa trabaja con conjuntos que no tienen sus límites

definidos de forma que la transición entre la pertenencia y no pertenencia a un conjunto es gradual; crea aproximaciones matemáticas produciendo resultados a partir de datos imprecisos y cuantifica expresiones humanas vagas tales como un poco caliente, muy frío, etc. Por otro lado las redes neuronales intentan imitar el funcionamiento del cerebro humano. Una red neuronal artificial es un sistema compuesto por muchos elementos procesadores simples conectados en paralelo, cuya función es determinada por la estructura de la red, la fuerza de las conexiones y el procesamiento realizado por los elementos en las neuronas [Jang 1997].

Existen dos tipos de aproximaciones principales en los planificadores basados en técnicas de Neuro-fuzzy: Utilizar algoritmos de aprendizaje derivados de las redes neuronales para ajustar los parámetros de un sistema de reglas borrosas (o generar nuevas reglas), o bien borrosificar redes neuronales o utilizar reglas borrosas para optimizar el procedimiento de una red neuronal [Magoulas 2001].

Una de las desventajas de las Redes Neuronales es que requieren la definición de muchos parámetros antes de poder aplicar la metodología. Por ejemplo hay que decidir la arquitectura más apropiada, el número de capas ocultas, el número de neuronas por capa, las interconexiones, la función de activación, etc. Otra desventaja de las Redes Neuronales es que no ofrecen una interpretación fácil.

1.4 Redes bayesianas

Las redes bayesianas o probabilísticas están diseñadas para hallar las relaciones de dependencia e independencia entre todas las variables que conforman el dominio de estudio. Esto permite realizar predicciones sobre el comportamiento de cualquiera de las variables desconocidas a partir de los valores de las otras variables conocidas. Esto presupone que cualquier variable de la base de datos puede comportarse como incógnita o como evidencia según el caso.

Uno de los inconvenientes de las redes bayesianas es que normalmente manejan variables multivaluadas discretas. Cuando se presentan variables continuas (temperatura, estatura, etc.), éstas se discretizan en un número de intervalos y se manejan como si fueran discretas y como consecuencia de esto, si el número de intervalos es pequeño, se pierde precisión y si el número de intervalos

es grande, el modelo se vuelve demasiado complejo y se requiere gran cantidad de datos para estimar las probabilidades. [Cook 1998]

1.5 Arquitectura de pizarra

Existen varios sistemas tutoriales inteligentes que usan una planificación basada en arquitecturas de pizarra y que incluyen sistemas multiagentes [Maes 1997]. La metáfora de la resolución de problemas en pizarra, es muy simple: se incluyen un conjunto de agentes reunidos alrededor de la pizarra, mirando las piezas de información que se exponen, pensando y escribiendo las conclusiones, con una serie de suposiciones:

- Los agentes pueden ver en todo momento la pizarra. Lo que ven representa el estado actual de la solución.
- En cualquier momento un agente puede escribir sus conclusiones en la pizarra.
- El hecho que un agente escriba en la pizarra no debe confundir a los otros agentes.

La implicación de estas suposiciones es que un problema se está resolviendo de forma asíncrona y en paralelo.

La principal desventaja de este sistema es que puesto que los datos de la pizarra tienen que ser consistentes el método se puede convertir en un cuello de botella a la hora de los accesos de los agentes a la pizarra [Dimitrova 1998].

1.6 Reglas

Un planificador basado en reglas consta de un conjunto de reglas mediante las cuales el planificador debe ser capaz de seleccionar una reacción (la activación de un consecuente mediante el cumplimiento del antecedente de la regla) a cualquier situación que ocurra durante la ejecución.

Las desventajas de este tipo de planificadores instruccionales es que el proceso de adquisición de conocimiento del experto puede implicar un largo periodo de tiempo, puede tener errores y por lo general es un proceso costoso. Es muy difícil realizar un programa de aplicación para la enseñanza que contenga *todas* las reglas imaginables (algunas pueden faltar, otras estar incompletas, etc.) [Vassileva 1996].

1.7 Belief vectors

Los belief vectors se basan en redes de inferencia o probabilidad y fueron usados por primera vez por Katz, Lesgold, Eggan y Gordin en 1993. Cada vector contiene varios puntos (por ejemplo (0.10 0.23 0.3 0.19 0) y la suma de los cuales puede ser como máximo 1 (en este ejemplo el valor de la suma es 0.82). En un planificador instruccional el vector contiene tantos puntos como niveles de dificultad existan. El valor en cada punto indica aproximadamente la probabilidad de que el estudiante esté en ese nivel de dificultad. Así el vector que usamos como ejemplo anteriormente indica cinco niveles de dificultad y el planificador indica que hay una probabilidad del 10% de que el alumno tenga que estar en el nivel 1, el 23% de que tenga que estar en el nivel 2, un 30% para el nivel 3, un 19% para el nivel 4 y no hay ninguna probabilidad de que tenga que estar en el nivel 5. Los alumnos suben o bajan de nivel en función de que respondan correcta o incorrectamente a las cuestiones que se les planteen [Santally 2005].

2 LÓGICA BORROSA Y SISTEMAS MULTIAGENTE

La mayoría de los ITS existentes utilizan como se ha mencionado anteriormente un planificador instruccional que decide los objetivos de enseñanza. En esta Tesis Doctoral nos hemos decantado por desarrollar un planificador instruccional que combine las técnicas de lógica borrosa y tecnología multiagente aprovechando las ventajas que nos proporcionan cada una de ellas. Proponemos la lógica borrosa porque el manejo de la imprecisión es una herramienta útil en el ITS y permite que el conocimiento fluya libremente del experto. Hay diferentes formas

de modelar la imprecisión, pero en la mayoría de los casos esa imprecisión es utilizada sólo en un pequeño grado. La mayoría de los métodos para el manejo de imprecisión son métodos probabilísticos, pero es interesante tener en cuenta que los expertos no piensan normalmente en términos de probabilidad sino en términos tales como mucho, poco, bien, regular, etc. Un sistema experto borroso significa que el sistema incorpora conjuntos borrosos y/o lógica borrosa en el proceso de razonamiento y/o en la representación del conocimiento. Las teorías de conjuntos y lógica borrosa están bien documentadas. Dichas teorías existen desde hace 25 años y han sido utilizadas en múltiples aplicaciones de control. Se han desarrollado también varios sistemas expertos que incorporan técnicas borrosas [Ammar 2004]. En nuestro ITS la metodología de lógica borrosa es empleada para modelar la incertidumbre de la base de conocimiento del estudiante y la estrategia de enseñanza. Por otra parte la utilización de un sistema multiagente es una gran ventaja para aplicaciones que puedan utilizar computación distribuida ya que, la posibilidad de dividir las tareas proporciona modularidad, flexibilidad y disminución del tiempo de cómputo. Al incorporar esta técnica a nuestro ITS conseguimos mejorar la estrategia de enseñanza de los objetivos y a la vez disminuir el tiempo de espera en la interacción con el usuario.

A continuación veremos un pequeño resumen de cada una de dichas teorías para posteriormente centrarnos en el diseño e implementación del planificador instruccional.

3 TEORÍA BORROSA

3.1 Conjuntos Borrosos

El concepto de conjunto borrosos fue introducido por primera vez por Lofti A. Zadeh, profesor de la Universidad de California Berkeley, en el año 1964, en un intento de representar y manipular datos que no eran precisos. Dicho concepto dio paso a la denominada *Teoría de los Subconjuntos Borrosos*, ya que ciertas magnitudes pueden tomar valores que difícilmente se pueden clasificar en un conjunto determinado. La Teoría de Subconjuntos Borrosos permite la definición adecuada de conjuntos que modelan las situaciones de imprecisión antes expuestas.

En la teoría clásica de conjuntos, un subconjunto A de un conjunto X puede ser definido por una función característica χ_A que puede tomar dos valores: 0 y 1

$$\chi_A : X \rightarrow \{0,1\}$$

La verdad o falsedad de la expresión "x está en A" viene dada por el par ordenado $(x, \chi_A(x))$, de forma que será cierta si el segundo elemento del par es 1 y falsa cuando sea 0.

De forma similar se puede definir un subconjunto borroso A de un conjunto X, denominado *universo de discurso*, como el conjunto de pares ordenados $(x, \mu_A(x))$, siendo μ_A la *función de pertenencia* al conjunto borroso A, definida como

$$\mu_A : X \rightarrow [0,1]$$

Dicha función puede tomar todos los valores del intervalo entre 0 y 1. El valor 0 representa la no-pertenencia al conjunto A y el valor 1 representa la pertenencia total a dicho conjunto.

3.2 Números Borrosos

Un número borroso es un caso particular de conjunto borroso cuya función de pertenencia es continua, convexa y definida sobre un intervalo cerrado de los números reales. Dicha función de pertenencia puede adoptar diferentes formas siendo las más habituales las triangulares o trapezoidales que se caracterizan por un número reducido de parámetros, aunque también son habituales la gaussiana o la campana generalizada. En la Figura 3-1 se pueden observar ejemplos de las diferentes funciones de pertenencia comentadas.

3.3 Variables lingüísticas

Los conjuntos borrosos se pueden utilizar para representar variables lingüísticas cuyos valores son números borrosos que están definidos en términos lingüísticos. Al conjunto de estos números borrosos, que abarcan todo el universo de discurso de la variable, se le denomina *partición borrosa*.

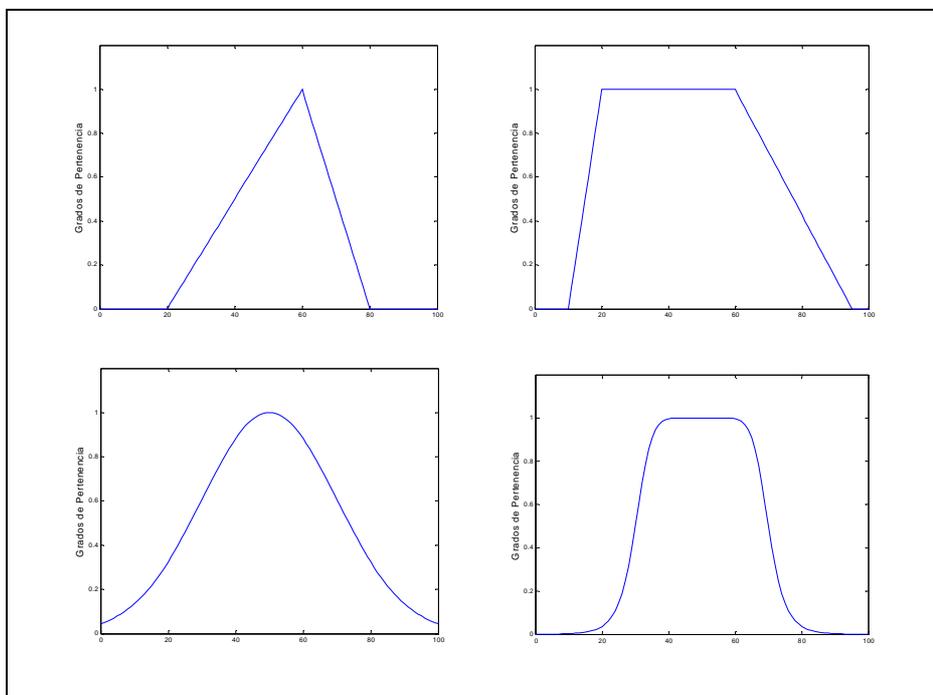


Figura 3-1.- Ejemplos de funciones de pertenencia triangular, trapezoidal, gaussiana y campana generalizada.

El número de conjuntos borrosos que componen dicha partición se suele tomar según el grado de precisión requerido para esa variable. Tomar una gran cantidad de conjuntos borrosos (siete o más de siete, en general), tiene como ventaja el poder precisar las acciones que se van a llevar a cabo en el sistema en función de esta variable. La desventaja es que la *base de conocimiento* del sistema debe contemplar todos los términos lingüísticos (conjuntos), por lo que el tamaño de la misma crecerá enormemente.

Por ejemplo, si la temperatura se interpreta como una variable lingüística, el conjunto de valores que puede tomar podría ser {muy fría, fría, media, templada, cálida, calurosa}. Cada uno de estos términos está caracterizado por un conjunto borroso definido en el universo de discurso ($[-6^{\circ}\text{C}, 48^{\circ}\text{C}]$) de la variable temperatura, tal y como muestra la figura 3-2.

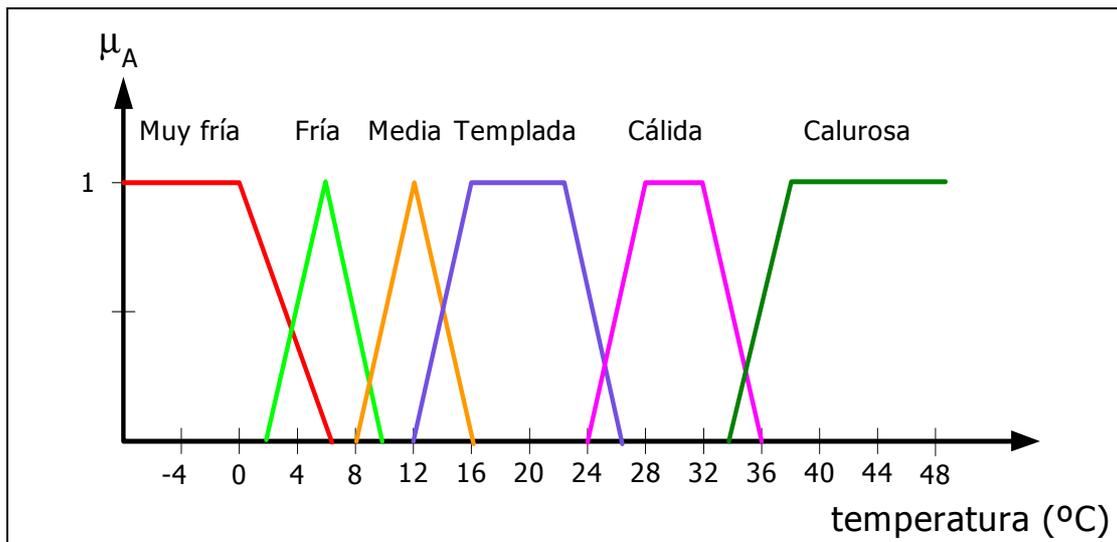


Figura 3-2.- Definición y partición borrosa de la variable lingüística temperatura.

3.4 Operaciones con Conjuntos Borrosos

Las operaciones que se definen en la lógica borrosa pueden reducirse a las definiciones clásicas si el grado de pertenencia a los conjuntos borrosos se limita al conjunto $\{0,1\}$. Por esta razón, se utiliza la misma notación que para los conjuntos concisos.

Así se puede definir el complemento (Tabla 3-1), la T-norma (o intersección) (Tabla 3-2) y la S-norma (o unión) (Tabla 3-3).

Estándar	$c(x) = 1 - x$
Sugeno	$c_{\lambda}(x) = \frac{1-x}{1+\lambda x}$, con $\lambda > -1$
Yager	$c(x) = (1-x^p)^{1/p}$, $p > 0$

Tabla 3-1.- Definición de complementos básicos.

Mínimo	$T(x,y) = \min\{x,y\}$
Lukasiewicz (producto acotado)	$T(x,y) = \max\{x+y-1,0\}$
Producto algebraico	$T(x,y) = xy$
Producto drástico	$T_w(x,y) = \begin{cases} \min\{x,y\} & \text{si } \max\{x,y\} = 1 \\ 0 & \text{en otro caso} \end{cases}$
Hamacher	$T(x,y) = \frac{xy}{\gamma + (1-\gamma)(x+y-xy)}$, $\gamma \geq 0$
Frank	$T(x,y) = \log_s \left[1 + \frac{(s^x - 1)(s^y - 1)}{s - 1} \right]$, $s \geq 0$
Dubois and Prade	$T(x,y) = \frac{xy}{\max\{x,y,\alpha\}}$, $\alpha \in [0,1]$
Yager	$T(x,y) = 1 - \min\left\{1, \left[(1-x)^p + (1-y)^p \right]^{1/p} \right\}$, $p > 0$

Tabla 3-2.- Definición de T-normas básicas

Máximo	$S(x,y) = \max\{x,y\}$
Lukasiewicz (suma acotada)	$S(x,y) = \min\{x+y,1\}$
Probabilística (suma algebraica)	$S(x,y) = x+y-xy$
Suma drástica	$S_w(x,y) = \begin{cases} \max\{x,y\} & \text{si } \min\{x,y\} = 1 \\ 0 & \text{en otro caso} \end{cases}$
Hamacher	$S(x,y) = \frac{x+y-(2-\gamma)xy}{1-(1-\gamma)xy}, \gamma \geq 0$
Frank	$S(x,y) = 1 - \log_s \left[1 + \frac{(s^{1-x}-1)(s^{1-y}-1)}{s-1} \right], s \geq 0$
Yager	$S(x,y) = \min \left\{ 1, \sqrt[p]{x^p + y^p} \right\} \quad p > 0$

Tabla 3-3.- Definición de S-normas básicas

3.5 Reglas borrosas si-entonces

Una regla borrosa del tipo "si-entonces", tiene la siguiente forma:

Si x es A entonces y es B

siendo A y B valoraciones particulares de sendas variables lingüísticas. El término "x es A" se denomina antecedente, mientras que el término "y es B" se denomina consecuente. Ejemplos sencillos de tales reglas serían:

- "Si la presión es alta, entonces el volumen es pequeño"
- "Si la temperatura es alta, entonces hace calor"

Para poder utilizar una regla borrosa es necesario formalizar el significado de la expresión "Si x es A entonces y es B", que abreviadamente se indica como $A \rightarrow B$. Para ello es útil definir esta relación borrosa binaria.

Una relación borrosa binaria es un conjunto borroso R, sobre $X \times Y$ (X e Y son dos universos de discurso), que asignan a cada elemento de $X \times Y$ un número entre 0 y 1. Esto es:

$$R = \{((x,y), \mu_R(x,y)) \mid (x,y) \in X \times Y\}$$

La relación borrosa asociada a la regla borrosa se puede obtener usando alguna de las T-normas citadas anteriormente, por ejemplo, si se usa el operador mínimo para la conjunción lógica (propuesto por Mamdani) tendremos:

$$R_m = (A \rightarrow B) = \min\{\mu_A(x), \mu_B(y)\} / (x,y)$$

3.6 El razonamiento borroso

El razonamiento borroso (o aproximado) permite obtener conclusiones a partir de un conjunto de reglas borrosas y un conjunto de hechos borrosos conocidos.

La regla de inferencia clásica es el Modus Ponens que nos permite inferir la verdad de la proposición B ("hace calor"), a partir de la verdad de la proposición A ("la temperatura es alta") y de la implicación $A \rightarrow B$ ("si la temperatura es alta, entonces hace calor"), de forma esquematizada tendríamos:

Premisa 1 (Regla):	Si x es A	ENTONCES	y es B
Premisa 2 (Hecho):	x es A		
Consecuente (Conclusión):			y es B

Sin embargo, en muchos razonamientos humanos, el modus ponens es utilizado de una forma aproximada. Por ejemplo, si tenemos la misma regla de

implicación $A \rightarrow B$ ("si la temperatura es alta, entonces hace calor"), y sabemos que "la temperatura es más o menos alta" (x es A'), entonces podemos inferir que "hace más o menos calor" (y es B'), que se podría esquematizar de la siguiente manera:

Premisa 1 (Regla):	Si x es A	ENTONCES	y es B
Premisa 2 (Hecho):	x es A'		
Consecuente (Conclusión):			y es B'

donde A' es "más o menos" A , y B' es "más o menos" B . Este tipo de razonamiento es el razonamiento borroso, siendo A , B , A' y B' conjuntos borrosos del universo del discurso correspondientes. A esta regla de inferencia también se le conoce como Modus Ponens Generalizado (GMP) ya que la regla de Modus Ponens es un caso especial de ésta.

Por lo tanto, el razonamiento borroso nos permite obtener conclusiones a partir de reglas borrosas de tipo si-entonces y de hechos conocidos. La base de este procedimiento es la regla composicional de inferencia. Ésta es una generalización del concepto de curva. Una curva viene dada por una función que permite obtener el valor de y a partir del valor de x , mediante $y=f(x)$, donde el papel de la variable independiente lo jugará el hecho (x es A'), el papel de la función lo tomará la relación borrosa asociada a la regla ($A \rightarrow B$) y finalmente, el papel de la variable dependiente será para una conclusión borrosa (y es B'), figura 3-5.

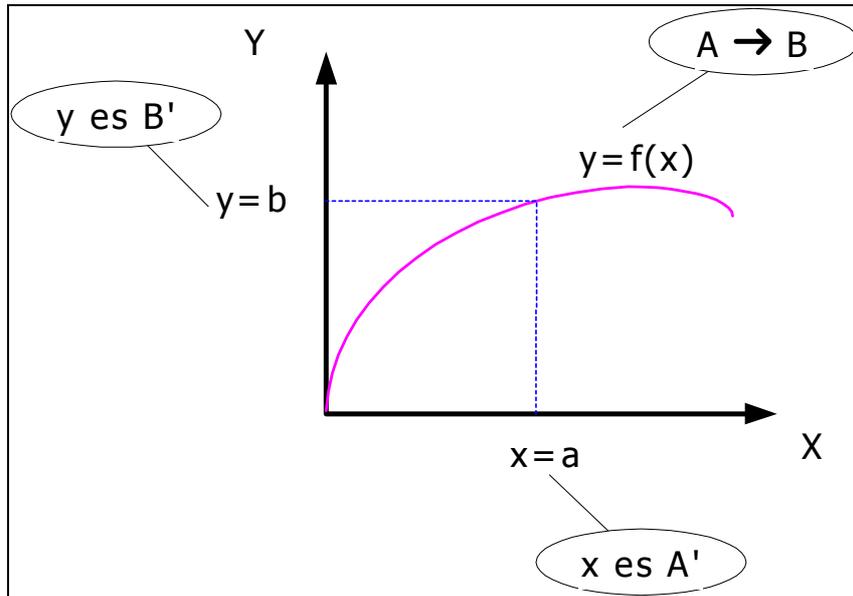


Figura 3-5.- Regla compocisional de inferencia como generalización del concepto de curva.

Entonces, sea R una relación borrosa sobre $X \times Y$, y sea A un conjunto borroso sobre X . Para encontrar el conjunto borroso resultante B , construiremos una extensión cilíndrica de A , $c(A)$, que tome como base A . La extensión cilíndrica se define como $\mu_{c(A)}(x,y) = \mu_A(x)$, y por lo tanto es un conjunto borroso sobre $X \times Y$ que nos permitirá calcular la intersección con el conjunto borroso de la relación borrosa R . Finalmente proyectamos esta intersección $(c(A) \cap R)$ sobre el dominio Y , obteniendo así un conjunto borroso B que representa la conclusión. Veamos gráficamente cada uno de estos pasos.

Partimos de la premisa 1 que es la regla $R = A \rightarrow B$ (en el concepto de curva sería la función $y=f(x)$), que nos da la relación borrosa R sobre $X \times Y$ que vemos en la figura 3-6.

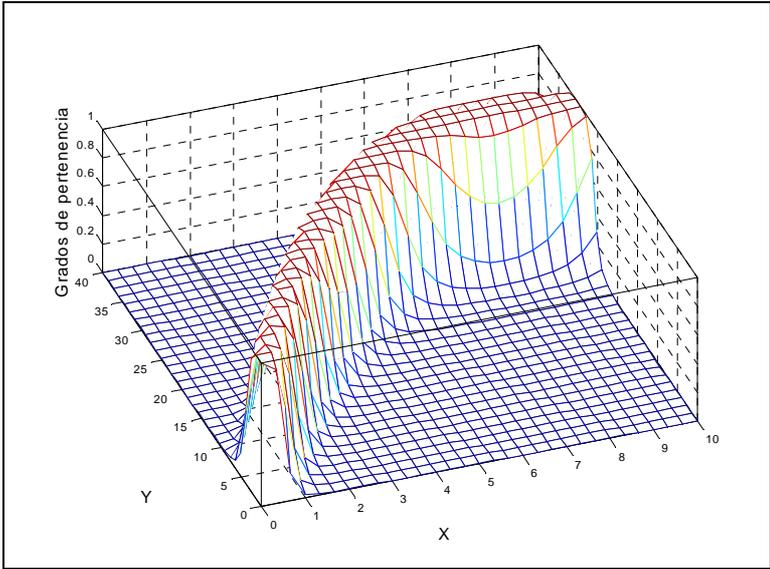


Figura 3-6.- Relación borrosa R sobre X y Y.

La segunda premisa consiste en el hecho x es A' (en el concepto de curva el valor $x=a$) para extender este hecho hasta que corte la relación borrosa tendremos que realizar la extensión cilíndrica de A', figura 3-7.

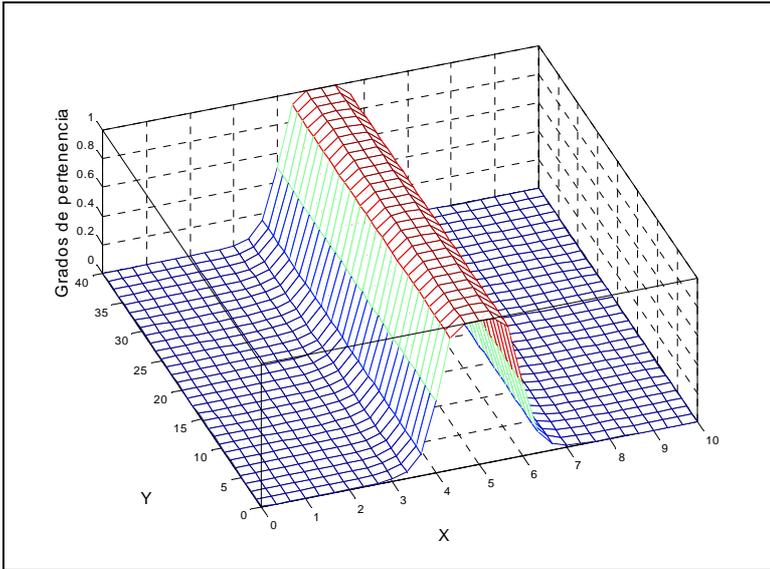


Figura 3-7.- Extensión cilíndrica de A', $c(A')$.

La intersección entre la extensión cilíndrica de A' $c(A')$ y la relación borrosa R , figura 3-8, forma una región que es análoga al punto de corte $x=a$ con la función $y=f(x)$, en el concepto de curva.

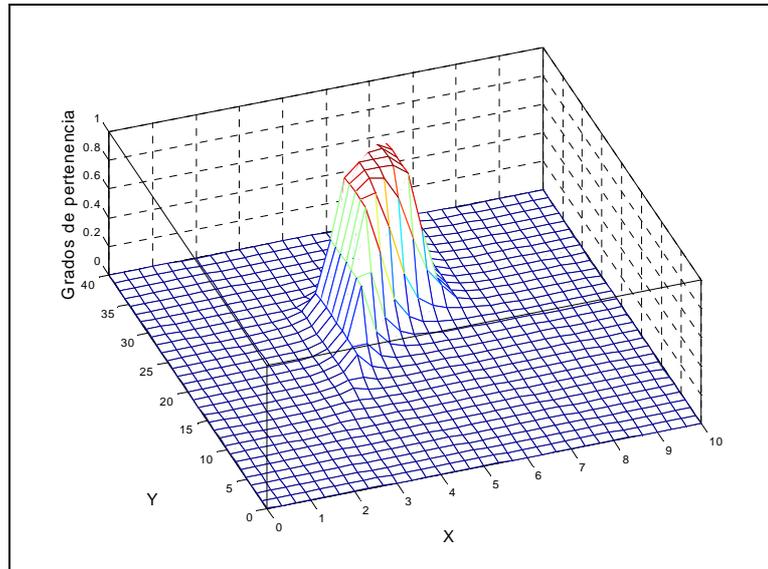


Figura 3-8.- Intersección (mínimo) de la relación borrosa R y la Extensión cilíndrica de A' .

Proyectando la intersección de $R \cap c(A')$ sobre el eje y , podremos inferir y' como el conjunto borroso B sobre el eje Y , esto es la conclusión y' es B , que se muestra en la figura 3-9.

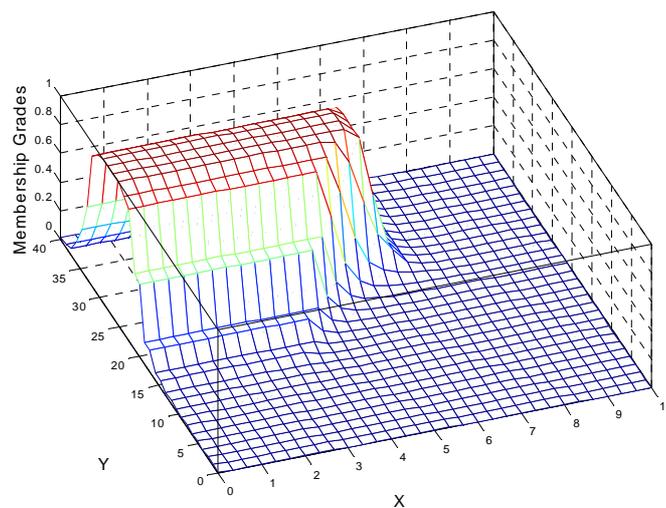


Figura 3-9.- Proyección (máximo) sobre el eje Y de la intersección entre la relación borrosa R y la extensión cilíndrica de A' para obtener la conclusión y' es B .

Podemos concluir que el conjunto borroso B' inducido por las premisas "x es A'" y la regla borrosa "si x es A entonces y es B" se define por:

$$B' = A' \circ R = A' \circ (A \rightarrow B) \quad \text{donde } \circ \text{ indica el operador composicional}$$

Si al aplicar la regla composicional de inferencia utilizamos el mínimo para la intersección de $R \cap (A')$ obtenemos:

$$\mu_{B'}(y) = \max_x \min[\mu_{A'}(x), \mu_R(x,y)] \quad \text{composición max-min}$$

Si la intersección de $R \cap (A')$ se realiza con la operación producto tendríamos:

$$\mu_{B'}(y) = \max_x [\mu_{A'}(x) \mu_R(x,y)] \quad \text{composición max-producto}$$

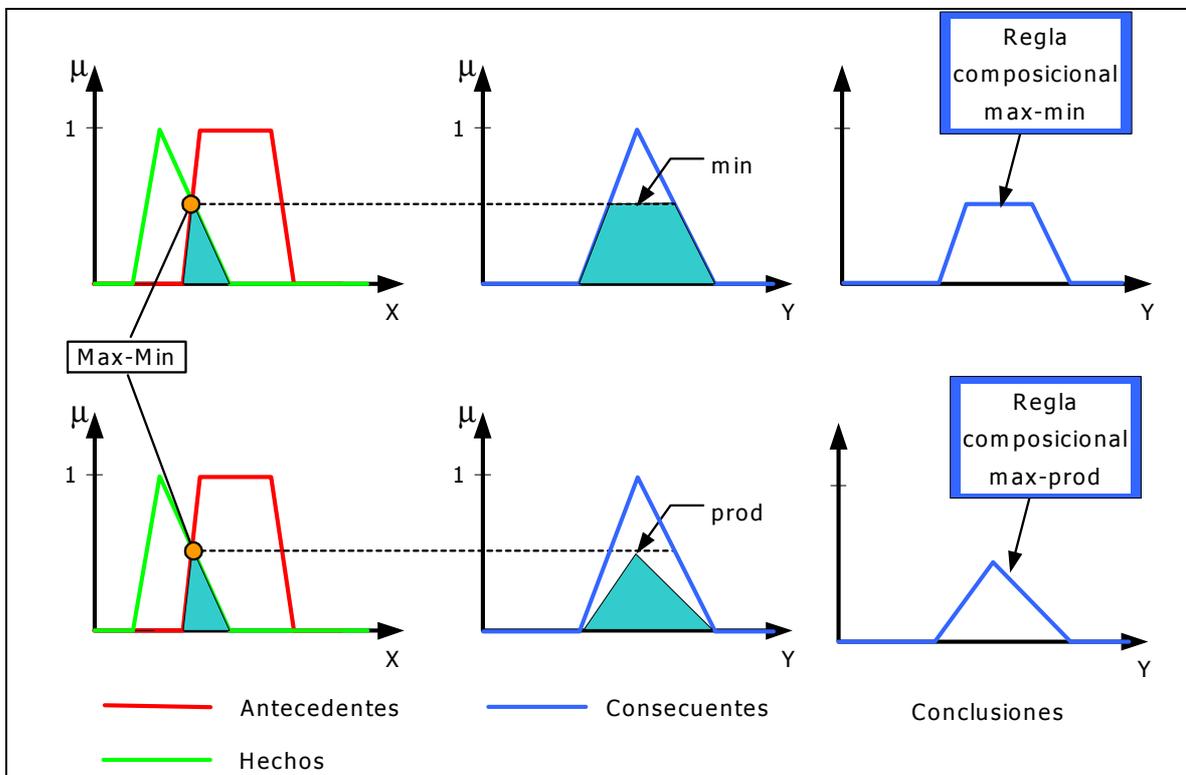


Figura 3-10.- Razonamiento borroso utilizando regla composicional max-min en las tres gráficas superiores y utilizando max-prod en las tres gráficas inferiores.

En la figura 3-10 se representa gráficamente el mecanismo de razonamiento aproximado para una regla con un antecedente. Las tres gráficas superiores se refieren a la regla composicional del tipo max-min. La primera gráfica (de izquierda a derecha) muestra como se componen el antecedente y el hecho. En este caso el hecho tiene asociada una función de pertenencia triangular, y el antecedente una función de pertenencia trapezoidal. La composición de ambas funciones de pertenencia mediante la operación max-min es lo que denominamos grado de compatibilidad. En este caso, como sólo hay un antecedente el grado de compatibilidad es igual a la fuerza de disparo de la regla. La siguiente gráfica se refiere al consecuente de la regla. La fuerza de disparo y la función de pertenencia asociada al consecuente son utilizadas para obtener la función de pertenencia del consecuente cualificado (resultado de la inferencia de la regla). En las tres gráficas superiores se utiliza el operador mínimo mientras que en las tres gráficas inferiores se utiliza el operador producto. Dependiendo del operador utilizado se obtiene un conjunto borroso resultado de la inferencia diferente.

3.7 Una regla con un antecedente

Este es el caso más sencillo, considerando una regla composicional de inferencia del tipo max-min, y una la implicación borrosa de Mamdani, tendremos:

$$\mu_{B'}(y) = \max_x \min[\mu_{A'}(x), \mu_R(x,y)] = \max_x \min[\mu_{A'}(x), \min[\mu_A(x), \mu_B(y)]]$$

Simplificando tendríamos:

$$\mu_{B'}(y) = \min\{\max_x \min[\mu_{A'}(x), \mu_A(x)], \mu_B(y)\} = \min[w, \mu_B(y)]$$

siendo $w = \max_x \min[\mu_{A'}(x), \mu_A(x)]$, que se denomina grado de compatibilidad entre el conjunto borroso del hecho (A') y el antecedente (A).

En otras palabras, primero encontramos el grado de compatibilidad w como el máximo de la región que resulta de la operación mínimo entre $\mu_{A'}(x)$ y $\mu_A(x)$ (zona sombreada en el antecedente, figura 3-11). La función de pertenencia del conjunto borroso B' será igual a la función de pertenencia de B recortada por w (zona sombreada en el consecuente, figura 3-11).

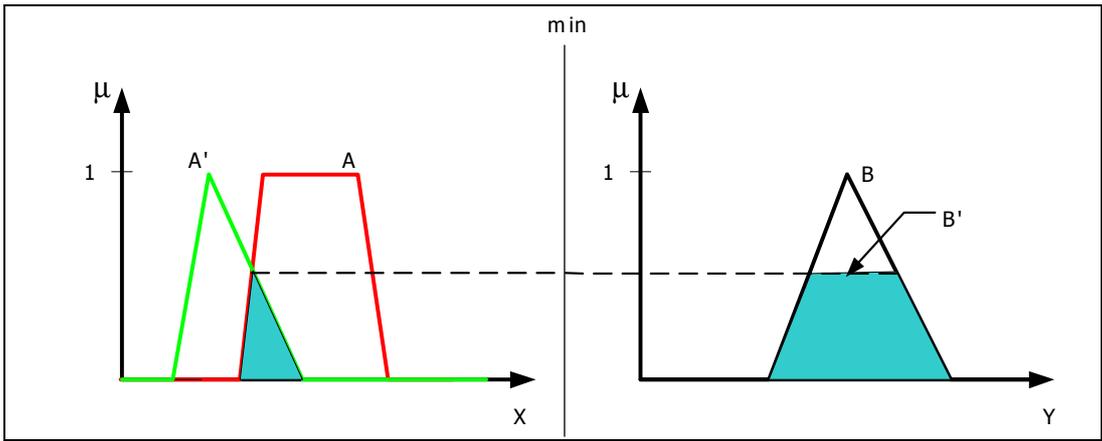


Figura 3-11.-Interpretación gráfica del Modus Ponens Generalizado, utilizando la implicación borrosa de Mamdani (mínimo) y el operador composicional max-min.

3.8 Una regla con múltiples antecedentes

Una regla borrosa del tipo si-entonces con dos antecedentes es de la forma: "Si x es A Y y es B ENTONCES z es C ", la Generalización del Modus Ponens para este problema se esquematizaría de la siguiente manera:

Premisa 1 (Regla):	SI x es A Y y es B	ENTONCES	z es C
Premisa 2 (Hecho):	x es A' Y y es B'		
Consecuente (Conclusión):			z es C'

El método para obtener C' se basa en utilizar una relación ternaria borrosa para describir esta regla con dos antecedentes, esto es, $A \times B \rightarrow C$ puede ser transformada, utilizando por ejemplo la implicación borrosa de Mamdani, como:

$$R_m = (A \times B \rightarrow C) = (A \times B) \times C = \min\{\mu_A(x), \mu_B(y), \mu_C(z)\} / (x,y,z)$$

El resultado C' es el siguiente:

$$C' = (A' \times B') \circ (A \times B \rightarrow C) = [A' \circ (A \rightarrow C)] \cap [B' \circ (B \rightarrow C)]$$

Esto es, el resultado del consecuente C' puede ser expresado como la intersección de $C_1'=[A' \circ (A \rightarrow C)]$ y $C_2'=[B' \circ (B \rightarrow C)]$, cada uno de estos términos corresponde a la inferencia borrosa para el caso más simple con un solo antecedente.

Utilizando el operador composiciona max-min tendremos:

$$\mu_{C'}(z) = \min\{\max_x \min[\mu_{A'}(x), \mu_A(x)], \max_y \min[\mu_{B'}(y), \mu_B(y)], \mu_C(z)\}$$

$$\mu_{C'}(z) = \min[w_{A,A'}, w_{B,B'}, \mu_C(z)]$$

Siendo $w_{A,A'}$ y $w_{B,B'}$ los grados de compatibilidad para los antecedentes A y B, respectivamente. El resultado de los grados de compatibilidad de los antecedentes se denomina fuerza de disparo de la regla:

$$f_u((A',B');(A,B)) = \min[w_{A,A'}, w_{B,B'}]$$

Una representación gráfica de este procedimiento se muestra en la figura 3-12, que utiliza el implicador de Mamdani y la regla composicional max-min. El resultado C' es igual a la función de pertenencia del conjunto borroso C recortada por la fuerza de disparo de la regla.

$$\mu_{C'}(z) = \max \{ \min[w_{A_1, A'}, w_{B_1, B'}, \mu_{C_1}(z)], \min[w_{A_2, A'}, w_{B_2, B'}, \mu_{C_2}(z)] \}$$

y en términos de las fuerzas de disparo, tenemos:

$$\mu_{C'}(z) = \max \{ \min[f((A', B'); (A_1, B_1)), \mu_{C_1}(z)], \min[f((A', B'); (A_2, B_2)), \mu_{C_2}(z)] \}$$

Denominamos consecuente cualificado al conjunto borroso que se obtiene cuando se opera la fuerza de disparo y el consecuente de una regla. Llamaremos salida conjunta a la agregación de todos los consecuentes cualificados.

La figura 3-13 muestra gráficamente el razonamiento borroso para múltiples reglas con múltiples antecedentes, utilizando el implicador borroso de Mamdani y la regla de inferencia composicional max-min. Como se puede apreciar, en cada una de las reglas se realiza la inferencia empezando por la obtención de los grados de compatibilidad (operación max-min) representado por $w_{A_1, A'}$ y $w_{B_1, B'}$ en la primera y $w_{A_2, A'}$ y $w_{B_2, B'}$ en la segunda. A continuación se aplica el operador "Y-lógico" mediante una T-norma, en este caso la T-norma mínimo. El resultado es la fuerza de disparo de la regla (w_1 y w_2 para la regla uno y dos respectivamente) que es utilizada junto a la función de pertenencia del consecuente y un método de inferencia para obtener el consecuente cualificado (C_1' y C_2' para la regla uno y dos respectivamente). La regla composicional de inferencia utilizada es la T-norma mínimo. Los consecuentes cualificados obtenidos se observan a la derecha de cada una de las filas que representan las reglas. La agregación de los consecuentes cualificados se realiza en este caso mediante la aplicación de la S-norma máximo, cuyo resultado se observa en la parte inferior de la figura 32. El borde superior del área sombreada es la función de pertenencia asociada al conjunto borroso resultado de la inferencia con múltiples reglas y múltiples antecedentes.

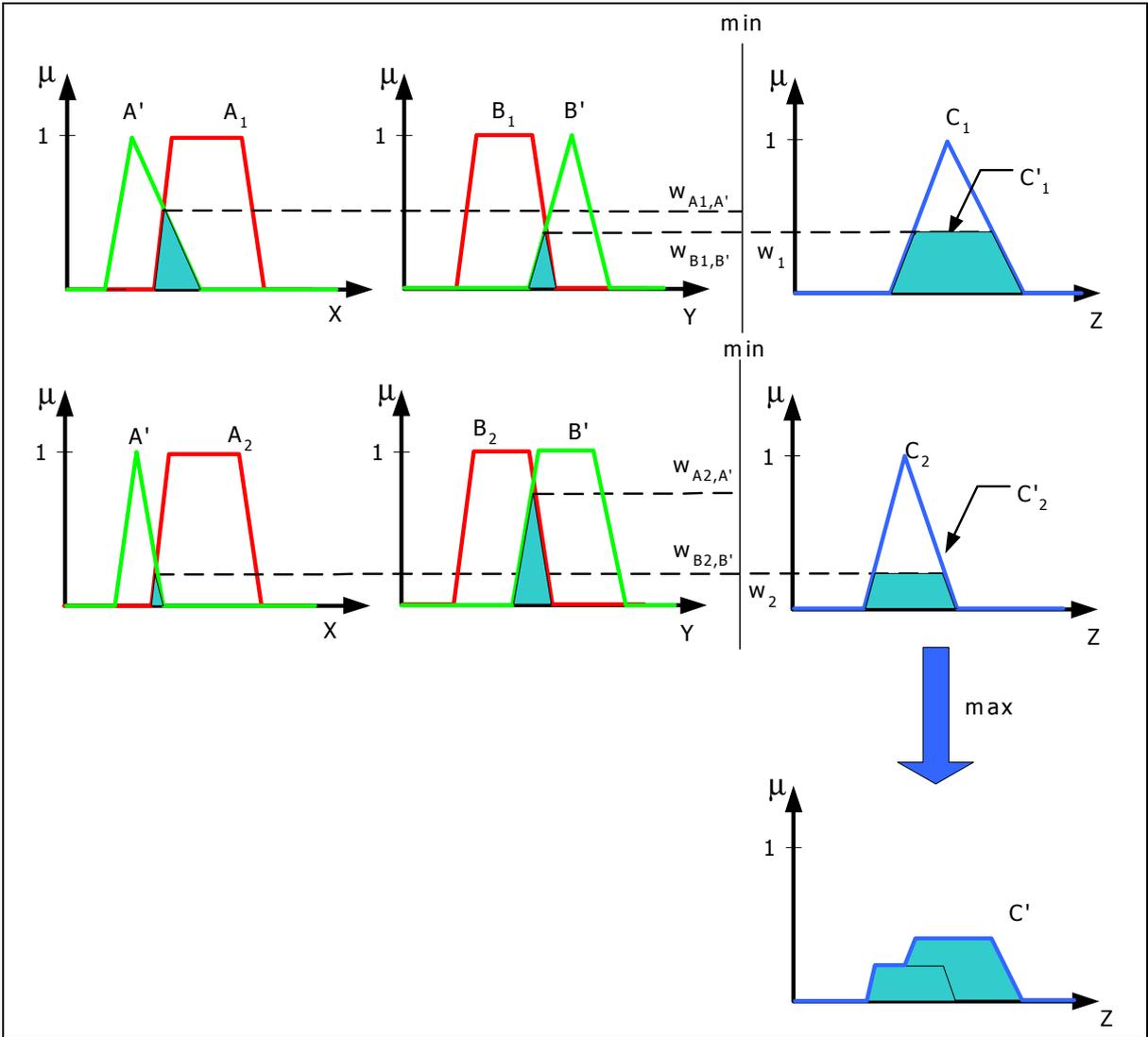


Figura 3-13.- Interpretación gráfica del Modus Ponens Generalizado para varias reglas con múltiples antecedentes, utilizando el implicador borroso de Mamdani (mínimo) y el operador composicional max-min.

3.10 Sistemas de inferencia borrosa

Los sistemas de inferencia borrosa constituyen una clase de algoritmos de cómputo basados en conjuntos borrosos y en el razonamiento aproximado.

Se componen de tres elementos. En primer lugar un conjunto de reglas lógicas, con múltiples antecedentes y un consecuente, denominado base de reglas. En segundo lugar el llamado diccionario que contiene la definición de los conjuntos borrosos asociados a los antecedentes y consecuentes de las reglas. Por último, hay que definir un mecanismo de inferencia.

Las entradas a este tipo de sistemas pueden ser tanto números concisos (*crisp*) como conjuntos borrosos. Cuando se trata de un número conciso hay que o bien considerarlo como un conjunto borroso particular, esto es, con pertenencia 1 para un solo valor de su universo y 0 en el resto (*singleton*), o bien convertirlo a un número borroso en un proceso llamado borrosificación (*fuzzyfication*). La salida del sistema es un conjunto borroso que se obtiene a partir de las entradas, las reglas y el mecanismo de inferencia elegido. Sin embargo, en muchas aplicaciones es necesario que la salida sea un valor numérico y no un número borroso. En esos casos se realiza un proceso de desborrosificación mediante el cual, el conjunto borroso se convierte a un valor del universo de salida representativo de la conclusión obtenida.

En la figura 3-14 se presenta el diagrama de bloques de un sistema de inferencia borroso [Jang 1997]. El proceso básico puede resumirse en las siguientes etapas: obtención de los grados de compatibilidad de las entradas con los antecedentes de las reglas, obtención de las fuerzas de disparo de las reglas, agregación de los consecuentes cualificados y finalmente, en los casos en los que se requiera la desborrosificación.

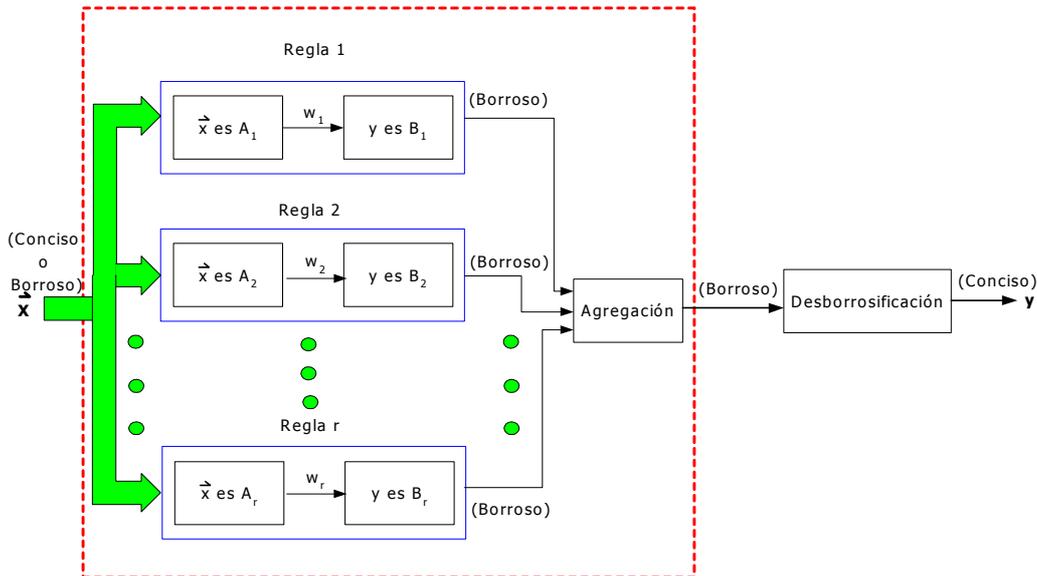


Figura 3-14.- Diagrama de bloques de un sistema de inferencia borrosa. El cuadro rojo agrupa los componentes de un sistema de inferencia básico con salida borrosa.

Una clasificación de los tipos más usuales de sistemas de inferencia borrosa se basa en el modelo utilizado para representar el consecuente de las reglas y en el método de agregación aplicado. Uno de los más utilizados y que es el que hemos utilizado en el planificador instruccional es el de Mamdani.

3.10.1 MODELO BORROSO DE MAMDANI

Este sistema fue propuesto [Mamdani 1975] para el control de una máquina de vapor mediante un conjunto de reglas de control lingüísticas obtenidas de la experiencia de los operadores de la máquina.

Este modelo toma como representación de los consecuentes, conjuntos borrosos que representan las valoraciones de una variable lingüística. Se admiten múltiples reglas con múltiples antecedentes. Cada regla debe tener el mismo conjunto de variables lingüísticas representadas en sus antecedentes. El método de inferencia borrosa, como se ha indicado anteriormente, admite un número de variantes. Concretamente en el modelo de Mamdani, estas variantes se refieren al operador "Y-lógico" donde normalmente se toma una T-norma, el operador "O-lógico" se suele emplear una S-norma, el operador implicación normalmente implementado con una T-norma para calcular el consecuente cualificado y el

operador de agregación corrientemente definido por una S-norma. En la figura 3-15 se puede observar un ejemplo.

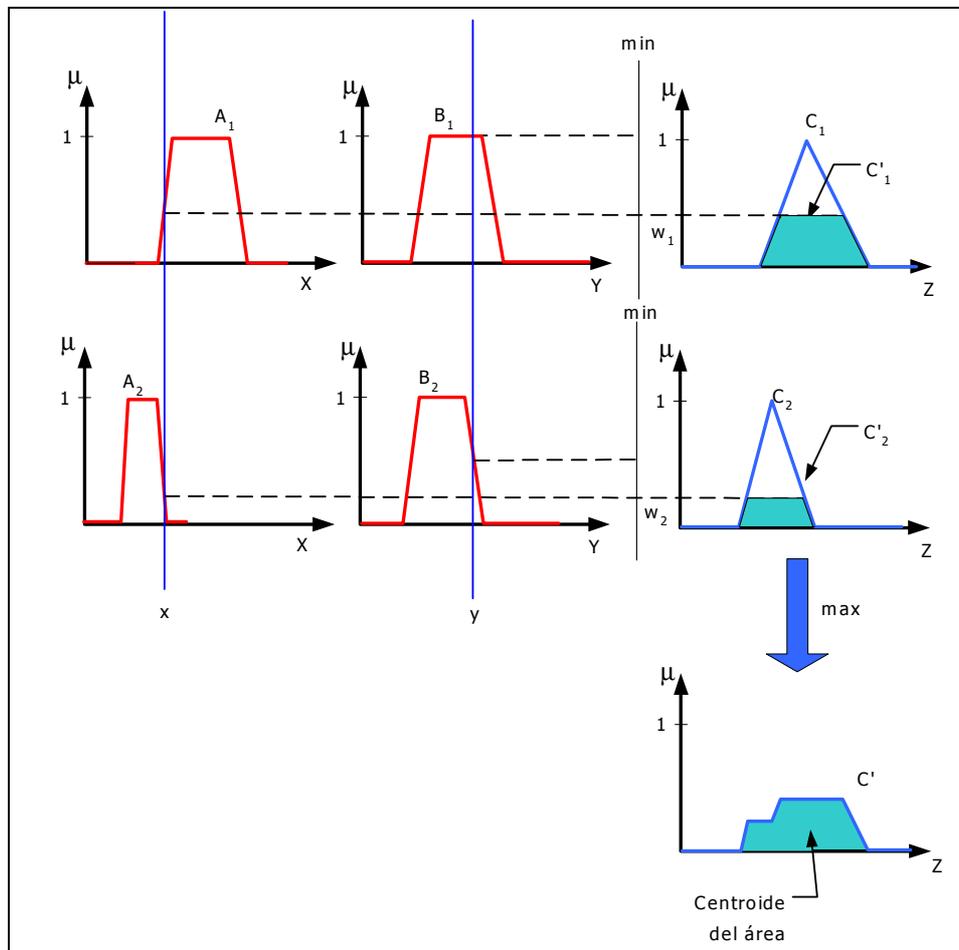


Figura 3-15.- Sistema de inferencia borroso de Mamdani utilizando el mínimo y máximo para la T-norma y S-norma, respectivamente.

En este modelo se suele utilizar un desborrosificador para convertir el conjunto borroso de salida en un valor conciso. Existen diversos métodos para realizar este proceso, cuyos resultados se pueden ver en la figura 3-16:

Desborrosificación basada en el centroide del área: se trata de calcular el centroide del conjunto borroso resultado de la agregación de todos los consecuentes cualificados.

$$z_{ca} = \frac{\int z \mu_A(z) dz}{\int \mu_A(z) dz}$$

Bisector del área: se trata de encontrar el valor numérico del elemento del universo del discurso que separa el área del conjunto borroso en dos mitades iguales.

Media de los máximos: se buscan aquellos elementos del universo del discurso en donde la función de pertenencia del conjunto borroso tome su valor máximo, y se calcula la media de estos puntos.

Mínimo de los máximos: el procedimiento es igual al anterior, únicamente varía en que se toma el menor de los puntos del universo del discurso en lugar de la media.

Mayor de los máximos: Idem caso anterior pero esta vez se toma el mayor de los puntos.

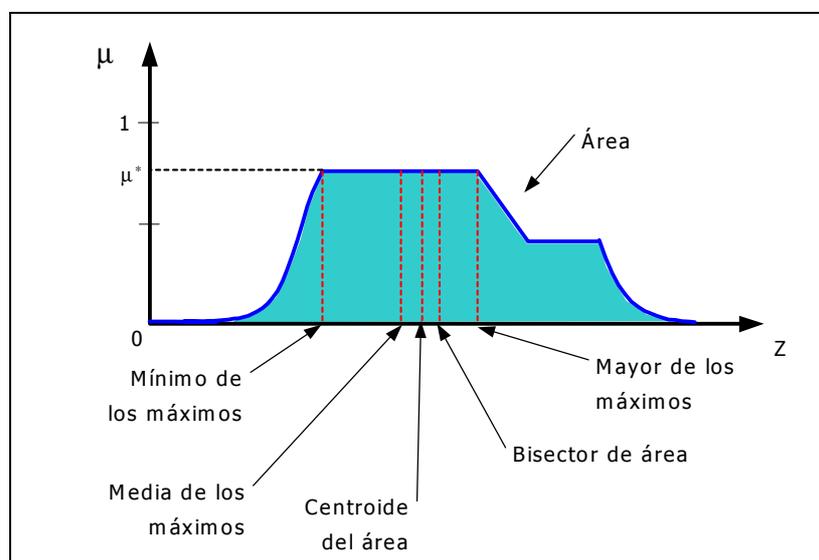


Figura 3-16.- Diferentes métodos de desborrosificación para obtener un valor conciso a la salida.

4 TEORÍA DE SISTEMAS MULTIAGENTE

El origen del concepto de agente surge en 1977 gracias a Carl Hewitt, el cual propone un objeto autocontenido, que se puede ejecutar de forma concurrente con otros e interactuar de la siguiente forma:

Es un agente computacional que tiene una dirección de correo y un comportamiento. Los actores se comunican mediante el paso de mensajes y llevan a cabo sus acciones de forma concurrente [Hewitt 1977].

Existen muchas definiciones de autores y organizaciones relativas al término agente. Ello es debido a que los investigadores que trabajan en este campo no poseen en exclusiva el término y a que además bajo dicho término se pueden incluir investigaciones muy heterogéneas [Nwana 1995].

A continuación vamos a citar algunas de las definiciones más significativas que se han aportado al término agente:

- Los agentes autónomos son sistemas capaces de actuar autónomamente y con determinación en el mundo real [Brustolini 1991].
- Un agente es un software persistente dedicado a un propósito específico [Smith 1994].
- Los agentes autónomos son sistemas computacionales que habitan en algún entorno dinámico complejo, perciben y actúan autónomamente en ese entorno, y haciendo eso, llevan a cabo un conjunto de metas o tareas para las que han sido diseñados [Maes 1994].
- Los agentes inteligentes llevan a cabo tres funciones: percibir las condiciones dinámicas en el entorno, actuar para afectar a las condiciones de dicho entorno y razonar para interpretar lo que percibe, resolver problemas, inferir y determinar acciones [Hayes 1995].
- Los agentes software son programas que se comunican a través de diálogos, son autónomos e inteligentes, deben de ser robustos, no invariantes en el tiempo y distribuidos en redes [Coen 1995]

- Un agente es una entidad que reside en entornos donde interpreta datos que reflejan eventos y ejecuta comandos que producen efectos en dicho entorno [FIPA 1996].
- Un agente autónomo es un sistema situado en un entorno, que siente ese entorno y actúa sobre él a través de tiempo, en busca del cumplimiento de su propia agenda y que de esa forma afecta a lo que sentirá en el futuro [Franklin 1996].
- Un agente software es un programa que se comunica con otros a través de un lenguaje de comunicación de agentes [Genesereth 1997].
- Los agentes inteligentes son entidades software que ayudan a la gente y actúan en su nombre. Son autónomos (tienen control sobre sus propias acciones), están orientados a metas (tienen un propósito y actúan según ese propósito), reaccionan ante los cambios de su entorno. Además dichos agentes pueden ser sociables (interactuar con otros agentes), adaptarse (cambiar su comportamiento en función de experiencias anteriores), ser móviles (moverse entre máquinas) y ser creíbles (pueden aparecer ante el usuario como una entidad visible o auditiva y tener aspectos emotivos o de personalidad) [Gilbert 1997].

Una de estas definiciones bastante aceptada es la caracterización de un agente [Woldridge 1994] como aquel sistema informático que satisface las siguientes propiedades (las cuatro primeras se consideran básicas y el resto opcionales):

- Autonomía: definida como la capacidad del agente de existir y actuar independientemente de un usuario o de otros agentes.
- Reactividad ante el entorno: un agente percibe los cambios producidos en su entorno y debe reaccionar en un tiempo preestablecido a estos cambios mientras se adapta a ellos para conseguir su objetivo.
- Sociabilidad: un agente tiene capacidad de interactuar con otros agentes, utilizando para ello algún lenguaje de comunicación entre agentes (Agent Communication Language, ACL).

- **Iniciativa:** un agente no sólo tiene que reaccionar a los cambios que se produzcan en su entorno o a los cambios que le afecten directamente a él, sino que tiene que tomar la iniciativa para actuar guiado por los objetivos que debe de satisfacer.
- **Movilidad:** es la habilidad de un agente de trasladarse a sí mismo en una red manteniendo su estado actual.
- **Credibilidad:** los agentes no proporcionan información falsa de forma intencionada.
- **Benevolencia:** un agente intenta colaborar con otros mientras esta cooperación no vaya en contra de sus objetivos.
- **Racionalidad:** un agente tiene unos objetivos específicos y siempre intenta llevarlos a cabo.
- **Continuidad:** mientras dure su ciclo de vida los agentes continúan ejecutando si se produce el evento adecuado.
- **Robustez:** los agentes deben de ser capaces de recuperarse ante errores del sistema o humanos que provoquen cambios inesperados en el entorno.
- **Inteligencia:** capacidad de aprender.

4.1 Sistemas Multiagente

Un sistema multiagente es un sistema en el que conviven un conjunto de agentes capaces de interactuar entre sí, donde cada uno de ellos tiene sus propios objetivos y deben cooperar o competir entre ellos.

Encontramos en la bibliografía varias razones para la aparición de los sistemas multiagente:

- En muchas tareas sería posible la existencia de un único agente que se encargara de todo, pero dividir la misma entre varios agentes proporciona más modularidad, flexibilidad y mejor posibilidad de modificación. Los sistemas multiagente son por tanto muy adecuados para problemas en los que se pueda usar la fórmula de divide y vencerás, puesto que se puede

asignar a cada agente uno de los subproblemas en los que se divide el problema global.

- El hecho de tener múltiples agentes en el sistema evita que el mismo se quede bloqueado cuando uno de ellos falla, el resto podría seguir funcionando con normalidad.
- Puesto que los agentes pueden ser diseñados como componentes autónomos que se ejecuten en paralelo, esto supone una gran ventaja para las aplicaciones que necesiten computación distribuida. Además el hecho de que el sistema pueda ser abierto posibilita la inclusión de nuevos agentes o la destrucción de los existentes cuando sea necesario durante la ejecución.

El diseño de un Sistema Multiagente es obviamente más complejo que diseñar un sistema que se componga de un único agente puesto que, además de los agentes propiamente dichos, es necesaria la implementación de métodos de coordinación y comunicación entre los agentes del sistema. Para ello se debe establecer un lenguaje de comunicación de agentes (ACL). Los dos lenguajes de comunicación de agentes más extendidos son KQML y FIPA-ACL.

4.1.1 KQML (KNOWLEDGE QUERY MANIPULATION LANGUAGE)

KQML es un lenguaje que permite a los agentes autónomos y asíncronos trabajar conjuntamente para la resolución de problemas y compartir su conocimiento. Tiene dos especificaciones: la primera versión obra de Finin, Weber, Wiederhold y otros, surge en 1993 [Finin 1993] siendo revisada posteriormente por Finin y Labrou en 1997 [Finin 1997].

Se trata de un lenguaje de alto nivel basado en mensajes, siendo un protocolo de intercambio de información independiente de la sintaxis, ontología y lenguaje del contenido del mensaje. Además es independiente del mecanismo de transporte del mensaje y de cualquier otro protocolo de alto nivel (negociaciones entre agentes, etc.) que se defina. Cada uno de los agentes posee y gestiona una base de conocimiento virtual (Virtual Knowledge Base, VKB), permitiéndose que un agente pueda manipular y realizar búsquedas en la VKB de los otros agentes del sistema. Se basa en una lista de paréntesis balanceados [Finin 1994], cuyo elemento inicial es el acto de comunicación implicado llamado performativa. El conjunto de dichas

performativas determinan el tipo de interacciones permitidas en una conversación. En el Apéndice F se encuentra una breve descripción de estas performativas.

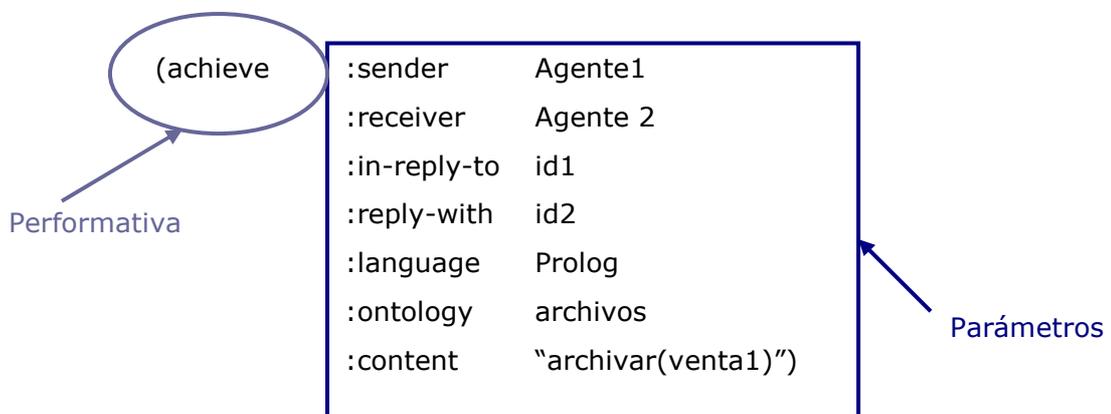
En KQML una conversación se puede definir como el conjunto de todos los mensajes del mismo hilo (secuencia de mensajes ejecutada en paralelo con otras secuencias). Un mensaje se puede dividir en tres capas: de contenido, de mensaje y de comunicación. El contenido contiene el conocimiento que se desea transmitir, el mensaje añade características al contenido como por ejemplo el lenguaje o la ontología que se utilizan, y la capa de comunicación añade los parámetros de comunicación relativos al emisor, receptor, etc.

Un mensaje en KQML se complementa con los parámetros para facilitar el envío y la interpretación de los mensajes de la comunicación (Tabla 3-4).

Sender	Indica el emisor del mensaje.
Receiver	Indica el receptor actual del mensaje.
From	Indica el origen de la performativa en el campo :content cuando se está usando la performativa forward.
In-reply-to	La etiqueta esperada en respuesta a un mensaje previo
Reply-with	La etiqueta esperada en respuesta al mensaje actual
Language	Indica el lenguaje usado en el campo :content
Ontology	Indica la ontología usada en el campo :content
Content	La información del mensaje

Tabla 3-4. Parámetros KQML

Un ejemplo de mensaje KQML en la que el Agente 1 le pide al Agente 2 que archive una venta realizada sería:



4.1.2 FIPA-ACL (FIPA AGENT COMMUNICATION LANGUAGE)

FIPA (Foundation for Intelligent Physical Agents) es una organización internacional sin ánimo de lucro con sede en Ginebra creada en 1995. Su objetivo es promover el uso de los agentes inteligentes desarrollando especificaciones para la comunicación entre agentes (y aplicaciones derivadas de ellas).

El lenguaje de comunicación de agentes FIPA-ACL es una especificación FIPA desarrollada como evolución del KQML.

Un mensaje se compone de varios elementos siendo el único obligatorio el de performativa (todos los demás son parámetros opcionales). El valor de este campo nos permite determinar que tipo de mensaje se está enviando o recibiendo. La descripción de las performativas se puede encontrar en el Apéndice G.

Los parámetros de un mensaje FIPA-ACL se pueden dividir en varias categorías: Relativos a los participantes en la comunicación, al contenido del mensaje, a la descripción del contenido, al control de la conversación y parámetros reservados (Tabla 3-5).

Las conversaciones entre agentes siguen una secuencia típica de mensajes intercambiados que se llama protocolo de conversación. Su explicitud constituye una diferencia sustancial con respecto a KQML puesto que KQML no permite el seguimiento inmediato de un protocolo al no definir el parámetro *:protocol*. En FIPA este parámetro permite identificar el protocolo de comunicación empleado por ambas partes en la conversación actual controlándose de este modo el flujo lógico de mensajes.

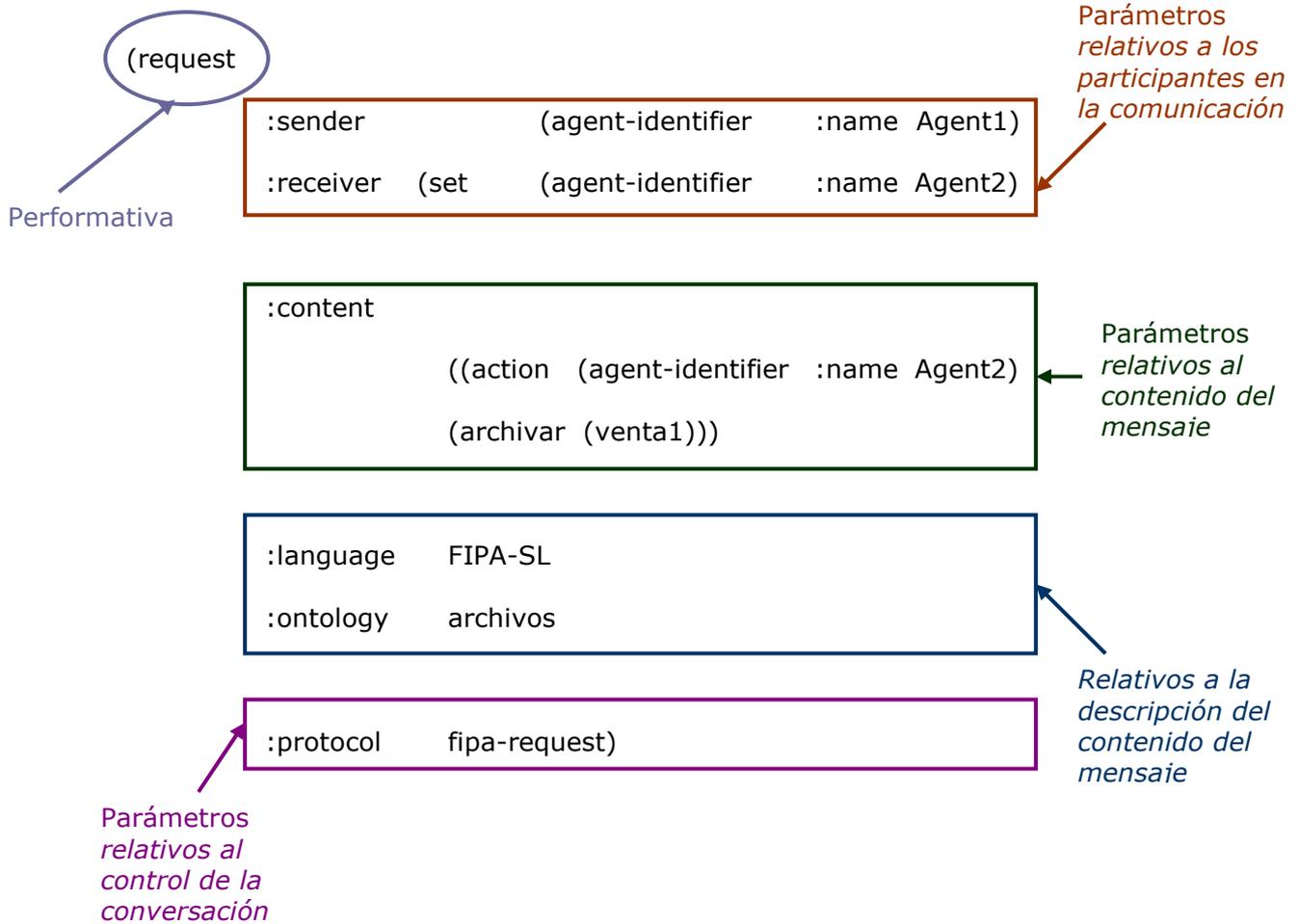
FIPA define once protocolos de interacción los cuales son definidos con mayor detalle y de un modo más formal que en las conversaciones KQML. Estos protocolos son los siguientes, los cuales son descritos con detalle en el Apéndice H: FIPA Propose, FIPA Request, FIPA Request When, FIPA Query, FIPA Contract Net, FIPA Iterated Contract Net, FIPA English Auction, FIPA Dutch Auction, FIPA Brokering, FIPA Recruiting y FIPA Subscribe. Además el modelo de comunicación FIPA se basa en la suposición de que dos agentes que desean conversar entre sí, comparten una ontología común en sus discursos. Según la definición dada por Gruber en 1993, una ontología es una descripción de los conceptos y relaciones que pueden existir para un agente o comunidad de agentes [Gruber 1993]. La declaración explícita de

ontologías tiene como ventajas que permite la consulta sobre conceptos, la actualización, reutilización y la interoperabilidad de MAS heterogéneos [González 2006].

Relativos a los participantes en la comunicación	
<i>Sender</i>	Indica quien es el emisor del mensaje
<i>Receiver</i>	Indica a quien va destinado el mensaje
<i>Reply-To</i>	Indica que los mensajes posteriores a esta conversación tienen que ser dirigidos al agente que aparezca en este campo en lugar de al que aparezca en el sender.
Relativos al contenido del mensaje	
<i>Content</i>	Indica el contenido del mensaje
Relativos a la descripción del contenido del mensaje	
<i>Lenguaje</i>	Indica el lenguaje en el que está escrito el contenido del mensaje.
<i>Encoding</i>	Indica la codificación del lenguaje en el que está escrito el mensaje.
<i>Ontology</i>	Indica la ontología empleada en el mensaje.
Relativos al control de la conversación	
<i>Protocol</i>	Indican el protocolo de la interacción que el emisor está empleando (FIPA Propose, FIPA Request, FIPA Query, ...).
<i>Conversation-ID</i>	Indica un identificador de conversación. Este se puede emplear para el seguimiento y el análisis de las conversaciones por parte de los agentes.
<i>Reply-With</i>	Introduce una expresión que será empleada por el agente que responda al mensaje actual para hacer referencia al mismo.
<i>In-Reply-To</i>	Expresión para responder a Reply-With
<i>Reply-By</i>	Introduce un tiempo máximo de espera para el mensaje de respuesta.
Reservado	
<i>Envelope</i>	Parámetro reservado para la información del transporte del mensaje.

Tabla 3-5. Parámetros FIPA-ACL

El mismo ejemplo del apartado 3.1.1 en la que el Agente 1 le pide al Agente 2 que archive una venta pero en lenguaje FIPA-ACL sería:



4.1.3 COMPARATIVA ENTRE KQML Y FIPA-ACL

Se pueden establecer las siguientes comparaciones entre KQML y FIPA-ACL [Vasudevan 1998, Labrou 1999, FIPA 2001a, McEntire 1998, Cohen 1995]:

- Ambos lenguajes presentan diferencias sintácticas menores, pero tienen una semántica diferente. Esto se debe, entre otras cosas, al diferente

grado de formalización en la definición de los actos de comunicación. No se puede establecer una correspondencia entre las performativas KQML y las de FIPA-ACL, aunque es posible establecer algunas equivalencias entre performativas KQML y actos de comunicación FIPA-ACL (por ejemplo *ask-if* en KQML y *query-if* en FIPA-ACL). En KQML no hay impedimentos para la creación de nuevas performativas pero hay que tener en cuenta la vaguedad en las definiciones de las mismas en este lenguaje.

- En KQML un agente puede inferir directamente en la VKB de otro agente mientras que en FIPA-ACL dicha inferencia está prohibida.
- Existen diferencias en el tratamiento de los actos de comunicación relativos a la gestión y administración de los sistemas. Las performativas *register*, *unregister*, etc. de KQML se ven sustituidas en FIPA-ACL por actos de comunicación *request* con contenido reservado.
- FIPA-ACL tiene un mayor poder de composición de nuevas primitivas para al descripción de los estados de los agentes. La debilidad de KQML es la falta de restricción sobre el lenguaje del contenido.

Según Labrou y Finin [Labrou 1999], existen tres aspectos fundamentales en los que KQML debe mejorar:

- Inclusión de performativas que abarquen otros aspectos de una comunicación como: *offer*, *accept*, etc.
- Empleo de nuevos parámetros como *protocol*, *version*, *reply-by*, *signature*, ...
- Convenciones para las conversaciones como, por ejemplo, una forma de poder determinar los valores por defecto de los parámetros de la conversación.

4.1.4 ARQUITECTURAS DE AGENTES

Además de la comunicación entre agentes es necesario definir la arquitectura de un Sistema Multiagente, es decir, la distribución de los agentes, la organización, gestión, etc. De todos los modelos existentes los tres más significativos que vamos a destacar son una arquitectura de agentes validada por la experimentación empleando lenguaje KQML [González 2004], y los estándares FIPA [FIPA 2001b] y OMG MASIF [OMG 2000]. De estos tres modelos se descartó en primer lugar la opción de un sistema multiagente basado en KQML, puesto que el estándar se restringe únicamente al lenguaje de comunicación entre los agentes. El uso de una arquitectura estándar proporciona al sistema una mayor robustez y facilidad de inclusión de nuevos agentes. La opción FIPA parece en el problema en cuestión más adecuada que la OMG MASIF, ya que no se van a emplear agentes móviles. Por ello, para permitir una mayor estandarización y compatibilidad se decide cumplir con los estándares FIPA, ampliamente aceptados y documentados, relativos no únicamente al lenguaje, sino a casi todos los aspectos de un sistema multiagente como son los protocolos de comunicación, el sistema de transporte de mensajes y la arquitectura de la plataforma.

FIPA es una organización cuyo propósito es promover el desarrollo de especificaciones genéricas para los agentes. Dicha organización define la Plataforma de Agentes (Agent-Platform, AP) que constituye la infraestructura física sobre la que se van a desarrollar los agentes, pero sólo describiendo su comportamiento externo, de forma que cada desarrollador pueda diseñar libremente su aplicación [Alonso 2000, FIPA 2002]. Dicha Plataforma de Agentes consiste en la máquina (o máquinas), el sistema operativo, el software que soporta a los agentes, los componentes de gestión de los agentes FIPA y los propios agentes. De hecho, son los agentes los elementos principales de dicha plataforma. En ella, cada agente debe tener un propietario y tener una identidad. El Identificador de Agente (Agent Identifier, AID) es la etiqueta que permite distinguir de forma inequívoca a cada agente dentro de la plataforma.

Los componentes de gestión de agentes FIPA que componen la Plataforma de Agentes son (figura 3-17):

- El Agent Management System (AMS) ofrece servicios de páginas blancas manteniendo un directorio que contiene las direcciones de los agentes registrados en la plataforma.
- El Directory Facilitator (DF) que proporciona servicios de páginas amarillas a los agentes para que los agentes que lo deseen puedan encontrar servicios ofrecidos por otros agentes.
- El Message Transport Service (MTS) es el método de la comunicación por defecto entre agentes en diversas plataformas. Los servicios de transporte son proporcionados por el Agent Communication Channel (ACC). Este tiene la ventaja de que puesto que el modelo de comunicación entre agentes es asíncrono no se va a quedar bloqueado ante el envío o recepción de mensajes.
- Además también existe el Internal Platform Message Transport (IPMT) que se encarga de toda la comunicación que se produce entre dos agentes de la misma plataforma.

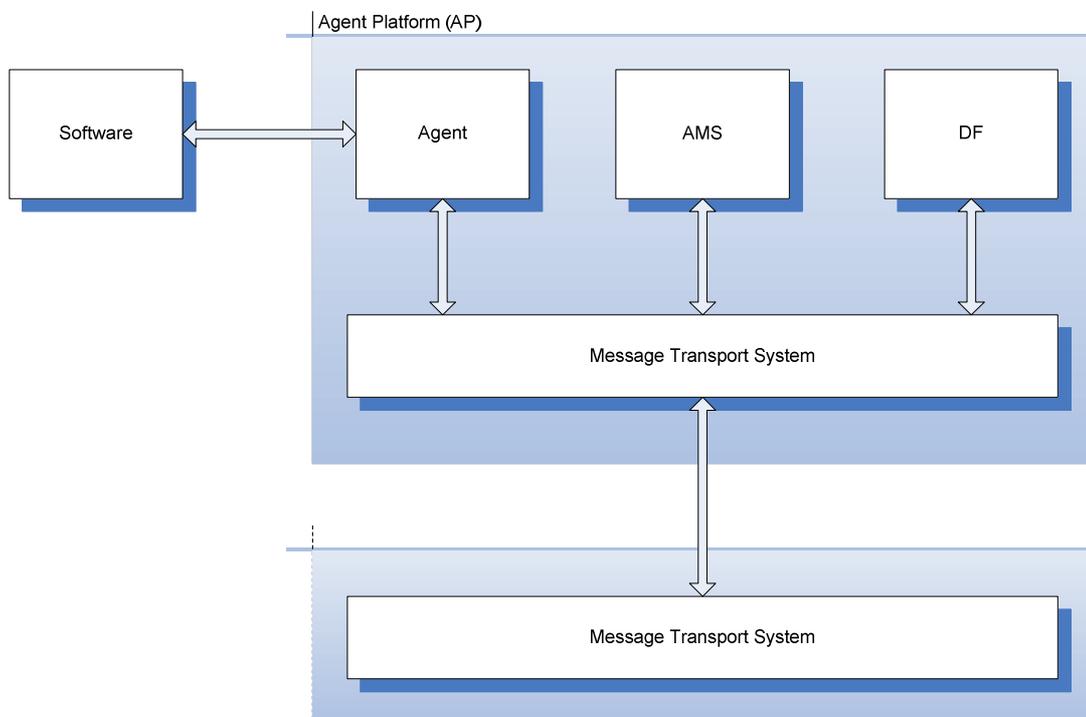


Figura 3-17. Modelo FIPA

Un agente tiene dentro de la Plataforma de Agentes un ciclo de vida que cumple las siguientes características:

- Está ligado a la Plataforma de Agentes: El ciclo de vida de un agente estático está siempre ligado a una Plataforma de Agentes específica.
- Es independiente de la aplicación: El modelo de ciclo de vida de un agente es independiente de cualquier aplicación.
- Es orientado a instancias: El agente descrito en el modelo de vida se considera una instancia (tiene un nombre único y es ejecutado de forma independiente).
- Es único: Cada agente sólo puede estar en un único estado en cada instante dentro del ciclo de vida.

La figura 3-18 muestra el ciclo de vida de un agente. Los estados por los que pasa un agente en su ciclo de vida son Activo, Iniciado, En Espera, Suspendido, En Tránsito y Desconocido. Según el estado del agente el modo de reparto del agente se puede modificar:

- Activo: el MTS reparte los mensajes de forma normal.
- Iniciado/En Espera/Suspendido/En Tránsito: El MTS almacena los mensajes en un buffer hasta que el agente alcanza un estado activo (o los reenvía a otra dirección si el sistema lo requiere).
- Desconocido: El MTS almacena los mensajes en un buffer o los rechaza según la política implementada.

Las transiciones entre estados son las siguientes:

- Create: Creación de un nuevo agente.
- Destroy: Finalización de un agente (de forma forzada).

- Execute: Activar un agente que estuviera en estado de tránsito.
- Invoke: Invocación de un nuevo agente.
- Resume: Reactiva al agente después de una suspensión.
- Suspend: Suspensión de un agente.
- Quit: Finalización normal de un agente.
- Wait: Colocación del agente en estado de espera.
- Wake Up: Activación de un agente que estuviera en espera.
- Move: Colocar al agente en tránsito.

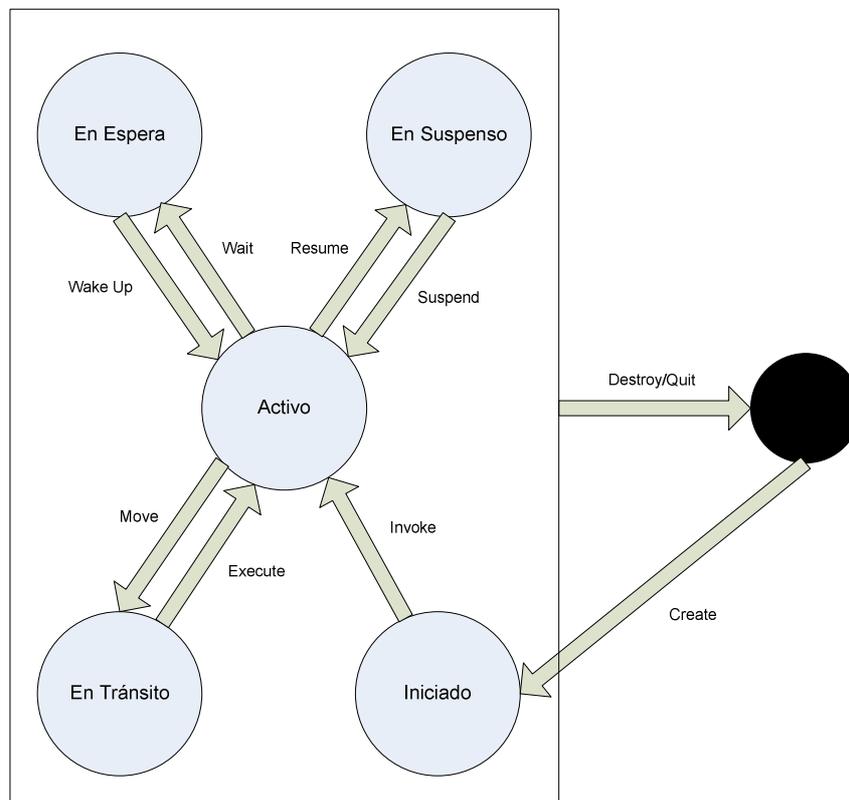


Figura 3-18. Ciclo de vida de un agente.

4.1.5 MARCOS DE TRABAJO DE AGENTES

Los marcos de trabajo de agentes son herramientas de programación para la construcción de agentes [Flores 1999, Piszcz 1998]. La gran mayoría de estas

herramientas han sido desarrolladas en lenguaje de programación Java haciendo uso de su cualidad de lenguaje orientado a objetos multiplataforma y multihilo [Silberschatz 2005]. Esto permite que los agentes puedan residir en varias máquinas, tener diferentes sistemas operativos, y ejecutarse con varios flujos de control. Estas herramientas emplean conceptos de la programación orientada a agentes (AOP). La AOP es una evolución de la programación orientada a objetos (OOP) en la que se toma a un agente como unidad principal de encapsulación. Ésta incorpora diferencias frente a la OOP tradicional [Jennings 1998]:

- En el caso de la OOP la decisión de la ejecución recae en el objeto que invoca al método, mientras que en la AOP esta decisión es del agente que recibe la petición, por lo que poseen autonomía de ejecución. Así los agentes pueden decidir de forma individual responder o no a mensajes que provengan de otros agentes.
- En la AOP cada agente tiene su propio flujo de control mientras que en la OOP hay un único flujo de control para todo el sistema.

De las herramientas existentes, puesto que se ha decidido utilizar la arquitectura FIPA las más representativas son: FIPA-OS [FIPAOS 2001], Zeus [Nwana 1999] y JADE [Bellifemine 1999]. FIPA-OS ha sido elegido debido a:

- Su uso es más intuitivo que el del JADE y de ZEUS.
- Fácil inclusión de nuevos agentes.
- FIPA-OS es software libre.
- Utiliza un hilo por tarea por lo que proporciona mayor robustez.
- Las tareas del mismo agente se pueden distribuir entre varios procesadores.
- FIPA-OS ofrece una clase de Java llamada DIAGNOSTICS que permite el control del flujo de mensajes entre los diversos agentes que facilita la depuración y eliminación de errores.

- La interfaz de sistema experto Jess lo que permite a los usuarios aprovechar las ventajas que presenta esta herramienta para el desarrollo de nuevas utilidades.

A continuación presentaremos una breve descripción de la herramienta elegida.

4.1.6 FIPA-OS

FIPA-OS es un marco de trabajo de agentes desarrollada en los laboratorios de Harlow de la red de Nortel. Implementa las especificaciones de FIPA sobre interoperabilidad de agentes. La plataforma está desarrollada en Java y es software libre bajo licencia EPL, la cual prohíbe su explotación comercial. Las comunicaciones entre los agentes de la misma plataforma se realiza mediante Java RMI y entre los de distintas plataformas usando CORBA, RMI y TCP. Para aprovechar la capacidad multihilo de Java, en la plataforma cada tarea es ejecutada en un hilo distinto de forma que se pueden ejecutar varios flujos de forma concurrente. Además FIPA-OS soporta perfiles de plataformas y de agentes codificados en XML/RDF [Laukkanen 2000].

Se incluyen en FIPA-OS las clases de los agentes de gestión de FIPA vistos en el apartado 3.1.3: el Agent Management System (AMS), el Directory Facilitator (DF), el Agent Communication Channel (ACC) y el Internal Platform Message Transport (IPMT). Existen también diversas utilidades para esta herramienta como por ejemplo: un monitor de las tareas pendientes de ejecución, un visualizador de los hilos activos en el sistema, una interfaz gráfica para enviar mensajes ACL de forma manual a los agentes, etc.

4.1.7 NEGOCIACION EN SISTEMAS MULTIAGENTE

El proceso de negociación entre agentes se puede definir como un proceso mediante el cual se toma una decisión común por dos o más partes. Las partes involucradas primero enuncian demandas contradictorias y entonces se desplazan hacia un acuerdo mediante un proceso de concesiones o de búsqueda de nuevas alternativas [Matos 1998, Sierra 1997, Jennings 1998]. Pueden haber

principalmente dos tipos de agentes: motivados colectivamente los cuales tienen metas comunes y cooperan para llevarlas a cabo y autointeresados que tienen objetivos propios siendo necesaria la negociación para llegar a un acuerdo.

Las características deseables por tanto para la negociación entre los agentes de un sistema son [Hunhs 1999]:

- Eficiencia: Mientras intentan llegar a un acuerdo los agentes no deben desperdiciar recursos.
- Estabilidad: Los agentes no deben por iniciativa propia desviarse de una estrategia acordada.
- Simplicidad: El mecanismo de negociación debería ser simple y con baja demanda computacional.
- Distribución: Es preferible un sistema distribuido por tener mayor robustez.
- Simetría: Todos los agentes deben tener las mismas oportunidades.

Existen diferentes mecanismos de negociación: Votaciones, Subastas, Regateos, Redes de contrato, Coaliciones y Negociaciones basadas en argumentación. A continuación pasamos a describir cada uno de ellos en orden creciente de complejidad.

4.1.7.1 Votaciones

Es un método para la obtención de una elección resultante de las preferencias manifestadas por un conjunto de votantes. Es necesaria una gran cantidad de comunicación entre los agentes por lo que debe ser empleado para un número reducido de agentes o de cuestiones. Existen varios métodos de votación:

- Método de pluralidad: Cada votante manifiesta su preferencia por una opción siendo elegida ganadora la que obtenga un mayor número de votos.

- Método de pluralidad con segunda vuelta: Simular al de pluralidad, pero se escogen las dos opciones con más número de votos. Se realiza una nueva votación entre estas dos opciones y la ganadora es la que obtenga el mayor número de votos.
- Método de pluralidad con eliminación: Se realiza una votación y se elimina una opción no deseada. El proceso se repite hasta que sólo quede una opción que es designada como ganadora.
- Método de votación secuencial por parejas: Se vota entre una pareja de opciones y se elimina la perdedora. A continuación la ganadora se empareja con una nueva opción y se vuelve a votar hasta que sólo queda una opción que sería la ganadora.
- Método de recuento de Borda: Si se tienen N opciones cada votante asigna N-1 puntos para la opción que más le guste, N-2 para la siguiente y así sucesivamente hasta llegar a la última que no recibe ningún punto. Se suman los puntos otorgados por todos los votantes a cada opción y la ganadora es la que haya obtenido más puntos.
- Método de Black: Si existe una opción que gane todas las votaciones una a una con el resto de las opciones existentes se elige como ganadora si no, se toma la opción ganadora según el método de recuento de Borda.
- Método de forma generalizada de Smith: Si se pueden dividir las opciones en dos conjuntos (C1 y C2) de forma que cada opción del primer conjunto (C1) gane a cada opción del otro conjunto (C2) el método de votación no debería escoger como ganadora una opción del conjunto perdedor (C2).
- Método de Nanson: Se elimina recursivamente la opción ganadora con una menor puntuación en el método de recuento de Borda hasta que sólo quede una opción (o un conjunto de opciones con igual puntuación).
- Método de Copeland: Se realiza una votación entre cada pareja de opciones. Se asigna 1 punto a la opción ganadora y -1 puntos a la perdedora de forma que la opción ganadora es la que obtenga más puntos después de todas las votaciones.

4.1.7.2 Subastas

En este tipo de negociación un agente llamado subastador ofrece un bien o recurso y los otros agentes llamados postores desean adquirir dicho artículo al precio más bajo posible. El subastador es quien determina el ganador de la subasta. Los métodos más usados son:

- Subasta inglesa: Cada postor anuncia de forma pública su oferta y cuando ningún agente suba la oferta actual la subasta finaliza. El agente que haya realizado la oferta más alta gana la subasta y paga el precio ofrecido.
- Subasta holandesa: El subastador va bajando continuamente el precio del artículo hasta que uno de los postores acepta el precio y gana la subasta.
- Subasta en sobre cerrado: Cada postor realiza una única puja desconociendo la de los demás. El agente que haya realizado la mayor oferta gana la subasta y paga el precio de la oferta realizada.
- Subasta Vickery: Similar a la subasta en sobre cerrado, pero el precio a pagar por el agente que realice la oferta más alta es el de la segunda oferta más alta.

4.1.7.3 Regateo

Este método de negociación implica que los agentes hagan propuestas y contrapropuestas hasta que se alcance un acuerdo o hasta que se aborte la negociación. Existen cuatro tipos principales de negociaciones basadas en regateo:

- Modelo de negociación bilateral: Es un regateo entre dos agentes con la intención de que se pongan de acuerdo en el objeto de la negociación. Consiste en la sucesión alternada de ofertas entre ambos hasta que se llegue a un acuerdo y la oferta de un agente sea aceptada por el otro; o hasta que uno de los dos agentes de por finalizada la negociación.

- Tácticas dependientes del tiempo: Si se tiene un tiempo límite para alcanzar un acuerdo las concesiones en el regateo deben aumentar según se acerque el final del plazo.
- Tácticas dependientes de los recursos: En este regateo existe una limitación de los recursos, por lo que las ofertas sucesivas dependerán del ritmo de consumo del objeto de la negociación. Esto provoca que los agentes se muestren más dispuestos a llegar a un acuerdo a medida que disminuya el número de recursos disponibles.
- Tácticas dependientes del comportamiento: Los agentes calculan la próxima oferta a realizar basándose en el comportamiento previo de su oponente.

4.1.7.4 Redes de Contrato

Es un protocolo de interacción para la resolución de problemas cooperativos entre los agentes. Este método consiste en encontrar un agente apropiado para cada tarea del sistema. Un agente llamado *manager* quiere que una tarea sea llevada a cabo y para ello negocia con los posibles agentes que pueden realizarla.

4.1.7.5 Coaliciones

Una coalición consiste en un subconjunto de agentes que se agrupan para conseguir un beneficio común y que se disgrega una vez que el objetivo se ha cumplido.

4.1.7.6 Negociación basada en argumentación.

En este tipo de negociación se emplean argumentos para convencer a un agente de que acepte una propuesta.

5 PLANIFICADOR INSTRUCCIONAL BORROSO Y MULTIAGENTE

El planificador debe ser un módulo dinámico que sea capaz de generar planes, monitorizar su ejecución y replanificar cuando sea necesario.

Vamos a ver en primer lugar el funcionamiento del Planificador Instruccional Borroso MultiAgente en su conjunto para posteriormente pasar a describir el diseño y desarrollo de cada una de sus partes en los siguientes epígrafes.

La forma de proceder del Planificador Instruccional sería [Aguilar 2004]:

1. Una vez que el alumno ha resuelto las actividades de un objetivo el Gestor se encarga de activar el Sistema Borroso y de enviarle la información del alumno.
2. El Sistema Borroso calcula cual es el siguiente nivel en el que tendrá que trabajar el Alumno para ese objetivo.
3. Una vez que se ha determinado el nivel el Gestor se encarga de activar el Sistema MultiAgente.
4. Dicho sistema es el que decide en función de los datos del alumno y de su historial el siguiente objetivo y el tipo de actividad (dentro de ese objetivo) en el que va a trabajar el alumno.

El esquema de las interacciones del Planificador Instruccional Borroso MultiAgente que hemos desarrollado se puede ver en la figura 3-19.

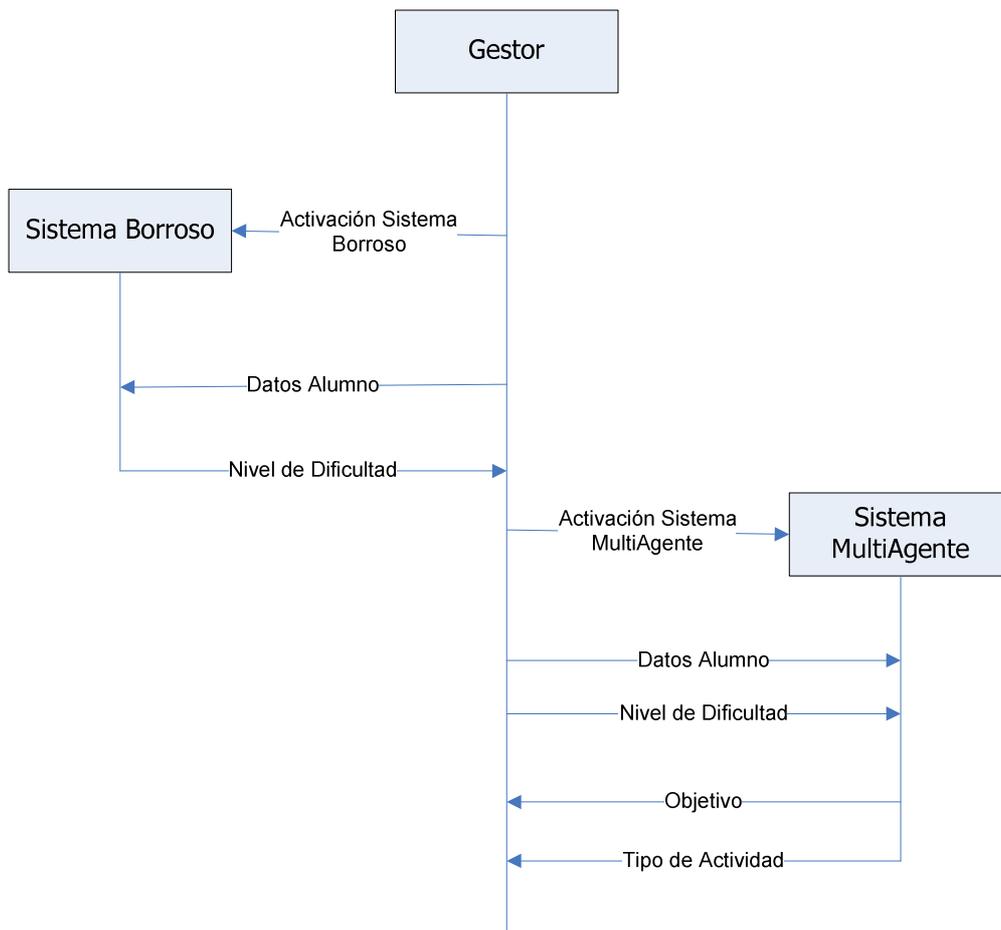


Figura 3-19. Esquema de las interacciones del Planificador Instruccional Borroso MultiAgente

6 EL PLANIFICADOR INSTRUCCIONAL BORROSO

Debido a las características del conocimiento pedagógico a modelar, se ha diseñado la parte del planificador para determinar el nivel de dificultad en el que deberá trabajar el alumno utilizando lógica borrosa. Se propone un diseño como el que se muestra en la figura 12, en el que conocido el objetivo de aprendizaje del alumno y los resultados de las acciones realizadas en el instante actual (estado alumno) y en el pasado (historial alumno) el planificador decidirá el nivel de complejidad de las actividades que se le presentarán al alumno.

El uso de la metodología de Lógica Borrosa en sistemas como el planificador instruccional tiene una aplicación inmediata debido a que su comportamiento se conoce en base a reglas definidas de forma poco precisa. Esta imprecisión surge de la complejidad del propio sistema. La manera de atacar dicho tipo de problemas es reducir la complejidad aumentando la incertidumbre sobre las variables. El comportamiento del planificador viene dado por una serie de reglas que a menudo son imprecisas, o que utilizan términos lingüísticos cargados de incertidumbre. Así se formulan reglas del tipo "Si el alumno avanza bien entonces aumentar el nivel de la complejidad de la actividad".

Por tanto, se dispone de una *base de conocimiento* del sistema, esto es, un conjunto de reglas que tratan de modelar las acciones que se han de llevar a cabo en el mismo para conseguir la acción deseada. Dichas reglas las proporciona el *experto pedagogo*, aquél que conoce cómo instruir a cada alumno con sus características particulares, gracias a la experiencia adquirida en el aula.

El planificador borroso (figura 3-20) está formado por los siguientes elementos [Passino 1998]:

- Base de Reglas: conjunto de reglas borrosas de tipo "si-entonces" que cuantifica mediante lógica borrosa las descripciones lingüísticas del pedagogo acerca de cómo enseñar un objetivo.
- Mecanismo de Inferencia: que emula el proceso de toma de decisiones del experto interpretando y aplicando el conocimiento existente para determinar cuál es la mejor acción en una situación dada.
- Interface de borrosificación: que convierte las entradas al controlador en información borrosa que el proceso de inferencia pueda fácilmente usar para activar y disparar las reglas correspondientes.
- Interface de desborrosificación: que convierte las conclusiones del mecanismo de inferencia en las entradas concisas que necesita la interface multimedia para determinar qué actividades se le muestran al alumno.

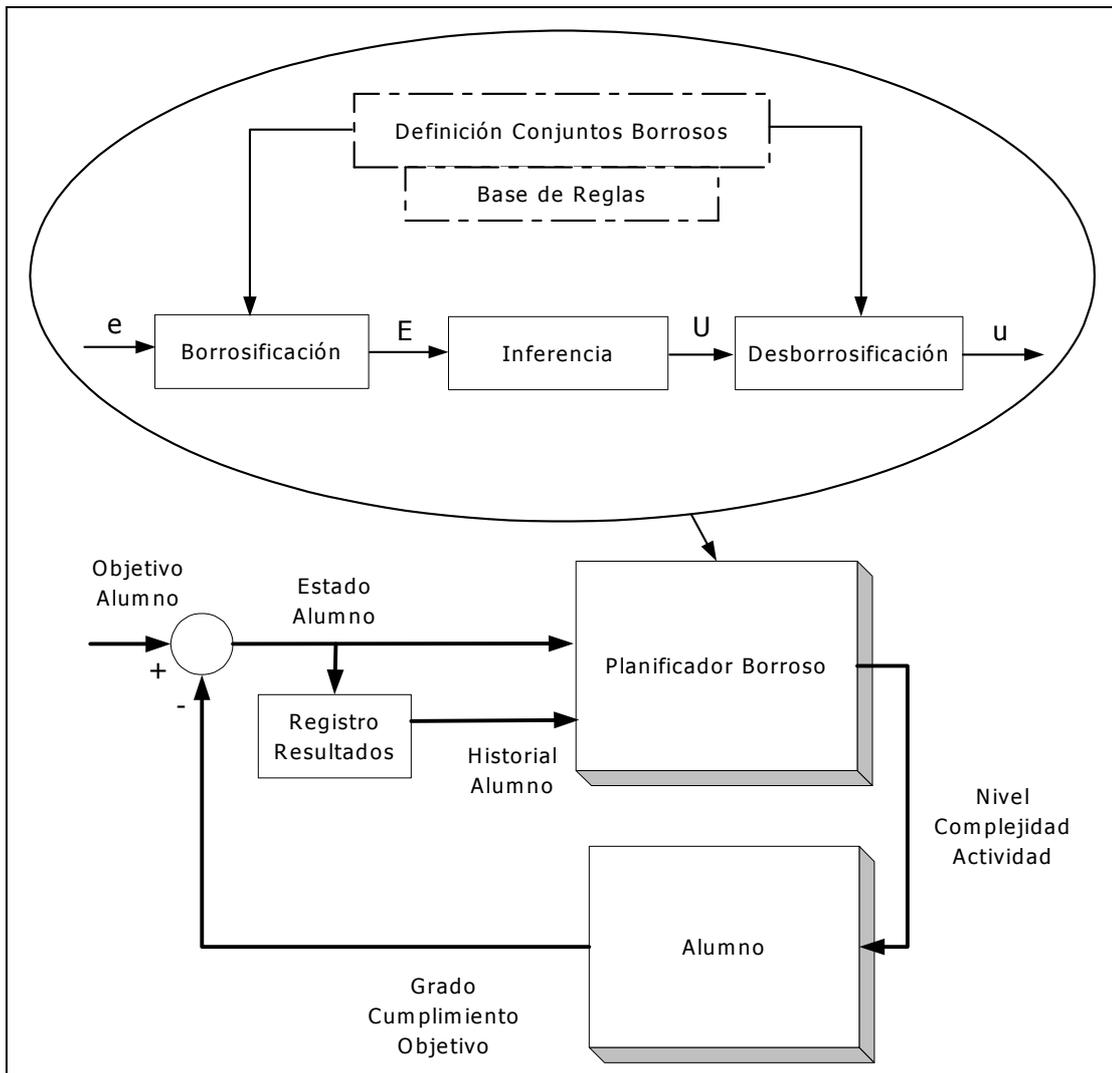


Figura 3-20.- Planificador Instruccional Borroso como elemento de toma de decisiones en la gestión del conocimiento a presentar al alumno.

A continuación vamos a introducir cada uno de estos componentes del planificador borroso siguiendo paso a paso el diseño del mismo.

El planificador instruccional borroso está formado por un conjunto de sistemas borrosos que van a inferir el encadenamiento de actividades a realizar por el alumno en función de los resultados obtenidos por éste. Como se ha mencionado en la teoría de la Lógica Borrosa cada uno de estos sistemas borrosos se compone de tres elementos:

- Un conjunto de reglas lógicas, con múltiples antecedentes y un consecuente, denominado base de reglas
- Un diccionario que contiene la definición de los conjuntos borrosos asociados a los antecedentes y consecuentes de las reglas
- Un mecanismo de inferencia.

Las entradas a nuestros sistemas borrosos serán números concisos que se considerarán como un conjunto borroso con pertenencia 1 para un solo valor de su universo y 0 en el resto. La salida del sistema es un conjunto borroso que se obtiene a partir de las entradas, las reglas y el mecanismo de inferencia elegido. Este conjunto será la realimentación del sistema pero puesto que también necesitamos que la salida sea un valor numérico se realizará un proceso de desborrosificación.

6.1 Esquema del Planificador Instruccional Borroso

Para aplicar las técnicas borrosas en la implementación de un planificador instruccional hay que seguir el siguiente procedimiento que consta de dos etapas bien diferenciadas:

La primera etapa, a realizar antes de que empiece a funcionar el planificador, consiste en:

1. Establecimiento de las variables de entrada y salida del planificador (variables lingüísticas).
2. Definición de los conjuntos borrosos de cada variable.
3. Definición de las funciones de pertenencia de los conjuntos.
4. Establecimiento de la base de reglas.
5. Definición del mecanismo de borrosificación, inferencia y desborrosificación.

La segunda etapa, a realizar en cada paso del algoritmo de toma de decisiones, consiste en:

1. Obtener los valores concisos de las entradas.
2. Borrosificación: Asignación de los valores concisos a los conjuntos borrosos de entrada y cálculo del grado de pertenencia a cada uno de esos conjuntos.
3. Inferencia: Aplicación de la base de reglas y cálculo de los conjuntos borrosos de salida inferidos de los conjuntos de entrada.
4. Desborrosificación: Cálculo de los valores concisos de salida a partir de los conjuntos borrosos inferidos. Estos valores concisos serán las salidas del planificador que informará qué actividades y con qué complejidad debe realizar el alumno.

6.2 Entradas y Salidas del Controlador Borroso

Si consideramos la forma en que un experto pedagogo enseña los diferentes objetivos, tendremos que diseñar un planificador borroso que automatice la forma en que este experto humano realiza su toma de decisiones. Para ello, el experto debe indicar (al diseñador del planificador instruccional borroso) qué información recibe como entrada en su proceso de toma de decisiones.

El profesor que tutoriza a un alumno observa el porcentaje de aciertos en un total de acciones, esto es, nos indica que una de las variables de entrada sería el grado de cumplimiento de los objetivos. La otra variable de entrada, según las indicaciones de los expertos, debería ser el historial del alumno, que informa la forma de aprender del alumno y como es su evolución en los objetivos.

A continuación debemos identificar las variables a controlar. Como el planificador instruccional debe decidir qué actividades debe realizar el alumno y el conjunto de actividades necesarias para el aprendizaje de un objetivo concreto está determinado por el conocimiento de dominio, la salida del sistema borroso debe ser

el nivel de complejidad de las actividades asociadas con el aprendizaje de un objetivo concreto.

Como en la enseñanza de los conceptos de número, suma y resta, se tienen que cubrir una serie de objetivos, debemos construir tantos sistemas borrosos como número de objetivos de aprendizaje tengamos. La estructura de estos sistemas es la que se muestra en la figura 3-21. Las dos entradas al sistema borroso son las vistas anteriormente: el porcentaje de acierto de un conjunto de acciones del alumno (grado de cumplimiento del objetivo), y una variable que recuerda la evolución del alumno. El grado de cumplimiento del objetivo es una entrada concisa cuyo valor está en el rango de 0 a 1 (por ejemplo si el alumno realiza correctamente el 75% de las actividades el valor de de esta entrada sería 0.75) y la evolución del alumno es una entrada borrosa que se corresponde con la salida del sistema borroso antes de desborrosificar (realimentación del sistema) [Moreno 2003]. A partir de las reglas implementadas el sistema es capaz de determinar cuál debe ser el nivel de complejidad de las actividades que se le van a presentar al alumno. Este valor una vez desborrosificado es la salida de nuestro sistema.

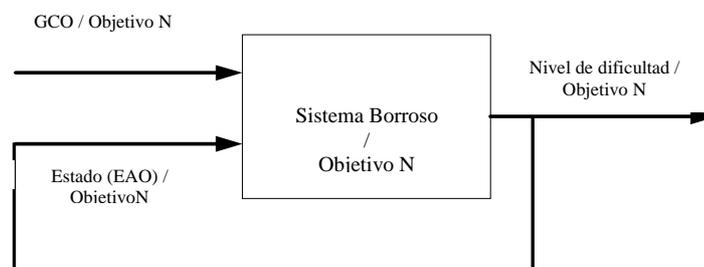


Figura 3-21.- Sistema Borroso para la tutorización del Objetivo N.

Como se ha mencionado en el capítulo II, los objetivos de aprendizaje se agrupan en fases. La transición entre fases (progresión, regresión y permanencia) se realiza mediante un sistema borroso como el que se muestra en la figura 3-22. Las entradas de este sistema borroso serán las salidas sin desborrosificar de cada uno de los objetivos de la fase en la que esté trabajando el alumno (es decir la evolución del alumno en cada objetivo). La salida una vez desborrosificada nos proporciona un número conciso que indica si el alumno permanece en la misma fase, avanza a la siguiente o retrocede a la anterior.

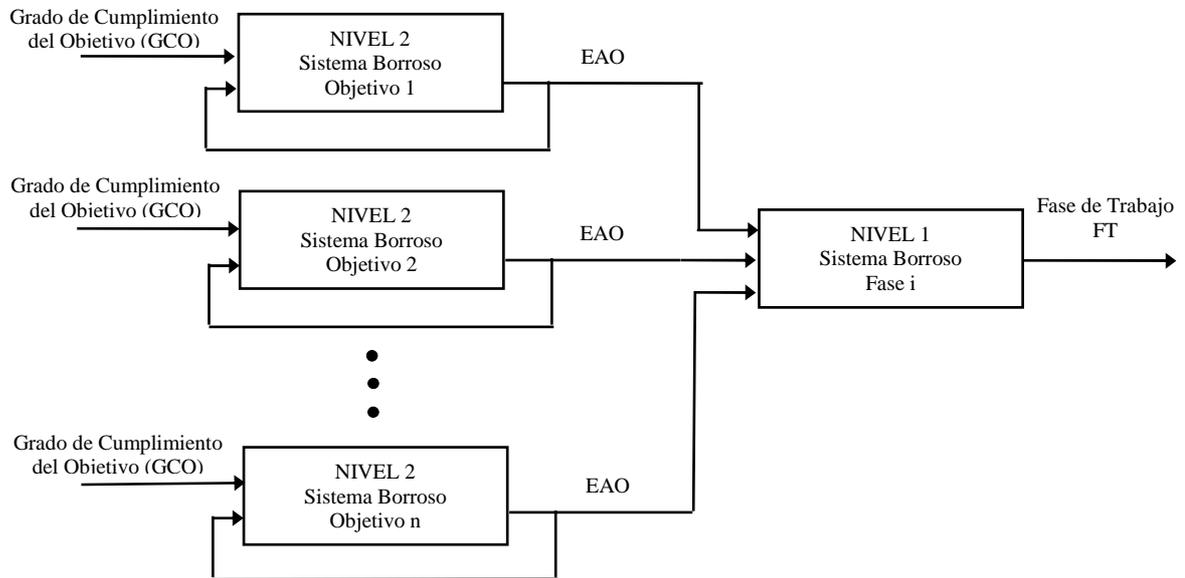


Figura 3-22.- Sistema Borroso que infiere la fase (conjunto de objetivos de trabajo) en la que se encuentra el alumno.

6.3 Descripción lingüística

El experto utiliza variables lingüísticas para describir cada una de las entradas y salidas que varían con el tiempo del planificador instruccional borroso. Así las variables lingüísticas del sistema borroso que trata cada objetivo de aprendizaje se pueden observar en la tabla 3-6. Mientras que la variable lingüística para el sistema borroso de cambio de fase se muestra en la tabla 3-7.

"Grado de Cumplimiento del Objetivo"	GCO
"Evolución de las Actividades del Objetivo"	EAO
"Nivel de complejidad de Actividades del Objetivo"	NCAO

Tabla 3-6. Variables lingüísticas para el Sistema Borroso de cada objetivo de aprendizaje.

"Fase de Trabajo"	FT
-------------------	----

Tabla 3-7. Variable lingüística para la el Sistema Borroso de la Fase de Trabajo.

Hemos utilizado las comillas para enfatizar que ciertas palabras o frases representan la descripción lingüística de variables que varían en el tiempo. Hay muchas posibles elecciones de descripciones lingüísticas para las variables. Sin embargo, la elección de unas u otras no influye en la forma de operar del planificador borroso, únicamente simplifica su construcción mediante lógica borrosa.

Las variables lingüísticas toman "valores lingüísticos", que son los valores de las variables a medida que el tiempo cambia. En nuestro planificador instruccional los valores para el GCO se pueden observar en la tabla 8. Y para las variables lingüísticas "Evolución de las Actividades del Objetivo" (EAO), "Nivel de complejidad de Actividades del Objetivo" (NCAO) y "Fase de Trabajo" (FT) en la tabla 9.

MB	Muy Bajo
B	Bajo
M	Medio
A	Alto
MA	Muy Alto

Tabla 3-8. Valores lingüísticos de la variable GCO

RT	Retroceder
PM	Permanecer
AV	Avanzar

Tabla 3-9. Valores lingüísticos de la variable FT.

A continuación se mostrará cómo se puede describir la instrucción de cada objetivo de aprendizaje a partir de las variables lingüísticas y los valores que toman. En el caso del planificador instruccional cada una de las siguientes frases representa estados diferentes del sistema. Por ejemplo podemos tener que:

- Si el 'Grado de Cumplimiento de Objetivos' es Muy Bajo significa que el alumno no sabe resolver las actividades.
- Si la 'Evolución de las Actividades del Objetivo' es Permanecer significa que el alumno en las anteriores interacciones con el sistema ha respondido más o menos bien.

6.4 Base de Reglas

Las cuantificaciones lingüísticas, definidas anteriormente, se utilizan para especificar un conjunto de reglas que capturen el conocimiento del experto acerca de cómo se realiza la planificación instruccional. Por ejemplo para el planificador borroso que determina el nivel de complejidad de cada objetivo, tendremos reglas como las siguientes:

1.- Si el '*Grado de Cumplimiento del Objetivo*' es Muy Bajo y la '*Evolución de las Actividades del Objetivo*' es Retroceder Entonces '*Nivel de complejidad de Actividades del Objetivo*' es Retroceder.

Esta regla cuantifica la situación en la que el alumno no ha respondido adecuadamente a las actividades propuestas y además sus interacciones anteriores han sido similares, por lo que la estrategia a seguir es disminuir la complejidad de las actividades que se le proponen.

2.- Si el 'Grado de Cumplimiento del Objetivo' es Alto y la 'Evolución de la Actividades del Objetivo' es Retroceder Entonces 'Nivel de complejidad de Actividades del Objetivo' es Permanecer.

Esta regla cuantifica la situación en la que el alumno ha respondido correctamente a las actividades propuestas y como sus interacciones anteriores no han sido muy buenas la complejidad de las actividades propuestas deben ser de nivel medio.

Cada una de las reglas anteriores son "reglas lingüísticas" ya que utilizan variables y valores lingüísticos. Como estos valores lingüísticos no son representaciones precisas de las magnitudes que describen, las reglas lingüísticas tampoco lo son. Simplemente son ideas abstractas de cómo lograr un buen control que pueden representar distintas cosas a diferentes personas. Sin embargo, las reglas lingüísticas son utilizadas muy frecuentemente por los expertos para la toma de decisiones.

Usando reglas del tipo descrito anteriormente podemos definir todas las posibles situaciones que se dan en el plan instruccional. Debido a que hemos utilizado un número finito de variables lingüísticas y de valores lingüísticos, existe un número finito de posibles reglas. Para el problema del planificador instruccional, con dos entradas, una con cinco valores lingüísticos y otra con tres, existen 15 posibles reglas (todas las posibles combinaciones de los valores de las variables lingüísticas).

Una forma cómoda de representar el conjunto de reglas, cuando el sistema borroso no tiene muchas entradas (menor o igual a 3), es mediante una tabla, donde cada casilla representa el valor lingüístico del consecuente de una regla, y la columna de la izquierda y la fila superior contienen los valores lingüísticos de las variables del antecedente. Unos ejemplos para el planificador instruccional dependiendo del tipo de alumno (ver Capítulo III, epígrafe 4.2) se pueden observar en las siguientes tablas: alumnos con miedo al fracaso que avanzan lentamente (Tabla 3-10), alumnos motivados a los que no les afectan los errores que representan el caso intermedio entre los anteriores (Tabla 3-11) y alumnos hiperactivos que avanzan muy rápido (Tabla 3-12).

EAO\GCO	Muy Bajo	Bajo	Medio	Alto	Muy Alto
Retroceder	Retroceder	Retroceder	Retroceder	Permanecer	Permanecer
Permanecer	Retroceder	Retroceder	Permanecer	Permanecer	Avanzar
Avanzar	Retroceder	Permanecer	Permanecer	Avanzar	Avanzar

Tabla 3-10.- Base de Reglas para el Planificador Instruccional Borroso para un alumno con miedo al fracaso.

EAO\GCO	Muy Bajo	Bajo	Medio	Alto	Muy Alto
Retroceder	Retroceder	Retroceder	Permanecer	Permanecer	Permanecer
Permanecer	Retroceder	Permanecer	Permanecer	Permanecer	Avanzar
Avanzar	Permanecer	Permanecer	Permanecer	Avanzar	Avanzar

Tabla 3-11.- Base de Reglas para el Planificador Instruccional Borroso para un alumno hiperactivo.

EAO\GCO	Muy Bajo	Bajo	Medio	Alto	Muy Alto
Retroceder	Retroceder	Retroceder	Permanecer	Permanecer	Avanzar
Permanecer	Retroceder	Permanecer	Permanecer	Avanzar	Avanzar
Avanzar	Permanecer	Permanecer	Avanzar	Avanzar	Avanzar

Tabla 3-12.- Base de Reglas para el Planificador Instruccional Borroso motivado al que no le afectan los errores.

6.5 *Funciones de pertenencia*

Hasta este momento, sólo hemos cuantificado de forma abstracta el conocimiento que tiene el pedagogo acerca de cómo se enseña la suma. A continuación veremos cómo, utilizando la lógica borrosa podemos cuantificar el significado de las descripciones lingüísticas, de forma que podamos automatizar en un planificador instruccional borroso las reglas de toma de decisiones especificadas por el experto.

Vamos a cuantificar el significado de los valores lingüísticos mediante las funciones de pertenencia. Dependiendo de la aplicación concreta y del diseñador (experto), podemos elegir unas u otras funciones de pertenencia.

Inicialmente pensamos usar conjuntos triangulares como aproximación inicial a nuestro problema (figuras 3-23 y 3-24), pero los resultados que se obtuvimos realizando una simulación con ellos no fueron los deseados puesto que se producían oscilaciones en el sistema. Es decir, al ir incrementando paulatinamente los valores de GCO se producían aumentos bruscos en la salida NCAO que no eran deseables y la función resultante no era monótona creciente. En vista de este resultado probamos con otro tipo de conjuntos como los trapezoidales, las campanas y las gaussianas decidiéndonos al final por éstas últimas puesto que eran las que mejor modelaban nuestro problema [Aguilar 2007].

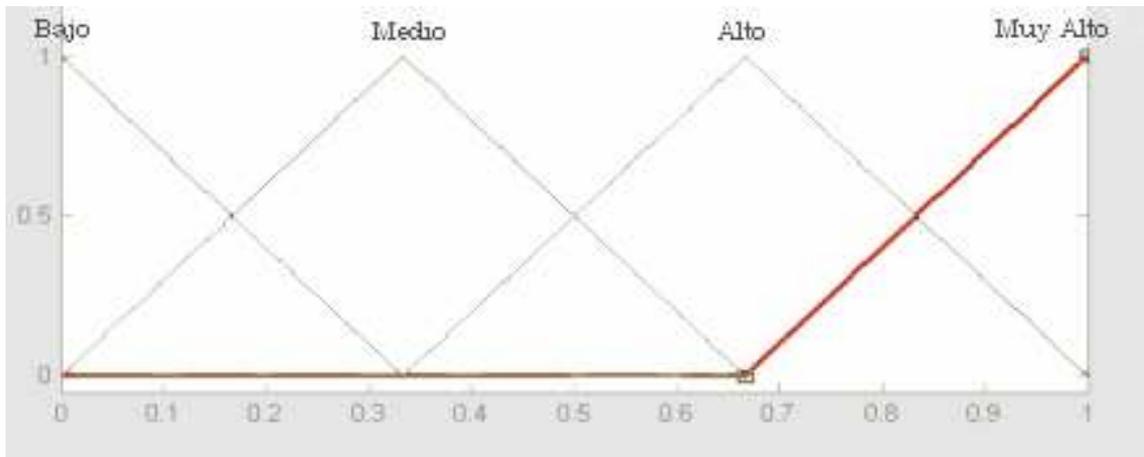


Figura 3-23.- Partición borrosa de la variables de entrada GCO.

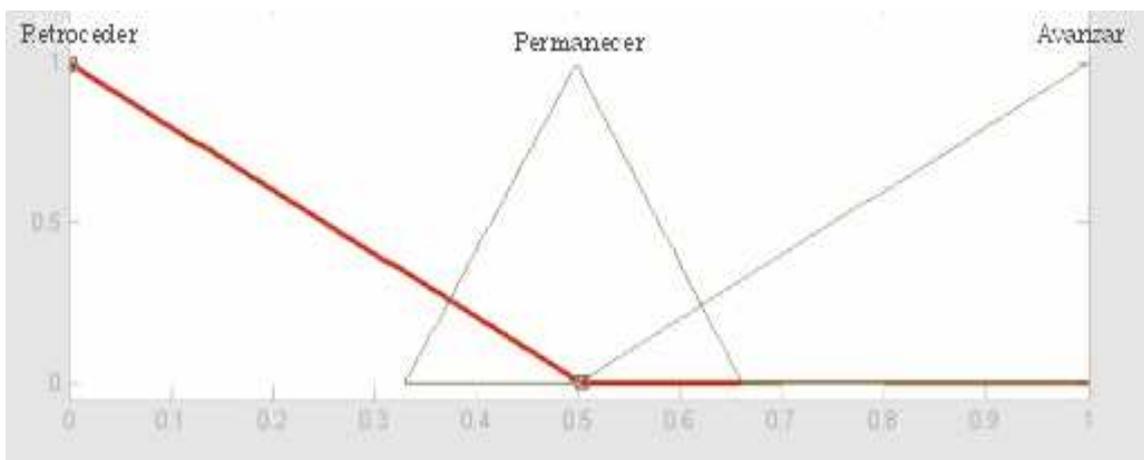


Figura 3-24.- Partición borrosa de la variables de EAO y NCAO.

En las siguientes gráficas se puede observar una comparación de los resultados obtenidos mediante simulaciones del Sistema Borroso para cada tipo de alumnos. La primera muestra los resultados usando los conjuntos borrosos triangulares con los que se probó inicialmente (figura 3-25) y la segunda los gaussianos que usamos finalmente (figura 3-26). Como se puede observar la primera aproximación con los triangulares hacía que el sistema presentara oscilaciones no deseadas en la salida del sistema que son corregidas con el uso de las funciones de pertenencia gaussianas.

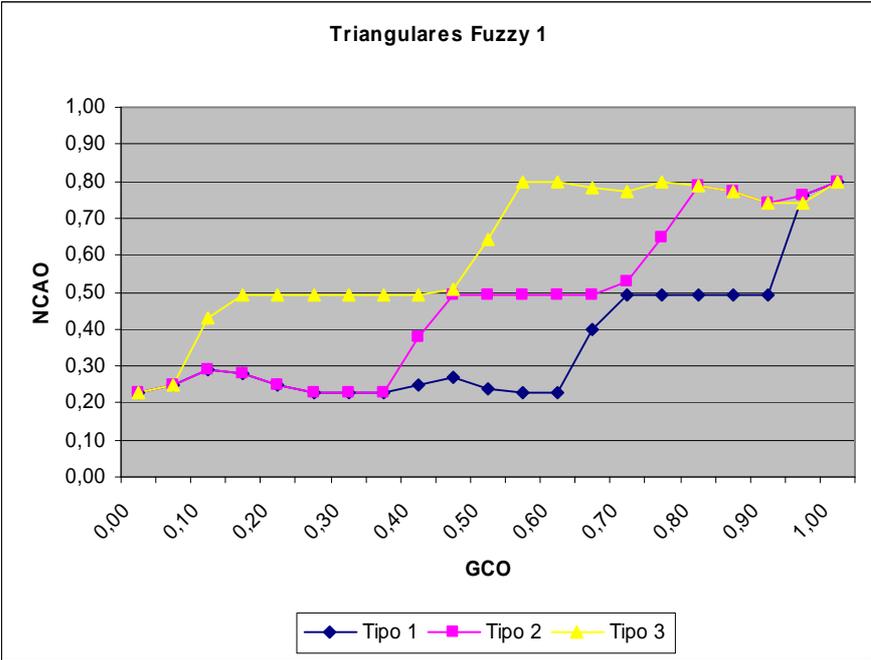


Figura 3-25. Resultados del Planificador Borroso usando conjuntos triangulares.

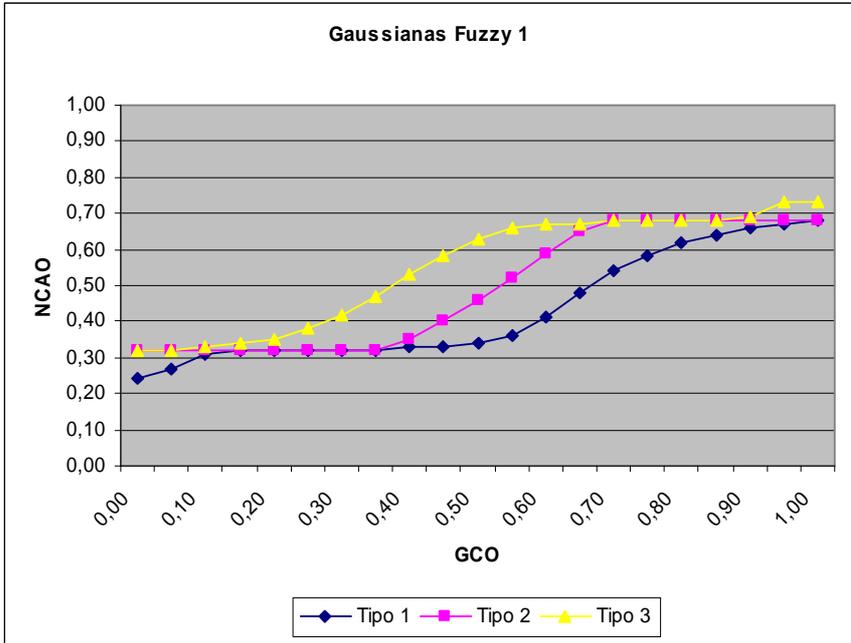


Figura 3-26. Resultados del Planificador Borroso usando conjuntos gaussianos.

Las particiones borrosas para la variable de entrada GCO (Grado de Cumplimiento de Objetivo) de cada sistema borroso que trata los objetivos de aprendizaje, consistirán por tanto en 5 conjuntos difusos gaussianos distribuidos en un universo de discurso normalizado con rango $[0,1]$, figura 3-27.

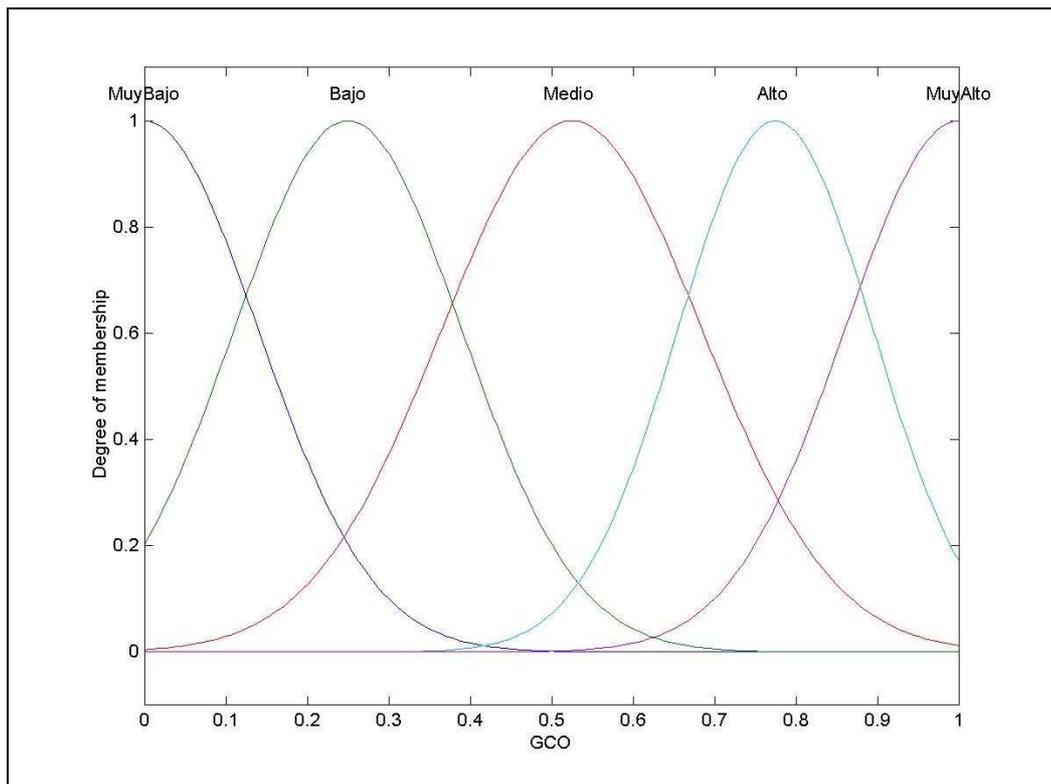


Figura 3-27.- Partición borrosa de la variables de entrada GCO.

En el caso de la variable de entrada EAO y en la variable de salida NCAO la partición borrosa será la misma puesto que la variable EAO es la realimentación de la salida borrosa del sistema. Las particiones borrosas consistirán en 3 conjuntos difusos gaussianos distribuidos en un universo de discurso normalizado con rango $[0,1]$, como se puede observar en las figuras 3-28 y 3-29.

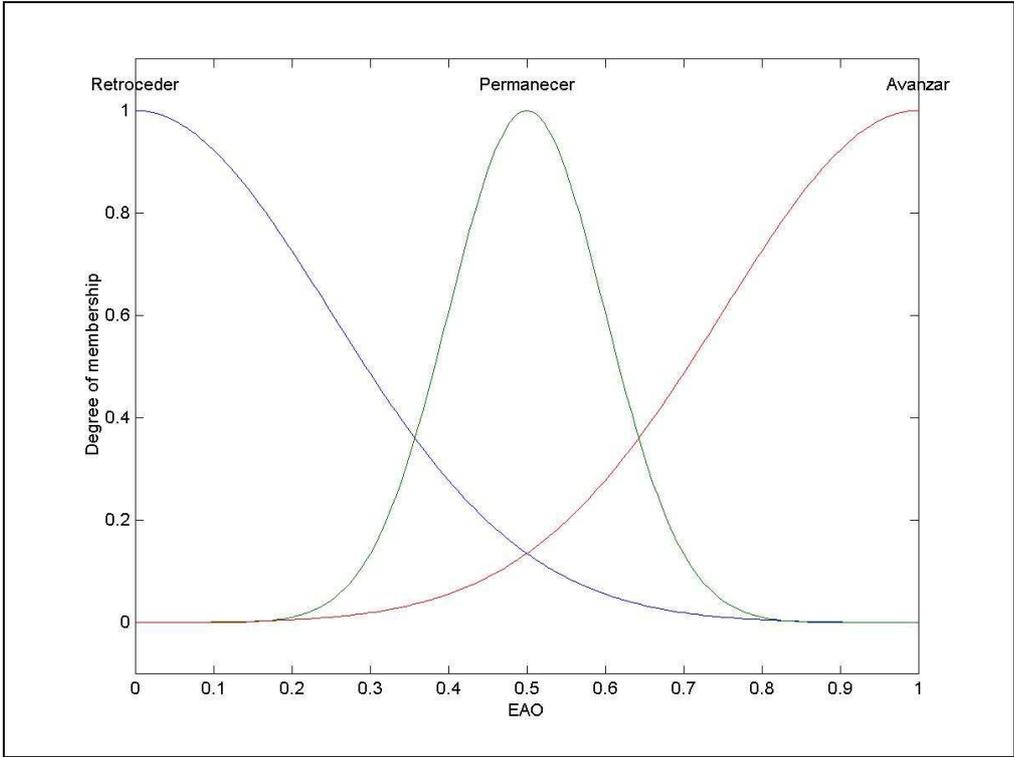


Figura 3-28.- Partición borrosa de la variable de entrada EAO.

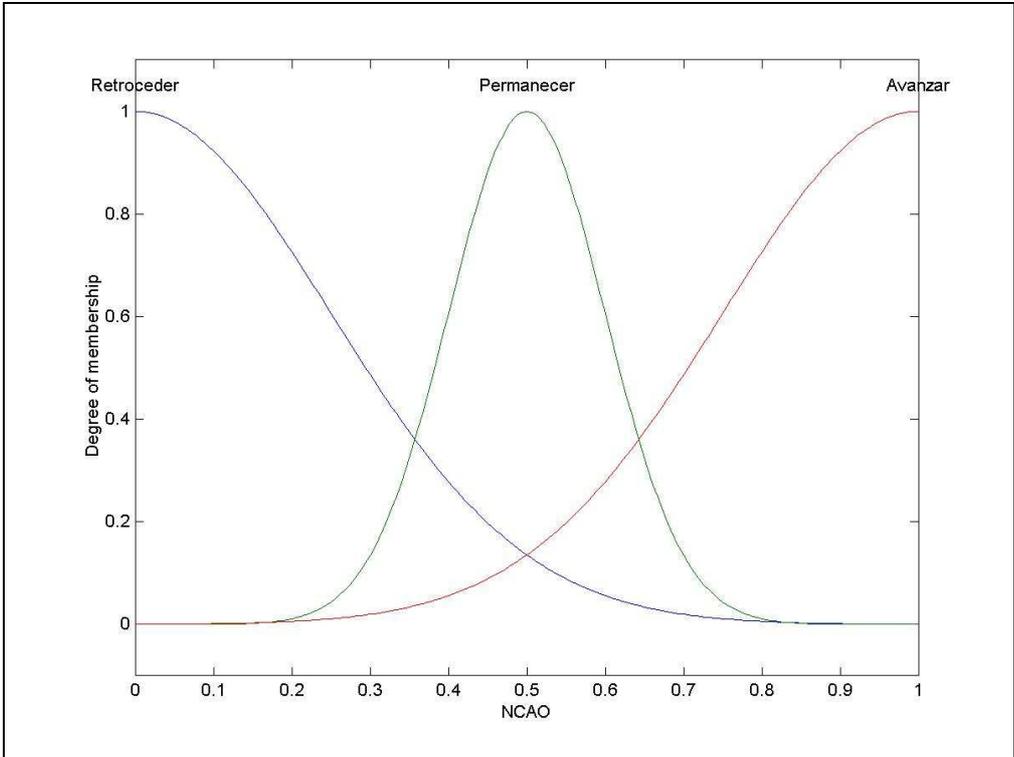


Figura 3-29.- Partición borrosa de la variable de salida NCAO.

6.6 Mecanismo de Borrosificación, Inferencia y Desborrosificación.

Para tener totalmente diseñado el planificador instruccional nos falta definir los procedimientos que nos permitan la borrosificación, la inferencia y la desborrosificación.

En la mayoría de los trabajos prácticos el proceso de borrosificación utilizado es el "singleton", donde la función de pertenencia se caracteriza porque tiene grado 1 para un solo valor de su universo (valor de entrada) y 0 en el resto. Esto es, la función impulso podría ser utilizada para representar una función de pertenencia de este tipo, figura 3-30.

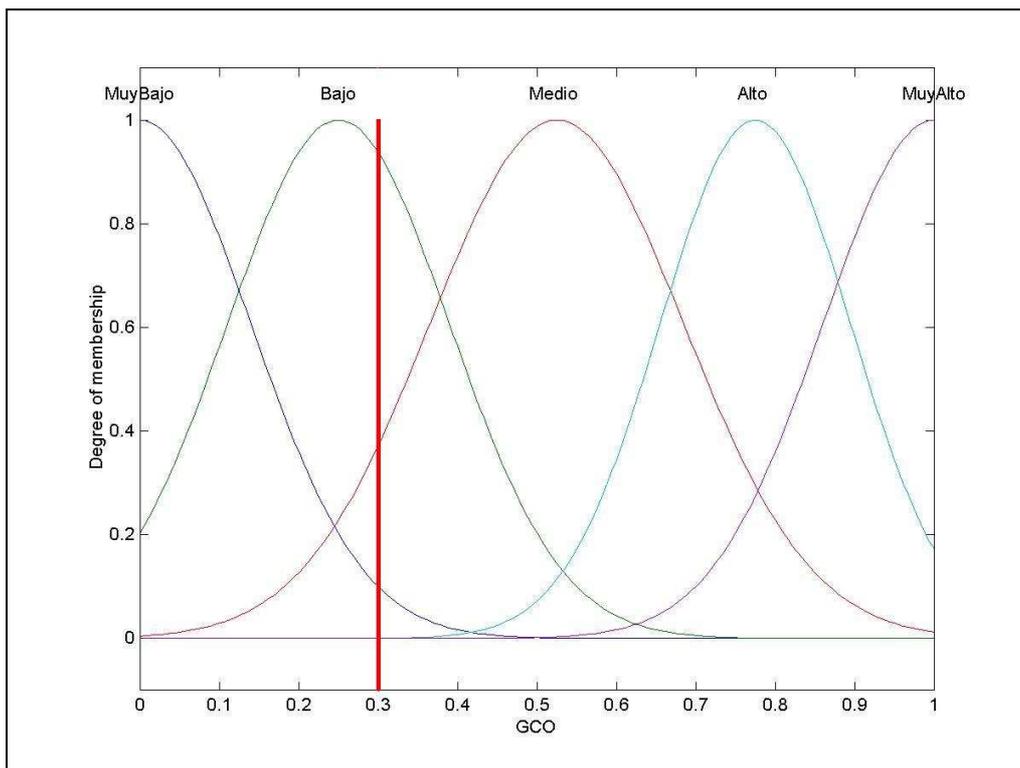


Figura 30.- Proceso de borrosificación de la variable de entrada GCO al planificador instruccional borroso.

Para definir el mecanismo de inferencia tenemos que determinar cómo realizar las operaciones básicas, consideramos el modelo de Mamdani y decidimos implementar la T-norma (intersección) como el mínimo y la S-norma (unión) como el máximo.

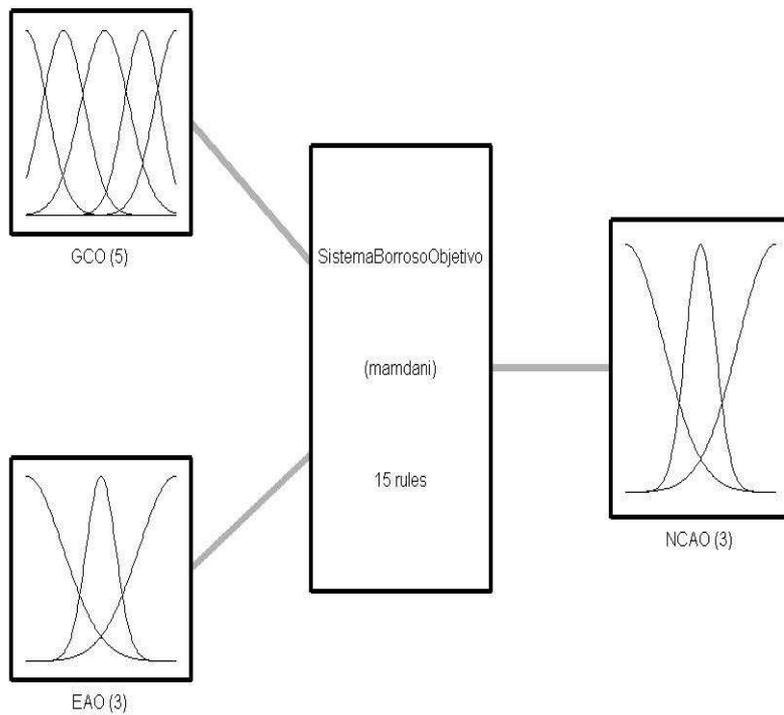
El modelo Mamdani es ampliamente aceptado a la hora de adquirir el conocimiento experto ya que permite describir los métodos de los expertos de un modo más intuitivo. En nuestro caso como ya hemos dicho la entrada GCO del Sistema Borroso para los objetivos es un número conciso (la otra entrada EAO es la realimentación de la salida) y nos interesa que la salida sea continua (graduada) para que pueda ser utilizada como entrada para el Sistema Borroso que calcula la fase de trabajo. Otra causa que nos llevó a decidimos por este modelo es que no tenemos una función exacta que defina el comportamiento del sistema sino que lo único que poseemos son algunos datos en forma de conocimiento heurístico obtenidos de los expertos.

Finalmente nos quedaría por definir el proceso de desborrosificación. Como ya se ha visto existen diversos métodos para realizar este proceso aunque nos hemos decidido por el método del centroide que es uno de los más ampliamente utilizados.

6.7 Simulación del Planificador Instruccional Borroso

Puesto que como hemos dicho anteriormente no disponemos de todos los datos necesarios para tener la función completa usaremos la simulación para ajustar los puntos de dicha función. Además los resultados de la simulación nos pueden servir para mejorar el diseño del planificador borroso y verificar que trabajará adecuadamente cuando sea implementado. Definimos el controlador diseñado anteriormente mediante la toolbox fuzzy de Matlab. El sistema borroso para cada uno de los objetivos de aprendizaje es el que se presenta en la figura 3-31. En cuanto a la superficie de salida ésta varía dependiendo de que tipo de alumno (puesto que las reglas son distintas). Estas superficies se pueden observar en las figuras 3-32, 3-33 y 3-34. Como se puede observar la primera se corresponde con un alumno con miedo al fracaso por lo que la superficie es mucho mayor para valores bajos de NCAO que para valores altos. La segunda representa un alumno motivado en el que las superficies para valores altos y bajos de NCAO están equilibradas con una zona central más amplia que en el caso anterior. Y por último

la figura 3-25 representa un alumno hiperactivo en la que la superficie mayor se corresponde con valores altos de la salida NCAO.



System SistemaBorrosoObjetivo: 2 inputs, 1 outputs, 15 rules

Figura 3-31.- Sistema Borroso para planificar el conjunto de actividades de cada objetivo de aprendizaje.

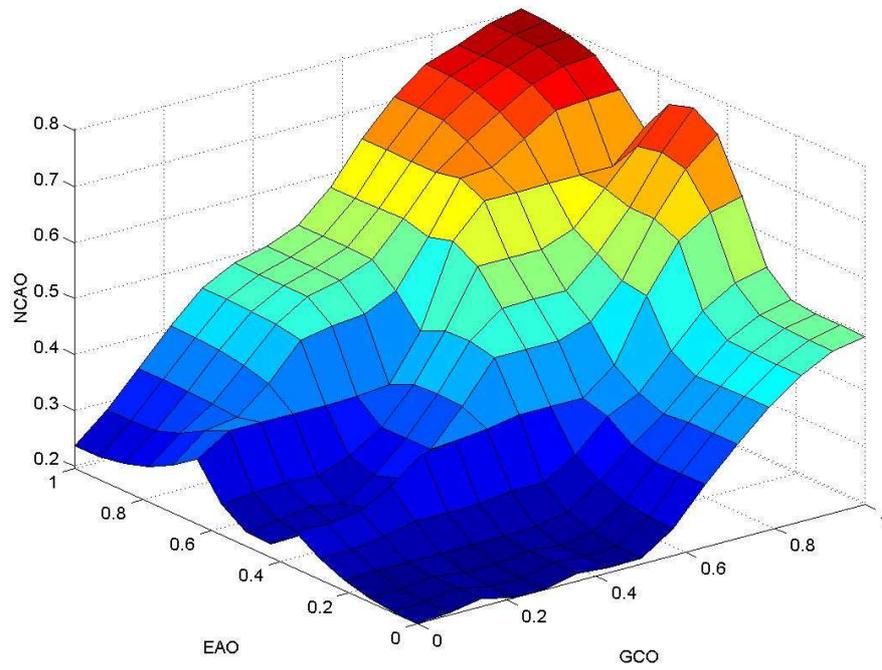


Figura 3-32.- Superficie de toma de decisiones para un alumno con miedo al fracaso.

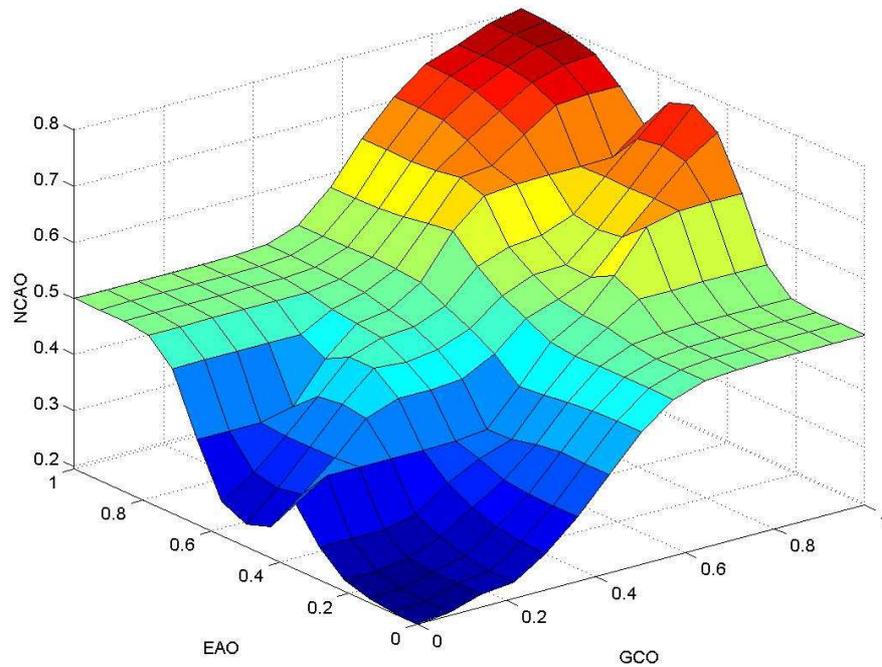


Figura 3-33.- Superficie de toma de decisiones para un alumno motivado.

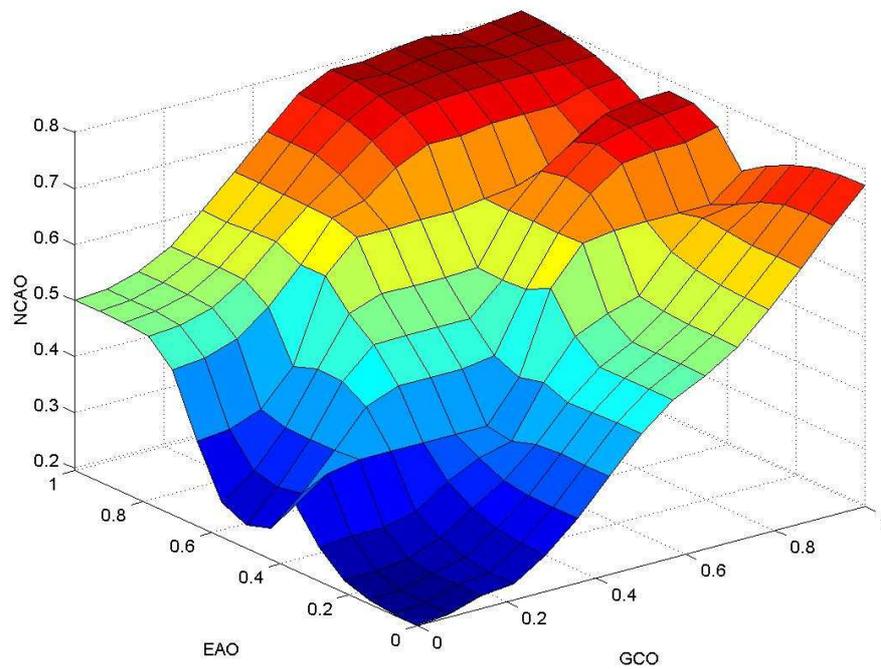


Figura 3-34.- Superficie de toma de decisiones para un alumno hiperactivo.

Con la herramienta fuzzy de Matlab podemos observar cómo se realiza el proceso de inferencia, aunque hay que tener en cuenta que ésta no nos permite que una de las variables de entrada sea borrosa, así que en la prueba de simulación tratamos las dos variables de entrada como si fueran concisas. En la figura 3-35 se puede ver un ejemplo de cómo sería el proceso de inferencia del sistema (variable de salida NCAO) para los valores de entrada $GCO = 0.5$ y $EAO = 0.5$.

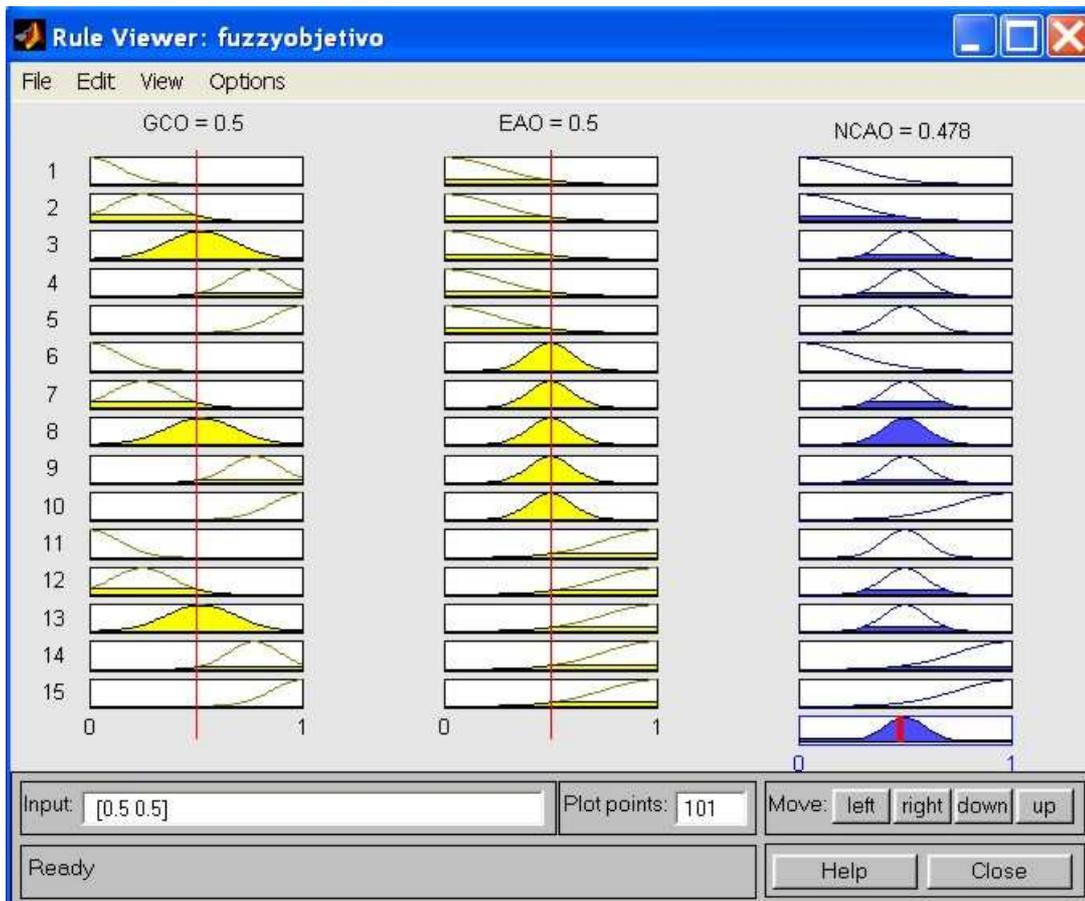


Figura 3-35.- Proceso de inferencia GCO(0.5) y EAO(0.5).

6.8 Implementación del Planificador Instruccional Borroso y resultados.

La implementación del planificador instruccional borroso ha sido realizada en Matlab utilizando, como en el caso de la simulación, la toolbox fuzzy, pero no pudiéndose utilizar la interfaz gráfica debido a la realimentación introducida por nuestro sistema.

Un primer problema era determinar el estado inicial de la variable EAO (realimentación del sistema) Para ello se probaron diferentes valores iniciales a ver cuál era el que mejor resultado proporcionaba. Los conjuntos iniciales con los que se experimentaron fueron:

- La entrada era igual a los propios conjuntos borrosos de retroceder, permanecer y avanzar (todos los valores del conjunto de entrada a 1).
- La entrada era el conjunto permanecer.
- La entrada era la equivalente al valor conciso 0.5 (todo a 0 y un 1 en el valor 50).

La prueba realizada consistió en introducir una secuencia variada de valores de entrada (GCO) como si fuera la primera vez que éste interactuaba con el sistema (por ejemplo 0.1, 0.2, 0.3, etc) y ver qué resultado obteníamos dependiendo del valor de inicialización de EAO. Esto se realizó para todos los tipos de alumno (con miedo al fracaso, hiperactivo y motivado).

Después de realizar las pruebas comprobamos que la mejor solución es la primera de las mencionadas puesto que para las otras se producían oscilaciones no deseadas en la salida (como en el caso de uso de conjuntos borrosos triangulares).

Para probar el funcionamiento adaptativo de nuestro planificador instruccional creamos varios casos de posibles interacciones del alumno con el Tutorial Inteligente y comparamos estos resultados con los obtenidos previamente con la simulación anterior (en la que los valores de EAO eran concisos). Pudimos comprobar que el hecho de utilizar como entrada el EAO (la realimentación del sistema) de forma borrosa y no como un número conciso después de la desborrosificación mejora de forma considerable el funcionamiento del planificador puesto que la cantidad de información histórica que mantenemos del estudiante es mayor.

Para comprobar la validez y generalidad de dicho planificador instruccional borroso se ha elegido como problema el diseño e implementación de un ITS para la ayuda a la enseñanza en los niños con discapacidades cognitivas (concretamente nos centramos en la enseñanza de la suma a alumnos con Síndrome de Down) que es una de las líneas de investigación del departamento de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadoras.

7 PLANIFICACIÓN MULTIAGENTE

Otra parte importante en la planificación es determinar el conjunto de actividades que se le van a mostrar al alumno, una vez que se ha determinado el nivel de dificultad de las mismas. Para determinar qué objetivo es el que se debe mostrar a continuación y qué actividad dentro de ese objetivo, hemos decidido usar un Sistema Multiagente (MAS).

Una primera aproximación del planificador instruccional fue realizado determinando aleatoriamente el objetivo y el tipo de actividad que se le mostraba al alumno una vez que el nivel de dificultad había sido obtenido usando lógica borrosa. Los experimentos realizados sobre ese prototipo mostraron que había objetivos que nunca trabajaba el alumno y otros, en cambio, se repetían mucho. Por esta razón se decidió usar un MAS que determina qué objetivo tiene que ser ejecutado, y qué tipo de actividad dentro de ese objetivo será mostrada al alumno.

En la tabla 3-13 se puede ver un ejemplo de como se elige el siguiente objetivo usando en a) método aleatorio y en b) el Sistema Multiagente. En el primer caso se puede observar que el objetivo 3 sólo aparece una vez, mientras que el objetivo 4 aparece cinco veces y además en ocasiones en iteraciones consecutivas. En cambio al usar el Sistema Multiagente existe una mayor variación a la hora de mostrar las actividades. Así se favorece la motivación del alumno frente al tutorial, ya que aunque lleve mal un objetivo y sea necesario incidir en éste (por ejemplo en el 4 en el caso de la tabla 3-13 el hecho de que se le muestren continuamente actividades de ese objetivo sin variación a otro objetivo (aunque se hayan generado suficientes actividades diferentes dentro de un objetivo) puede hacer que el tutorial se vuelva repetitivo y disminuya el interés del alumno por él.

a) Aleatorio				b) Sistema MultiAgente				
Siguiete Objetivo (t+1)				GCO del Objetivo Actual (t)	Siguiete Objetivo (t+1)			
1	2	3	4		1	2	3	4
X					X			
		X		Obj1 (70%)		X		
	X			Obj2 (90%)			X	
			X	Obj3 (60%)				X
	X			Obj4 (20%)	X			
			X	Obj1 (80%)				X
			X	Obj4 (20%)			X	
	X			Obj3 (50%)				X
X				Obj4 (40%)		X		
			X	Obj2 (70%)			X	
			X	Obj3 (60%)				X

Tabla 3-13.- Cálculo del Siguiete Objetivo usando a) Aleatoriedad o b) MAS

El enfoque de utilización de un Sistema Multiagente resulta de vital importancia para la creación de una plataforma inteligente de educación, pues provee de un entorno interactivo de gran capacidad de reacción. Esto se alcanza debido a la distribución de tareas a agentes que trabajan de forma concurrente. Es decir, que mientras el alumno realiza los ejercicios finales, los agentes pueden estar decidiendo la estrategia de enseñanza a seguir en el próximo objetivo lo cual permite disminuir el tiempo de espera en la interacción del usuario con el ITS.

La estructura de nuestro sistema multiagente es la que se puede observar en la figura 3-36.

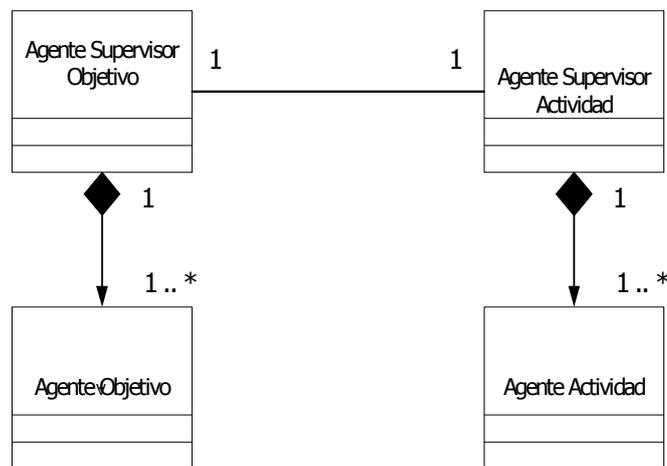


Figura 3-36. Estructura del MAS

Para determinar cuál es el siguiente objetivo que va a trabajar a continuación el alumno, se dispone de un Agente Supervisor Objetivo que se encarga de negociar con varios Agentes Objetivo. El número de éstos varía dependiendo de la fase en la que esté trabajando el alumno existiendo uno por cada objetivo de aprendizaje que se haya definido para dicha fase. Así en la fase 1 existen cuatro Agentes Objetivo, seis para la fase 2, etc. Una vez que se determine dicho objetivo, el siguiente paso es decidir a qué tipo de actividad dentro de dicho objetivo pertenecen los ejercicios que se le mostrarán al alumno. Para ello, se dispone un Agente Supervisor Actividad al que el Agente Supervisor Objetivo indica cuál es el objetivo que ha ganado. Para cada objetivo existen diferentes tipos de actividad por lo que, igual que en el caso de los Agentes Objetivo, se generará un número variable de Agentes Actividad en el sistema que negociarán entre ellos bajo las indicaciones del Agente Supervisor Actividad.

Para la negociación entre agentes se ha decidido usar los mecanismos basados en subastas porque para nuestro problema se obtiene con ellos un mejor rendimiento que usando los mecanismos de votación o regateo. Hay que tener en cuenta que el alumno está esperando el resultado del MAS (que se determine el siguiente objetivo y actividad) para seguir con el tutorial, por lo que necesitamos que el tiempo de respuesta sea lo más bajo posible [Aguilar 2005b].

7.1 Cálculo de la Función Objetivo

En primer lugar, el Agente Supervisor Objetivo se encarga de instanciar tantos Agentes Objetivo como objetivos de aprendizaje haya en la fase a tratar y de indicar a estos agentes el comienzo de la subasta. Cada uno de los Agentes Objetivo se encarga de obtener los datos del alumno, para el objetivo al que representan, y de calcular la siguiente función objetivo:

$$F_{obj_i} = 100 * (1 - (p_1 * NumActObj_i + p_2 * GCO_i + p_3 * Ant_i))$$

Donde:

- NumActObj_i es el número de actividades realizadas del objetivo i frente al número de actividades realizadas en total (para toda la fase).
- GCO_i es el grado de cumplimiento del objetivo i (porcentaje de aciertos del alumno para el objetivo i).
- Ant_i representa la antigüedad del objetivo i , es decir una medida que nos indique el orden en el que se han mostrado los objetivos.
- p_1, p_2, p_3 : coeficientes de pesos

Éste es el valor que los Agentes Objetivos usarán para pujar en la subasta. Como ya se ha indicado anteriormente usamos la subasta en sobre cerrado, así pues: cada Agente Objetivo va enviando el valor de su puja al Agente Supervisor Objetivo a medida que va calculando la Fobj_i ignorando los valores de los demás Agentes Objetivos. Una vez que han sido realizados todos los cálculos y el Agente Supervisor Objetivo dispone de todos los valores de las pujas elige como ganador de la subasta al Agente Objetivo que haya realizado la puja más alta (mayor valor de Fobj_i). El diagrama de tiempo se puede observar en la figura 3-37.

7.1.1 AJUSTE DE PESOS DE LA FUNCIÓN OBJETIVO

Se han realizado diversas pruebas variando los coeficientes de pesos de la Fobj_i . Inicialmente el parámetro de la antigüedad de los objetivos no había sido incluido dentro de la función. Pero entre los requisitos dados por los expertos para que el sistema funcione correctamente se encuentra que hay que trabajar no sólo los objetivos con peores resultados, sino también favorecer que las actividades sean variadas para aumentar la motivación y atención de los alumnos. En las primeras pruebas que se realizaron sin usar este parámetro el mismo objetivo podía salir varias veces seguidas debido a que los resultados obtenidos por el alumno en él eran muy bajos.

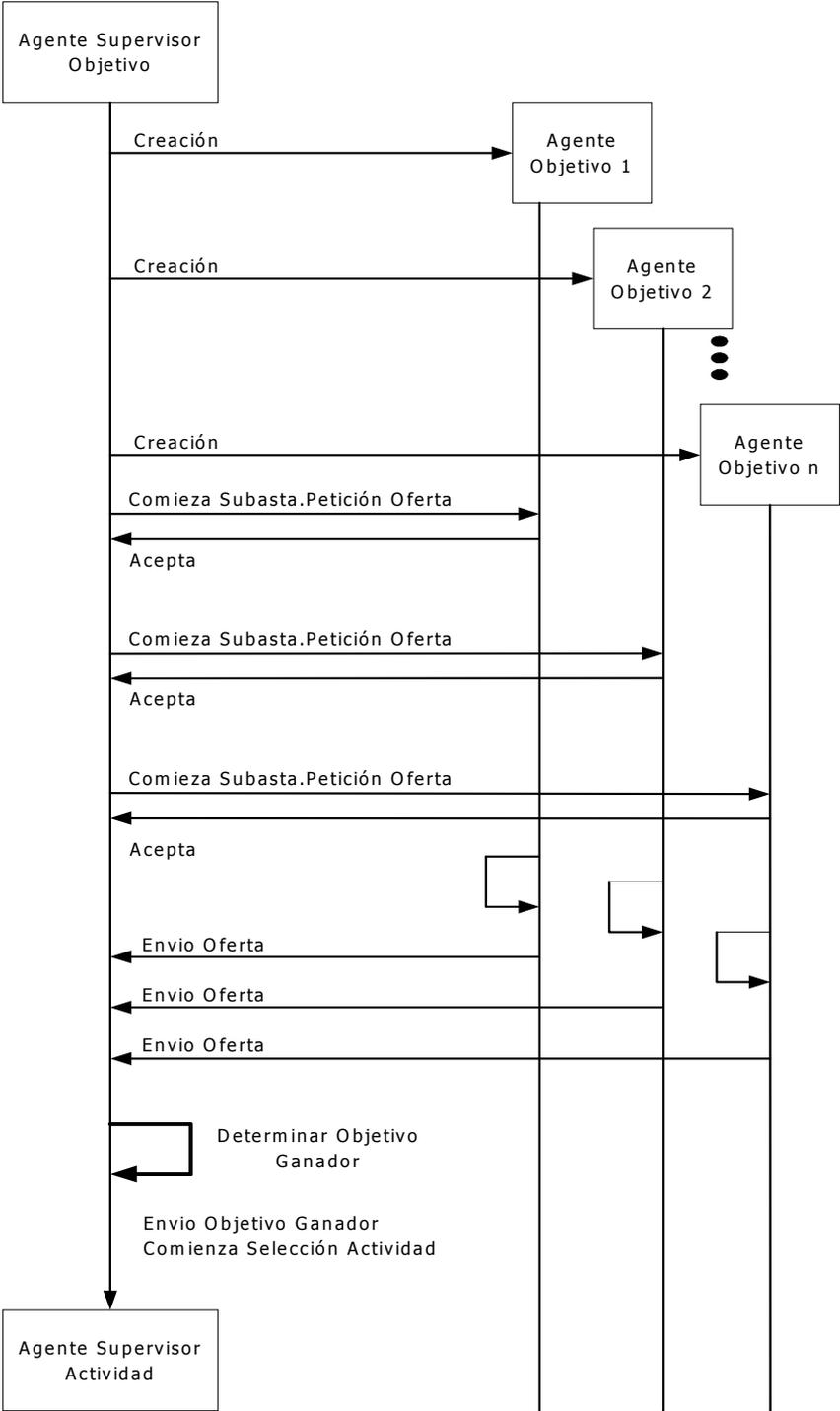


Figura 3-37. Diagrama de tiempo del Sistema MAS para el cálculo del siguiente objetivo.

En las pruebas inicialmente empezamos con un ajuste de los coeficientes de peso de $p_1 = 0,75$, $p_2 = 0,15$ y $p_3 = 0.1$ para los parámetros GCO_i , $NumAct_i$ y Ant_i respectivamente, con los resultados que se observan en la tabla 3-14. Como se puede observar en la tabla en la iteración 0, que es el estado inicial del que partimos, hay igualdad en la $Fobj_i$ para todos los objetivos y se elige el primero. La casilla sombreada en la tabla indica el objetivo ganador. En la siguiente iteración el alumno resuelve las actividades que se le muestran del Objetivo 1 y obtiene un porcentaje de aciertos del 70%, es decir un $GCO_1 = 0.7$ como se puede ver en la columna Objetivo Actual. Se vuelve a realizar la subasta, calculándose de nuevo las $Fobj_i$ siendo el ganador el Objetivo 2. Se le muestran al alumno las actividades obteniendo éste un $GCO_2 = 0.9$ y así sucesivamente. Al final de las 10 iteraciones se ha mostrado dos veces el Objetivo 2 y tres veces los Objetivos 1, 3 y 4.

Iter.	Objetivo Actual	Agente Objetivo 1	Agente Objetivo 2	Agente Objetivo 3	Agente Objetivo 4
0		Fobj = 92.5	Fobj = 92.5	Fobj = 92.5	Fobj = 92.5
1	Obj1 (GCO = 0.7)	Fobj = 7.5	Fobj = 92.5	Fobj = 92.5	Fobj = 92.5
2	Obj2 (GCO = 0.9)	Fobj = 47	Fobj = 45	Fobj = 92.5	Fobj = 92.5
3	Obj3 (GCO = 0.6)	Fobj = 61.16	Fobj = 54.83	Fobj = 57.5	Fobj = 92.5
4	Obj4 (GCO = 0.2)	Fobj = 68.25	Fobj = 62.75	Fobj = 64.75	Fobj = 63.75
5	Obj1 (GCO = 0.8)	Fobj = 57.49	Fobj = 61.5	Fobj = 70	Fobj = 71
6	Obj4 (GCO = 0.2)	Fobj = 61.33	Fobj = 65.66	Fobj = 73.5	Fobj = 59
7	Obj3 (GCO = 0.5)	Fobj = 59.42	Fobj = 72.92	Fobj = 59.57	Fobj = 67
8	Obj2 (GCO = 0.7)	Fobj = 60.32	Fobj = 55.07	Fobj = 62.32	Fobj = 68.07
9	Obj4 (GCO = 0.4)	Fobj = 65.7	Fobj = 63.94	Fobj = 68.05	Fobj = 62
10	Obj3 (GCO = 0.6)	Fobj = 68	Fobj = 66.5	Fobj = 60.5	Fobj = 62.5

Tabla 3-14. Resultados del MAS Obj. para los coeficientes de peso 0.75, 0.15, 0.1

En vista de estos resultados se intenta darle un poco más de importancia al porcentaje de aciertos de los alumnos para incrementar la frecuencia de los objetivos en los que el alumno tiene peor GCO_i . Es decir en el ejemplo anterior hubieran ganado más veces la subasta los Objetivos 3 y 4 que son en los que peores resultados tiene el alumno. Se ajustan los pesos a $p_1 = 0.75$, $p_2 = 0.2$ y $p_3 = 0.05$. Se puede comprobar en la tabla 3-15 que con este ajuste gana dos veces seguidas la subasta el objetivo 4 (iteraciones 3 y 4). Si se quiere tener en cuenta que no aparezca dos veces seguidas el mismo objetivo para que el alumno no se

aburra se llega a la conclusión de que el coeficiente de peso p_3 tiene que ser mayor de 0.05.

Iter.	Objetivo Actual	Agente Objetivo 1	Agente Objetivo 2	Agente Objetivo 3	Agente Objetivo 4
0		Fobj = 90	Fobj = 90	Fobj = 90	Fobj = 90
1	Obj1 (GCO = 0.7)	Fobj = 10	Fobj = 90	Fobj = 90	Fobj = 90
2	Obj2 (GCO = 0.9)	Fobj = 46	Fobj = 47.5	Fobj = 90	Fobj = 90
3	Obj3 (GCO = 0.6)	Fobj = 59.33	Fobj = 53.66	Fobj = 60	Fobj = 90
4	Obj4 (GCO = 0.2)	Fobj = 66	Fobj = 60.75	Fobj = 65.5	Fobj = 66.25

Tabla 3-15. Resultados del MAS Obj. para los coeficientes de peso 0.75, 0.2, 0.05

En la siguiente prueba se usan los pesos $p_1 = 0.7$, $p_2 = 0.2$ y $p_3 = 0.1$ (tabla 3-16) pero se obtienen los mismos resultados que en la primera prueba (con $p_1 = 0.75$, $p_2 = 0.15$ y $p_3 = 0.1$). Si se quiere que en la 8ª iteración vuelva a salir el objetivo 4 que es el que peor GCO_i tiene, pero cumpliendo los requisitos de variación de actividades, hay que aumentar el coeficiente de peso p_2 , pero manteniendo $p_3 = 0.1$.

Iter.	Objetivo Actual	Agente Objetivo 1	Agente Objetivo 2	Agente Objetivo 3	Agente Objetivo 4
0		Fobj = 90	Fobj = 90	Fobj = 90	Fobj = 90
1	Obj1 (GCO = 0.7)	Fobj = 10	Fobj = 90	Fobj = 90	Fobj = 90
2	Obj2 (GCO = 0.9)	Fobj = 46	Fobj = 45	Fobj = 90	Fobj = 90
3	Obj3 (GCO = 0.6)	Fobj= 59.33	Fobj = 52	Fobj = 56.66	Fobj = 90
4	Obj4 (GCO = 0.2)	Fobj = 66	Fobj = 59.5	Fobj = 63	Fobj = 62.5
5	Obj1 (GCO = 0.8)	Fobj = 48	Fobj = 64	Fobj = 68	Fobj = 74
6	Obj4 (GCO = 0.2)	Fobj= 52.33	Fobj = 67	Fobj = 71.33	Fobj = 62.66
7	Obj3 (GCO = 0.5)	Fobj= 56.85	Fobj = 69.14	Fobj = 58	Fobj = 67.42

Tabla 3-16. Resultados del MAS Obj. para los coeficientes de peso 0.7, 0.2, 0.1

Así pues, probamos que unos coeficientes de peso $p_1 = 0.6$, $p_2 = 0.3$ y $p_3 = 0.1$. Como se puede observar en la tabla 3-17 ahora se muestran dos veces los Objetivos 1 y 2 que son en los que mejores porcentajes de acierto tiene el alumno, tres veces el Objetivo 3 en el que los resultados son intermedios y cuatro veces el Objetivo 4 que es el de peor GCO_i. Además de las cuatro veces que se muestra

dicho Objetivo ninguna de ellas es de forma consecutiva. Puesto que obtenemos los resultados deseados con estos coeficientes de peso son éstos los que finalmente elegimos para el MAS que determina el siguiente objetivo.

Iter.	Objetivo Actual	Agente Objetivo 1	Agente Objetivo 2	Agente Objetivo 3	Agente Objetivo 4
0		Fobj = 85	Fobj = 85	Fobj = 85	Fobj = 85
1	Obj1 (GCO = 0.7)	Fobj = 15	Fobj = 85	Fobj = 85	Fobj = 85
2	Obj2 (GCO = 0.9)	Fobj = 50	Fobj = 45	Fobj = 85	Fobj = 85
3	Obj3 (GCO = 0.6)	Fobj = 55.66	Fobj = 46.33	Fobj = 55	Fobj = 85
4	Obj4 (GCO = 0.2)	Fobj = 61.5	Fobj = 53	Fobj = 59	Fobj = 60
5	Obj1 (GCO = 0.8)	Fobj = 45	Fobj = 57	Fobj = 64	Fobj = 74
6	Obj4 (GCO = 0.2)	Fobj = 47.66	Fobj = 59.66	Fobj = 67	Fobj = 64
7	Obj3 (GCO = 0.5)	Fobj = 51.71	Fobj = 61.57	Fobj = 54.85	Fobj = 68.28
8	Obj4 (GCO = 0.4)	Fobj = 54.75	Fobj = 63	Fobj = 61.25	Fobj = 61.5
9	Obj2 (GCO = 0.7)	Fobj = 55.4	Fobj = 48	Fobj = 62.22	Fobj = 56.61
10	Obj3 (GCO = 0.6)	Fobj = 57.66	Fobj = 56.66	Fobj = 55	Fobj = 60

Tabla 3-17. Resultados del MAS Obj. para los coeficientes de peso 0.6, 0.3 y 0.1

7.2 Cálculo de la Función Actividad

Una vez que el objetivo ganador ha sido determinado procedemos a determinar que tipo de actividad dentro de ese objetivo va a ser la siguiente que se le va a mostrar al alumno. Para ello, el Agente Supervisor Objetivo le comunica al Agente Supervisor Actividad este resultado y le cede el control. Este Agente Supervisor Actividad se encarga de instanciar tantos Agentes Actividad como tipos de actividades compongan el objetivo ganador y de indicarles a éstos que da comienzo la subasta. Igual que en el caso de los Agentes Objetivos, cada uno de los Agentes Actividad se encarga de obtener los datos del alumno de la base de datos y de calcular la siguiente función objetivo:

$$\text{Fact}_i = p1 * \text{NumAct}_i + p2 * \text{GCOAct}_i$$

Donde:

- NumAct_{*i*} es el número de actividades realizadas del tipo *i* frente al número de actividades totales realizadas para ese objetivo.
- GCOAct_{*i*} es el grado de cumplimiento de la actividad *i* (porcentaje de aciertos del alumno para el tipo de actividad *i*).
- p₁, p₂: coeficientes de pesos

El diagrama de funcionamiento para el MAS que determina la siguiente actividad es igual al del que determina el siguiente objetivo. Su esquema es por tanto igual al que se pudo observar en la figura 26 sustituyendo los Agentes Objetivo por Agentes Actividad y eliminando la última interacción de envío al Agente Supervisor Actividad. Esto es debido a que el MAS finaliza al determinar el tipo de la actividad ganadora de la subasta mientras que en el caso anterior una vez determinado el objetivo ganador se continuaba para determinar el tipo de la actividad ganadora.

Como ya hemos comentado cada objetivo se compone de varios tipos de actividades. Algunas de las actividades dentro del objetivo son más prioritarias que otras. Por esta razón el ajuste de los coeficientes de peso depende de si la actividad es prioritaria o no. Inicialmente se dispuso que las actividades prioritarias tuvieran una proporción de importancia del 80% frente a las no prioritarias que tenían por tanto un 20%. Pero al realizar diferentes simulaciones constatamos que, por ejemplo, en el Objetivo 4 que consta de tres actividades prioritarias (actividades 1, 2 y 3) y una no prioritaria (actividad 4), esta última nunca ganaba la subasta aún disminuyendo esa proporción de importancia (tablas 3-19, 3-20 y 3-21). Por esta razón decidimos multiplicar la Fact_{*i*} por una función de prioridad basándonos en las técnicas de Sistemas Operativos (S.O.) para la planificación de procesos [Stallings 2002, Silberschatz 2005].

Iter.	Actividad Actual	Agente Actividad 1	Agente Actividad 2	Agente Actividad 3	Agente Actividad 4
0		Fact = 80	Fact = 80	Fact = 80	Fact = 20
1	Act1 (GCO = 0.7)	Fact = 2.4	Fact = 80	Fact = 80	Fact = 20
2	Act2 (GCO = 0.7)	Fact = 38.4	Fact = 38.4	Fact = 80	Fact = 20
3	Act3 (GCO = 0.7)	Fact = 50.4	Fact = 50.4	Fact = 50.4	Fact = 20
4	Act1 (GCO = 0.7)	Fact = 38.4	Fact = 56.4	Fact = 56.4	Fact = 20
5	Act2 (GCO = 0.7)	Fact = 45.6	Fact = 45.6	Fact = 60	Fact = 20
6	Act3 (GCO = 0.7)	Fact = 50.4	Fact = 50.4	Fact = 50.4	Fact = 20

Tabla 3-19. Resultados del MAS Actividad para el Objetivo 4 con coeficientes de peso 0.9, 0.1 y proporción de importancia 80%-20%

Iter.	Actividad Actual	Agente Actividad 1	Agente Actividad 2	Agente Actividad 3	Agente Actividad 4
0		Fact = 70	Fact = 70	Fact = 70	Fact = 30
1	Act1 (GCO = 0.7)	Fact = 2.1	Fact = 70	Fact = 70	Fact = 30
2	Act2 (GCO = 0.7)	Fact = 33.6	Fact = 33.6	Fact = 70	Fact = 30
3	Act3 (GCO = 0.7)	Fact = 44.1	Fact = 44.1	Fact = 44.1	Fact = 30
4	Act1 (GCO = 0.7)	Fact = 33.6	Fact = 49.35	Fact = 49.35	Fact = 30
5	Act2 (GCO = 0.7)	Fact = 39.9	Fact = 39.9	Fact = 52.2	Fact = 30
6	Act3 (GCO = 0.7)	Fact = 44.1	Fact = 44.1	Fact = 44.1	Fact = 30

Tabla 3-20. Resultados del MAS Actividad para el Objetivo 4 con coeficientes de peso 0.9, 0.1 y proporción de importancia 70%-30%

Iter.	Actividad Actual	Agente Actividad 1	Agente Actividad 2	Agente Actividad 3	Agente Actividad 4
0		Fact = 65	Fact = 65	Fact = 65	Fact = 35
1	Act1 (GCO = 0.7)	Fact = 1.95	Fact = 65	Fact = 65	Fact = 35
2	Act2 (GCO = 0.7)	Fact = 31.2	Fact = 31.2	Fact = 65	Fact = 35
3	Act3 (GCO = 0.7)	Fact = 40.95	Fact = 40.95	Fact = 40.95	Fact = 35
4	Act1 (GCO = 0.7)	Fact = 31.2	Fact = 45.82	Fact = 45.82	Fact = 35
5	Act2 (GCO = 0.7)	Fact = 37.05	Fact = 37.05	Fact = 54.6	Fact = 35
6	Act3 (GCO = 0.7)	Fact = 40.95	Fact = 40.95	Fact = 40.95	Fact = 35

Tabla 3-21. Resultados del MAS Actividad para el Objetivo 4 con coeficientes de peso 0.9, 0.1 y proporción de importancia 65%-35%

En el caso de los procesos en S.O. la unidad básica son los milisegundos. Como en nuestro caso la unidad básica que nosotros consideramos son las actividades la fórmula que vamos a usar es:

$$F_{prio_i} = (100 * QBAct_i) / NumAct_i$$

Para calcular el $QBAct_i$ partimos del valor 100 (por similitud con los S.O. donde se suele coger 100 ms como estándar) y calculamos el valor correspondiente a cada actividad considerando que las actividades prioritarias pesan 3 veces más que las no prioritarias. Un ejemplo de valores de $QBAct_i$ para diferentes casos se muestra en la tabla 3-22:

Ejemplo	Número de actividades prioritarias	Número de Actividades no prioritarias	$QBAct_i$
1	1	1	$100 = 3y + y$ $QBAct_1 = 75$ $QBAct_2 = 25$
2	3	1	$100 = 3y + 3y + 3y + y$ $QBAct_1 = QBAct_2 = QBAct_3 = 30$ $QBAct_4 = 10$

Tabla 3-22. Ejemplo de cálculo de $QBAct_i$

Se han realizado una serie de pruebas tomando el resultado de la $Fact_i$ y multiplicándola por F_{prio_i} para las actividades del Objetivo 4 con igualdad de $GCOAct_i$ para las cuatro actividades (las tres primeras prioritarias y la última no prioritaria). Los resultados de estas pruebas se pueden ver en la tabla 3-23. Como se puede observar ahora, la Actividad 4 que es la menos prioritaria ya tiene opciones de competir y puede ganar la subasta ya que ha sido la ganadora en dos ocasiones en las iteraciones 6 y 11.

Iter.	Actividad Actual	Agente Actividad 1	Agente Actividad 2	Agente Actividad 3	Agente Actividad 4
0		Fact = 28.5	Fact = 28.5	Fact = 28.5	Fact = 9.5
1	Act1 (GCO = 0.5)	Fact = 0.75	Fact = 28.5	Fact = 28.5	Fact = 9.5
2	Act2 (GCO = 0.5)	Fact = 7.50	Fact = 7.50	Fact = 28.5	Fact = 9.5
3	Act3 (GCO = 0.5)	Fact = 9.75	Fact = 9.75	Fact = 9.75	Fact = 9.5
4	Act1 (GCO = 0.5)	Fact = 3.75	Fact = 10.87	Fact = 10.87	Fact = 9.5
5	Act2 (GCO = 0.5)	Fact = 4.25	Fact = 4.25	Fact = 11.55	Fact = 9.5
6	Act3 (GCO = 0.5)	Fact = 4.87	Fact = 4.87	Fact = 4.87	Fact = 9.5
7	Act4 (GCO = 0.5)	Fact = 5.10	Fact = 5.10	Fact = 5.10	Fact = 4.10
8	Act1 (GCO = 0.5)	Fact = 3.06	Fact = 5.43	Fact = 5.43	Fact = 4.18
9	Act2 (GCO = 0.5)	Fact = 3.25	Fact = 3.25	Fact = 5.62	Fact = 4.25
10	Act3 (GCO = 0.5)	Fact = 3.4	Fact = 3.4	Fact = 3.4	Fact = 4.3
11	Act4 (GCO = 0.5)	Fact = 3.5	Fact = 3.5	Fact = 3.5	Fact = 1.96
12	Act1 (GCO = 0.5)	Fact = 2.4	Fact = 3.62	Fact = 3.62	Fact = 2
13	Act2 (GCO = 0.5)	Fact = 2.52	Fact = 2.52	Fact = 3.71	Fact = 2.02
14	Act3 (GCO = 0.5)	Fact = 2.59	Fact = 2.59	Fact = 2.59	Fact = 2.05

Tabla 3-23. Resultados del MAS Actividad para el Obj4 con QB.

Una vez que todas las actividades tienen posibilidades de competir gracias a la inclusión de la F_{prio_i} , hay que ajustar los coeficientes de peso de la $Fact_i$. Al igual que pasaba al calcular los coeficientes de peso de la F_{obj_i} , hay que favorecer la alternancia de actividades siempre que sea posible por cuestiones de motivación pero debe de reflejarse de forma conveniente el hecho de que si el alumno tiene resultados negativos en una actividad se le deben de mostrar más actividades de ese tipo para que pueda practicar y mejorar el nivel en ese objetivo. Después de varias pruebas se comprobó que los mejores resultados se obtenían usando los valores $p_1 = 0,7$ y $p_2 = 0.3$. Para ello se realizaron varias simulaciones con la intención de comprobar que los resultados que se obtienen con variaciones del GCO_{Act_i} (incluyendo los casos extremos) son los correctos. Las tablas 3-24, 3-25, 3-26, 3-27 y 3-28 muestran los resultados obtenidos.

En la tabla 3-24 se realiza la simulación para el Objetivo 1 que consta de dos actividades, la primera prioritaria y la segunda no, usando $GCO = 0.5$ para las dos. Se observa que en 10 iteraciones gana 6 veces la actividad prioritaria (actividad 1) y 4 la no prioritaria (actividad 2). Hay alternancia de actividades siempre que es

posible, aunque en este caso como sólo hay dos actividades y además una de ellas es prioritaria, es casi inevitable que no aparezca a veces de forma consecutiva.

La tabla 3-25 muestra las pruebas para el Objetivo 1 pero siendo peores los coeficientes de acierto del alumno para la actividad prioritaria que para la que no lo es ($GCO_{Act_1} = 0.3$ y $GCO_{Act_2} = 0.7$). Por este motivo después de las 10 iteraciones gana 7 veces la Actividad 1 frente a las 3 de la Actividad 2.

En la tabla 3-26 ocurre lo contrario que en la anterior. Ahora es la actividad prioritaria del Objetivo 1 la que tiene buen coeficiente de acierto frente a la no prioritaria ($GCO_{Act_1} = 0.7$ y $GCO_{Act_2} = 0.3$). Los resultados que se obtienen ahora es que gana 6 veces la Actividad 1 y 4 la Actividad 2. Son iguales que en la tabla 24, pero ahora la alternancia es mayor, ya que en la iteración 8 gana la Actividad 2 y antes se repetía dos veces la Actividad 1 (iteraciones 7 y 8).

Las tablas 3-27 y 3-28 muestran los resultados extremos de que el porcentaje de aciertos para las actividades sea el 100% o el 0%. En la primera el porcentaje para la Actividad 1 es del 100% y para la Actividad 2 del 0% por lo que se muestran con más frecuencia actividades del tipo 1 que del tipo 2. En la segunda tabla pasa exactamente lo contrario.

Iteración	Actividad Actual	Agente Actividad 1	Agente Actividad 2
0		Fact = 63.75	Fact = 21.25
1	Act1 (GCO = 0.5)	Fact = 5.625	Fact = 21.25
2	Act2 (GCO = 0.5)	Fact = 18.75	Fact = 6.25
3	Act1 (GCO = 0.5)	Fact = 7.18	Fact = 7.70
4	Act2 (GCO = 0.5)	Fact = 9.375	Fact = 3.125
5	Act1 (GCO = 0.5)	Fact = 5.375	Fact = 3.56
6	Act1 (GCO = 0.5)	Fact = 3.59	Fact = 3.85
7	Act2 (GCO = 0.5)	Fact = 4.21	Fact = 2.29
8	Act1 (GCO = 0.5)	Fact = 3.09	Fact = 2.44
9	Act1 (GCO = 0.5)	Fact = 2.39	Fact = 2.57

Tabla 3-24. Resultados del MAS Act. para el Obj1 con $GCO_{Act_1} = GCO_{Act_2} = 0.5$

Iteración	Actividad Actual	Agente Actividad 1	Agente Actividad 2
0		Fact = 63.75	Fact = 21.25
1	Act1 (GCO = 0.3)	Fact = 5.625	Fact = 21.25
2	Act2 (GCO = 0.7)	Fact = 21	Fact = 6.25
3	Act1 (GCO = 0.3)	Fact = 8.31	Fact = 6.95
4	Act1 (GCO = 0.3)	Fact = 4.81	Fact = 7.68
5	Act2 (GCO = 0.7)	Fact = 6.125	Fact = 3.18
6	Act1 (GCO = 0.3)	Fact = 4.15	Fact = 3.48
7	Act1 (GCO = 0.3)	Fact = 3.075	Fact = 3.68
8	Act2 (GCO = 0.7)	Fact = 3.54	Fact = 2.19
9	Act1 (GCO = 0.3)	Fact = 2.77	Fact = 2.32

Tabla 3-25. Resultados del MAS Act. para el Obj1 con $GCO_{Act1}=0.3$ y $GCO_{Act2}=0.7$

Iteración	Actividad Actual	Agente Actividad 1	Agente Actividad 2
0		Fact = 63.75	Fact = 21.25
1	Act1 (GCO = 0.7)	Fact = 5.625	Fact = 21.25
2	Act2 (GCO = 0.3)	Fact = 16.5	Fact = 6.25
3	Act1 (GCO = 0.7)	Fact = 6.06	Fact = 8.45
4	Act2 (GCO = 0.3)	Fact = 8.25	Fact = 3.5
5	Act1 (GCO = 0.7)	Fact = 4.625	Fact = 3.93
6	Act1 (GCO = 0.7)	Fact = 3.03	Fact = 4.22
7	Act2 (GCO = 0.3)	Fact = 3.65	Fact = 2.54
8	Act1 (GCO = 0.7)	Fact = 2.64	Fact = 2.69
9	Act2 (GCO = 0.7)	Fact = 3.00	Fact = 1.87

Tabla 3-26. Resultados del MAS Act. para el Obj1 con $GCO_{Act1}=0.7$ y $GCO_{Act2}=0.3$

Iteración	Actividad Actual	Agente Actividad 1	Agente Actividad 2
0		Fact = 63.75	Fact = 21.25
1	Act1 (GCO = 1)	Fact = 5.625	Fact = 21.25
2	Act2 (GCO = 0)	Fact = 13.125	Fact = 6.25
3	Act1 (GCO = 1)	Fact = 4.375	Fact = 9.58
4	Act2 (GCO = 0)	Fact = 6.56	Fact = 4.06
5	Act1 (GCO = 1)	Fact = 3.5	Fact = 4.5
6	Act1 (GCO = 1)	Fact = 4.375	Fact = 2.70
7	Act2 (GCO = 0)	Fact = 2.81	Fact = 2.91
8	Act1 (GCO = 1)	Fact = 3.28	Fact = 2.03
9	Act2 (GCO = 1)	Fact = 2.33	Fact = 2.15

Tabla 3-27. Resultados del MAS Act. para el Obj1 con $GCO_{Act1}=1$ y $GCO_{Act2}=0$

Iteración	Actividad Actual	Agente Actividad 1	Agente Actividad 2
0		Fact = 63.75	Fact = 21.25
1	Act1 (GCO = 0)	Fact = 5.625	Fact = 21.25
2	Act2 (GCO = 1)	Fact = 24.375	Fact = 6.25
3	Act1 (GCO = 0)	Fact = 10	Fact = 5.83
4	Act1 (GCO = 0)	Fact = 5.93	Fact = 6.56
5	Act2 (GCO = 1)	Fact = 7.25	Fact = 2.625
6	Act1 (GCO = 0)	Fact = 5	Fact = 2.91
7	Act1 (GCO = 0)	Fact = 3.75	Fact = 3.125
8	Act1 (GCO = 0)	Fact = 2.96	Fact = 3.28
9	Act2 (GCO = 1)	Fact = 3.33	Fact = 1.94

Tabla 3-28. Resultados del MAS Act. para el Obj1 con $GCO_{Act1}=0$ y $GCO_{Act2}=1$

Por último se planteó el problema de resetear las prioridades. Se decidió usar epoch (épocas) de 100 actividades ya que éste fue el valor inicial elegido a la hora de inicializar el $QBAct_i$. Otras técnicas que se barajaron fueron resetear las prioridades cuando alguna de las prioridades llegara a 1 o usar épocas basadas en el número de sesiones realizadas o en la duración de éstas. Como se quiere que el tiempo de respuesta sea el menor posible nos decantamos por la opción mencionada para evitar la complejidad y la duración de procesamiento.

En la ejecución del ITS cada vez que se elige el Objetivo y el Tipo de Actividad se le muestran al alumno dos actividades que tiene que realizar. Por tanto, el cambio de época se produce cuando se llevan simuladas 50 iteraciones. En la figura 3-27 se muestra cuántas actividades han sido mostradas para los Objetivos 1 (una actividad prioritaria y una no prioritaria) y 4 (tres prioritarias y una no prioritaria) usando $GCOAct_i = 0.5$.

Al llevar a cabo el cambio de época existen varias opciones para definir las nuevas prioridades:

1. Beneficiar en la nueva época a las actividades que no consumieron su $QBAct_i$ en la época anterior y perjudicar a las que lo sobrepasaron.
2. Sólo beneficiar a las actividades que no consumieron su $QBAct_i$ en la época anterior, pero no perjudicar a las que lo sobrepasaron.

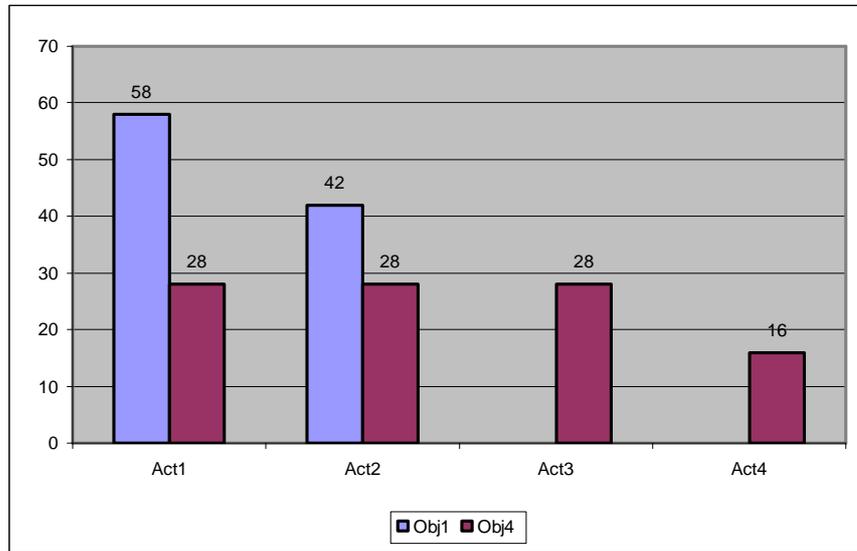


Figura 3-27. Número de Actividades mostradas para los Objetivos 1 y 4 en el cambio de época.

Los resultados obtenidos al simular la nueva época usando estas dos versiones son los mismos tanto para el Objetivo 1 como para el Objetivo 4 como se puede ver en la figura 3-28. El resultado en cuanto al número de actividades es el mismo, pero en cuanto a la distribución de actividades (aún siendo bastante similares), el método de sólo beneficiar a las actividades que no consumieron su $QBAct_i$ y no perjudicar a las que si lo hicieron funciona un poco mejor. En función de estos resultados decidimos adoptar finalmente este método. Las tablas donde se muestran con detalles todas estas simulaciones se pueden encontrar en Apéndice I.

Finalmente la fórmula que se ha empleado para determinar la prioridad después del cambio de época beneficiando a las actividades que no consumieron su $QBAct_i$ es la siguiente:

$$F_{prio_i} = 50 * ((2 * QBAct_i + (QBAct_{anterior} - NumAct_{anterior})) / NumAct_i)$$

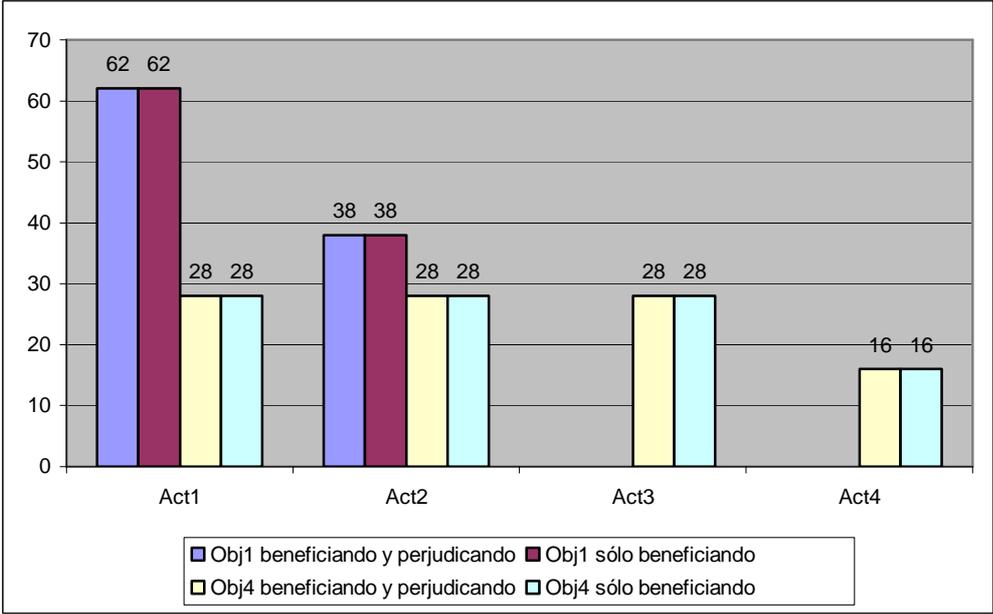


Figura 3-28. Número de Actividades mostradas para los Objetivos 1 y 4 en el cambio de época beneficiando y perjudicando o sólo beneficiando.

CAPÍTULO IV

VERIFICACIÓN Y VALIDACIÓN

1 INTRODUCCIÓN

La verificación y validación de los sistemas inteligentes es una tarea muy compleja. Sin embargo resulta esencial para que el Sistema Inteligente pueda ser utilizado. La verificación comprueba si los resultados concuerdan con la especificación del modelo, mientras que la validación comprueba que el sistema funciona correctamente y los resultados son los que se esperaban [Sargent 1998]. Algunas de las dificultades que presentan son debidas a que un modelo válido para un proceso no tiene por qué serlo para otro y además en algunos casos es difícil realizar comparaciones sobre el mundo real, ya que diferentes personas pueden tener diferentes visiones o interpretaciones de un mismo sistema. Hay que tener en cuenta que en el caso de que el sistema no actúe de forma aislada o independiente sino que se encuentre integrado en un sistema mayor es necesaria la verificación y validación también con los otros elementos que formen parte de ese sistema. Además, en ocasiones los datos con los que se trabaja no son precisos y aún cuando lo fueran, la verificación y validación se realiza con casos o muestras, lo que siempre crea imprecisión. Por esto, en muchas ocasiones en lugar de demostrar que el sistema es correcto, se intenta probar que el modelo y funcionamiento del sistema no es incorrecto. Esto permitirá aumentar la confianza en el modelo y en los resultados que se obtengan con el mismo [Domingo 1999].

Hay diversas posturas sobre en que momento es conveniente realizar la validación de un sistema. Bachant y McDermott [Bachant 1984] consideran que validar un sistema incompleto puede no resultar útil, mientras que Buchanan y Shortliffe [Buchanan 1984, 1989] recomiendan realizar la verificación y validación durante todo el desarrollo del sistema. El método usado preferentemente consiste

en la verificación y validación a medida que se va desarrollando, siguiendo sobre todo un desarrollo incremental en el cual al final de cada etapa de desarrollo se realice una validación. Por otra parte, el proceso de refinamiento del conocimiento surge en la etapa de adquisición y consiste en verificar y validar dicho conocimiento en busca de resultados incorrectos, posibles incoherencias, etc.

Así pues, cuando se construye un sistema se atraviesan varias etapas de modelización. Se comienza estudiando el sistema real, para construir el modelo conceptual que contiene la información relevante del sistema. Después se pasa a un modelo lógico que contiene las relaciones entre los elementos y por último se construye el modelo en el ordenador que ejecuta la lógica anterior. Este desarrollo incremental es un proceso iterativo en el que se refina cada etapa y en el que la verificación y validación correcta del sistema representa el paso a la siguiente.

Se han de establecer por tanto una serie de criterios para decidir si el modelo supera el proceso de verificación y validación. A continuación vamos a ver una serie de métodos para realizar este proceso.

La verificación y validación de un sistema inteligente debe ser llevada a cabo por varios tipos de personas para que sea lo más completa posible y abarque todas las etapas de desarrollo del modelo. Así, en las primeras fases la verificación y validación consiste en pruebas con casos ya resueltos, que será llevada a cabo por el ingeniero de conocimiento, mientras que a medida que el desarrollo se va incrementando, el número de casos será mayor y la validación será llevada a cabo no sólo por el ingeniero del conocimiento sino por expertos ajenos al desarrollo del sistema. En las fases finales la validación será en el entorno de los usuarios y se validarán sobre todo los aspectos de usabilidad del sistema. El personal involucrado en el proceso será por tanto [Mosquera 2002a]:

- El ingeniero del conocimiento que desarrolla el sistema que es quien mejor conoce las características del mismo.
- Expertos humanos que realizarán la validación de casos de prueba que serán contrastados con los resultados proporcionados por el sistema inteligente.
- Evaluadores independientes que son expertos ajenos al proyecto de desarrollo del sistema inteligente.

- Usuarios finales que participan generalmente en la validación orientada al uso.

2 MÉTODOS DE VERIFICACIÓN

El modelo lógico se encuentra entre el modelo conceptual y el modelo de ordenador. Para ser correcto tiene que contener todos los casos incluidos en el modelo conceptual, esto es, todos los sucesos, acciones y conexiones entre ellos. Para la verificación del proceso lógico uno de los métodos que se suele usar consiste en una revisión en la que el desarrollador del modelo explique con detalle la lógica del mismo [Jacobson 1997].

Una vez que el modelo lógico ha sido verificado se implementa el modelo de ordenador. En la verificación del mismo es necesario mostrar que la implementación ha sido correcta y que no hay errores en el programa. Existen varios métodos que se pueden usar y que se describirán a continuación:

- Métodos de programación estructurada.
- Testeo
- Simulación
- Verificación de modelos analíticos

2.1 Métodos de programación estructurada

Para realizar una verificación usando este método se siguen los principios de la programación estructurada. Éstos consisten en que el programa debe seguir un diseño descendente modular, empezando por el nivel más alto, que se compone de módulos o subsistemas que pueden estar interrelacionados, los cuales a su vez se pueden dividir en más módulos y así sucesivamente. Cada módulo es responsable de una función y no debe tener muchas líneas de código. Se debe verificar cada

módulo o subsistema de forma que se compruebe su correcto funcionamiento independientemente de los demás antes de verificar el sistema completo. Además, no se pueden incluir en el código (independientemente del lenguaje en el que esté escrito el programa) sentencias de ruptura que alteren el orden lógico de ejecución del programa (por ejemplo goto, exit, break, etc. en lenguaje C).

2.2 Testeo

Otro método de verificación del programa de ordenador consiste en realizar pruebas controladas sobre él (tests). Estas pueden ser de dos tipos:

- Tests ascendentes: Se comprueban en primer lugar todos los subsistemas que componen el sistema, luego las relaciones entre ellos y por último se prueba todo el sistema.
- Tests descendentes: Se comprueba el sistema principal y se va bajando hasta comprobar todos los subsistemas.

El sistema se puede verificar también usando condiciones extremas en los tests de forma que se asegure el buen funcionamiento del mismo incluso en los peores casos posibles.

2.3 Simulación

En ocasiones el sistema que se está desarrollando se aplica en un entorno crítico en el que el coste de una decisión errónea es muy alto (por ejemplo, operaciones, sistemas militares, fábricas, etc). En estos casos la verificación debe ser muy rigurosa, pero los tests que se pueden aplicar en estos entornos son muy limitados. Por ello se suelen usar modelos de simulación para verificar el funcionamiento del sistema en el entorno real.

2.4 Verificación de modelos analíticos

Un problema de análisis es aquél en el que a partir de una serie de casos las interpretaciones son asignadas dentro de una determinada categoría (problemas de diagnóstico, de predicción, etc.). Si se está modelando un sistema de este tipo puede ser posible verificar los datos y parámetros que se obtengan del modelo de ordenador con los resultados de un modelo analítico. Esto puede ayudar a comprobar si el modelo creado es correcto o no.

3 MÉTODOS DE VALIDACIÓN

Hay dos criterios para validar un sistema inteligente: validación contra los expertos o validación contra el problema.

En el primer criterio lo ideal es contar con un grupo de expertos o realizar un consenso entre ellos que pueda ser utilizado como estándar en la validación.

Los problemas que presenta este tipo de validación es que expertos de un mismo nivel pueden tener opiniones y soluciones diferentes para un mismo problema o incluso el mismo experto puede tener actitudes diferentes ante el mismo caso dependiendo de factores como estrés, cansancio, ambigüedades, etc.

Existen tres tipos posibles de validación contra los expertos:

- Validación contra un único experto: Sólo es recomendable en el caso de que la disponibilidad de expertos sea escasa, ya que al sólo contar con una opinión, los resultados del sistema pueden ser cuestionables o erróneos para otros expertos.
- Validación contra un grupo de expertos: Esta técnica tiene la ventaja respecto a la anterior de que se tiene la opinión de varios expertos lo que permite que hayan menos errores en los resultados y se pueda comparar el grado de consistencia existente entre los expertos del dominio. Si todos los expertos son de un nivel similar, lo que se

busca es que las interpretaciones del sistema inteligente se parezcan todo lo posible a las de los expertos.

- Validación contra un consenso de expertos: Esta opción consiste en unir las opiniones de varios expertos en una única opción, de forma que el resultado procura ser lo más objetivo posible. Si el acuerdo del sistema inteligente con el consenso es grande la confianza en el sistema inteligente aumenta considerablemente. El problema es que como en el caso de la validación contra único experto, todo lo que se encuentre fuera del consenso es considerado erróneo.

En cuanto a la validación contra el problema, se suele usar en tareas de pronóstico en las que se pueda saber si el resultado proporcionado por el sistema fue el correcto o no.

3.1 Métodos de Validación Cuantitativa

Los métodos cuantitativos se basan en la utilización de medidas estadísticas. Estas son medidas de pares en las que se comparan los resultados del tutorial con las interpretaciones de cada uno de los expertos para los casos propuestos. Se pueden agrupar en tres tipos: Medidas de acuerdo, de asociación y de asociación predictiva [Mosquera 2002b].

Las medidas de acuerdo proporcionan una medida de la coincidencia existente entre la interpretación de dos expertos, o de un experto y del sistema tutorial, Tabla 4-1.

Las medidas de asociación nos proporcionan el grado de asociación lineal existente entre el sistema tutorial y el experto. Una de las características de estas medidas es que sólo es posible aplicarlas en el caso de que las categorías de interpretación para los resultados sean ordinales, es decir, que se pueda dar un rango de ordenación entre los posibles resultados de los casos, Tabla 4-2.

Por último las medidas de asociación predictiva permiten determinar en que grado la interpretación del nuestro sistema puede utilizarse para predecir la interpretación del experto, Tabla 4-3.

Índice de acuerdo	Es el cociente entre el número de observaciones en las cuales hay acuerdo entre el Sistema y los expertos y el número de observaciones totales. Toma valores en el intervalo [0,1]. Cuanto más cerca esté de 1 mayor será el acuerdo.
Índice kappa	Medida de acuerdo en la que se corrigen los acuerdos que son debidos a la casualidad. Para ello se elimina un porcentaje de acuerdos que son los esperados debido a la casualidad, siendo éste la suma de los productos de las proporciones marginales correspondientes a la diagonal principal. (las proporciones marginales son la suma de todos los valores de una fila o columna). Un valor igual a 1 indica un acuerdo perfecto. Un valor igual a 0 indica que el acuerdo no es mejor que el que se obtendría por azar.
Kappa ponderada	Es una medida de acuerdo que corrige los acuerdos debidos a la casualidad y pondera de distinta forma los desacuerdos encontrados. Para ello se usa una matriz de pesos en la que se cuantifica el desacuerdo existente de forma que a la diagonal principal se le asigna el valor 0 indicando un acuerdo perfecto y a medida que se desplaza de la diagonal se va aumentando el valor del peso.

Tabla 4-1.- Medidas de acuerdo usadas en la validación de sistemas inteligentes

Tau-b de Kendall	Medida no paramétrica de la correlación para variables ordinales o de rangos que tiene en consideración los empates. No tiene en cuenta el grado de separación entre categorías. El signo del coeficiente indica la dirección de la relación y su valor absoluto indica la magnitud de la misma, de tal modo que los mayores valores absolutos indican relaciones más fuertes. Los valores posibles van de -1 a 1
Rho de Spearman	Coeficiente de correlación basado en rangos. Es adecuado a datos ordinales e invariables ante transformaciones que mantienen el orden. Tiene en cuenta el orden de separación de las categorías. Resultado entre 0 y 1 igual que Tau. Cuanto más cerca esté el valor de 1 mayor será la correlación existente.
Gamma	Medida de asociación simétrica entre dos variables ordinales cuyo valor siempre está comprendido entre -1 y 1. Los valores próximos a 1, en valor absoluto, indican una fuerte relación entre las dos variables. Los valores próximos a cero indican que hay poca o ninguna relación entre las dos variables.

Tabla 4-2.- Medidas de asociación usadas en la validación de sistemas inteligentes

Lambda	Medida de asociación que refleja la reducción proporcional en el error cuando se utilizan los valores de la variable independiente para pronosticar los valores de la variable dependiente. Un valor igual a 1 significa que la variable independiente pronostica perfectamente la variable dependiente. Un valor igual a 0 significa que la variable independiente no ayuda en absoluto a pronosticar la variable dependiente.
Tau de Goodman-Kruskal	Se obtiene usando el principio de reducción proporcional del error. Es una medida de asociación direccional. Un valor de 0 indica que no hay asociación y 1 una perfecta o completa asociación.

Tabla 4-3.- Medidas de asociación predictiva usadas en la validación de sistemas inteligentes

3.2 Métodos de Validación Cualitativa

Los métodos cualitativos son métodos que emplean técnicas subjetivas de comparación del rendimiento. Éstos implican a personas que conocen aspectos del sistema y pueden hacer juicios sobre el modelo que se ha usado. Existen varias pruebas de validación cualitativa que se describen a continuación:

- Validación Superficial
- Pruebas de Turing
- Test de Campo
- Validación de Subsistemas
- Análisis de Sensibilidad
- Grupos de Control
- Usabilidad del Sistema

3.2.1 VALIDACIÓN SUPERFICIAL

Se basa en reuniones entre los desarrolladores del Sistema Inteligente, expertos humanos y en ocasiones algún usuario. Las conclusiones a las que llega el sistema con los casos de prueba son analizados de forma subjetiva en estas reuniones.

3.2.2 PRUEBAS DE TURING

Estas pruebas fueron desarrolladas a partir de una idea propuesta por el matemático Alan Turing en 1950. Se pretende reunir a un grupo de expertos humanos y hacer que estos valoren una serie de casos. Posteriormente estos mismos expertos deben analizar los resultados de sus compañeros y del sistema inteligente sin saber qué resultado pertenece a cada uno. De esta forma se pretenden evitar tendencias a favor o en contra del sistema o de las opiniones de otros expertos.

3.2.3 TEST DE CAMPO

Consisten en colocar el sistema inteligente en el entorno en el que se va a utilizar y permitir que los usuarios interaccionen con él.

Las ventajas de este método son que:

- Parte de las tareas de verificación y validación las efectúan los propios usuarios del sistema.
- Cuando no se notifiquen más problemas se ha llegado a un rendimiento aceptable del sistema.
- Se pueden descubrir errores que anteriormente no habían sido descubiertos.

En cambio presenta las siguientes desventajas:

- Pueden surgir cuestiones de los usuarios que tengan poca relación con el rendimiento del sistema.

- Si estamos probando un prototipo (no el sistema definitivo) y este es muy incompleto o presenta muchos errores se puede perder credibilidad.
- Sólo se puede probar en entornos no críticos.

3.2.4 VALIDACIÓN DE SUBSISTEMAS

Se hace uso de una técnica de "divide y vencerás" para detectar de forma más fácil los errores del sistema y facilitar la validación. Se divide la base de conocimiento en subsistemas o módulos que se validan por separado. Lo que hay que tener en cuenta es que puede ser complicado dividir algunos sistemas y que la validación de todos los subsistemas por separado podría no ser equivalente a la validación del sistema completo [Mosquera 2001].

3.2.5 ANÁLISIS DE SENSIBILIDAD

Esta técnica es especialmente útil en sistemas con incertidumbre. Consiste en presentar a la entrada del sistema, una serie de casos similares, pero con pequeñas diferencias. Así el impacto de dichas variaciones en los casos de entrada puede ser estudiado observando los cambios en la salida.

3.2.6 GRUPOS DE CONTROL

Esta técnica consiste en presentar los casos a dos grupos de expertos: unos que no utilicen el sistema inteligente y otros que si trabajen con él. Estos últimos constituyen el grupo de control. Así se puede comparar el rendimiento de los expertos cuando utilizan el sistema experto y cuando no lo utilizan con el objeto de comprobar si el sistema realmente simplifica el trabajo a realizar por el experto.

3.2.7 USABILIDAD DEL SISTEMA

Uno de los aspectos más importantes a analizar en la evaluación de Sistemas Inteligentes es la usabilidad del sistema. Si el sistema resulta difícil de manejar, es incómodo, poco intuitivo, etc., aunque los resultados sean los correctos los usuarios finales pueden estar poco interesados en usar el sistema.

Existen varios métodos para analizar la usabilidad de los sistemas inteligentes que se verán en los siguientes apartados:

- Métodos Heurísticos
- Métodos Subjetivos
- Métodos Empíricos

3.2.7.1 *Métodos heurísticos*

Los métodos heurísticos evalúan la usabilidad a través de técnicas heurísticas que desarrollan expertos que analizan las interfaces de los módulos, evalúan su arquitectura y determinan sus puntos débiles y fuertes desde el punto de vista del usuario final del sistema. Las técnicas que pueden emplearse son:

- Cuestionarios ergonómicos: Contienen una serie de directrices que el sistema debe seguir para ser usable.
- Inspección de interfaces: Técnica de muestreo cuyo objetivo es reunir datos acerca de los aspectos físicos de la interfaz del sistema. Se diferencia de los cuestionarios ergonómicos en que el análisis es menos detallado (se centra en aspectos más globales).
- Evaluación de la navegación: Se evalúa la facilidad de navegación de los usuarios por los menús del sistema.
- Análisis formales: Se usan como modelos predictivos de la interacción del usuario con el sistema. El más popular es el modelo GOMS (goals, operators, methods and selection rules). Otros métodos son: CCT (teoría de la complejidad cognitiva), CLG (gramática del lenguaje de

comandos), TAG (gramática de tareas-acciones), TAKD (análisis de tareas para las descripciones basadas en el conocimiento) y el modelo de las siete etapas de acción de Norman [Mosquera 2002b].

3.2.7.2 *Métodos Subjetivos*

Se basan en obtener información de los usuarios mediante prototipos. Se pueden distinguir seis métodos subjetivos:

- **Pensar en alto:** Test en los que el usuario va comentando en voz alta las incidencias que ocurren al probar el sistema.
- **Observación:** Consiste en observar como el usuario maneja el sistema para sacar conclusiones acerca de la usabilidad. Las técnicas a emplear pueden ser: la observación directa, la grabación en vídeo, etc.
- **Cuestionarios:** Consisten en presentarle al usuario un cuestionario de preguntas y analizar sus respuestas. Hay dos tipos básicos: cuestionarios abiertos en los que el usuario desarrolla sus propias respuestas u opiniones y cuestionarios cerrados en los que el usuario tiene que escoger una alternativa entre varias que se le plantean.
- **Entrevistas:** Se realizan entrevistas con los usuarios para obtener información de la usabilidad del sistema. Las entrevistas estructuradas se parecen bastante a los cuestionarios vistos anteriormente pero que tienen la ventaja de que el entrevistador puede comentar la pregunta con el usuario. En las entrevistas flexibles en cambio la estructura de la entrevista dependerá de respuestas previas del usuario.
- **Grupos de control:** Son entrevistas en grupo realizadas a usuarios del sistema.
- **Retroalimentación con el usuario:** Consiste en que los usuarios comuniquen a los desarrolladores del sistema sus problemas o dudas a la hora de la utilización.

3.2.7.3 Métodos Empíricos

Este método consiste en obtener datos objetivos sobre la utilización del sistema por parte de los usuarios finales. Suelen referirse a los aspectos de exactitud y tiempo.

Las medidas objetivas según Nielsen que pueden utilizarse para evaluar la usabilidad de un sistema son:

- Medidas de exactitud
 - Número de tareas diversas que pueden realizarse en un periodo de tiempo.
 - Proporción entre interacciones correctas y errores.
 - Número de errores cometidos por el usuario.
- Medidas de tiempo
 - Tiempo que los usuarios consumen para realizar una tarea.
 - Tiempo consumido en recuperación de errores.
- Otras
 - Número de características del sistema utilizadas por los usuarios.
 - Frecuencia de uso de los manuales o sistemas de ayuda y tiempo consumido utilizándolos.
 - Proporción de usuarios que prefieren usar este sistema en vez de otro similar o competidor.

4 VERIFICACIÓN DEL ITS

En el diseño de nuestro ITS hemos usado una metodología incremental en la que el sistema se fue construyendo en base a tres fases principales:

- El sistema Multiagente que es el que se encarga de determinar el siguiente objetivo (y actividad dentro de ese objetivo) en el que tiene que trabajar el alumno.
- El sistema borroso que es el que se encarga de determinar el nivel de dificultad en el que tiene que trabajar el alumno en un objetivo concreto.
- Interfaz gráfica que se encarga de mostrarle la siguiente actividad al alumno e interactuar con él.

Así pues, después de la terminación de cada una de estas fases se tenía un prototipo que podía ser verificado y validado independientemente de los demás. Una vez que se había realizado la verificación de cada subsistema por separado, realizamos la verificación del sistema completo para comprobar que al unir todas las partes seguían funcionando correctamente.

Como el sistema inteligente considera tres tipos de usuarios diferentes (motivados, con miedo al fracaso e hiperactivos) se simularon casos de cada uno de los tipos. Esto es así porque el flujo a seguir por el Sistema Inteligente depende en gran medida de esta característica. Se observó que el flujo de información era el que indicaban los expertos profesores. Así por ejemplo en la simulación de un alumno con miedo al fracaso realizaba muchas más actividades antes de pasar a una nueva fase, mientras que los hiperactivos se adelantaban mucho aunque se volvía con facilidad a fases ya estudiadas. El alumno motivado representa el caso intermedio.

Para realizar esta verificación definimos los casos de uso del sistema. En los apartados siguientes vamos a ir viendo cada uno de ellos. En la tabla 4-4 se

encuentra un resumen de los mismos. En la figura 4-1 se pueden ver de forma gráfica las interacciones de los actores (profesores, alumno, tutorial inteligente).

Casos de Uso	Actores	Propósito	Resumen
Calcular Nivel	Alumno, Gestor, Sistema Borroso	Calcular el nivel de dificultad para un objetivo.	Una vez que el Alumno ha resuelto actividades de un objetivo el Gestor se encarga de activar el Sistema Borroso y de enviarle la información. Éste calcula cuál es el siguiente nivel en el que tendrá que trabajar el Alumno para ese objetivo.
Calcular Objetivo	Gestor, Sistema MultiAgente	Calcular el siguiente objetivo en el que trabajará el Alumno	Para determinar el siguiente objetivo en el que va a trabajar el Alumno, el Gestor se encarga de activar el Sistema MultiAgente y enviarle la información del mismo. Dicho Sistema determina en función de estos datos el próximo objetivo a mostrar.
Calcular Tipo de Actividad	Gestor, Sistema MultiAgente	Calcular el tipo de actividad que se le mostrará al Alumno.	Cuando se ha determinado el siguiente objetivo y el nivel en el que tiene que trabajar el Alumno, el Gestor se encarga de activar el Sistema MultiAgente y enviarle la información de éste. El Sistema MultiAgente determina para dicho objetivo cuál es el tipo de actividad que va a tener que resolver el Alumno.
Mostrar Actividad	Alumno, Gestor, Interfaz Multimedia	Mostrarle al Alumno la actividad (o actividades) que tiene que resolver	El Gestor conociendo el objetivo, el nivel para ese objetivo y el tipo de actividad dentro de ese nivel que le corresponden al Alumno selecciona una actividad o varias, que se le muestran al Alumno mediante la Interfaz Multimedia.
Insertar Alumno	Profesor, Asistente BD	Incluir un nuevo Alumno en el Tutorial	El Profesor utiliza el Asistente BD para introducir los datos personales de un nuevo Alumno incluyéndolo en la aplicación.
Modificar Alumno	Profesor, Asistente BD	Modificar los datos de un Alumno.	El Profesor utiliza el Asistente BD para modificar los datos personales de un Alumno existente en la aplicación.
Eliminar Alumno	Profesor, Asistente BD	Eliminar un Alumno del Tutorial	El Profesor usa el Asistente BD para eliminar un Alumno existente en la aplicación.
Consultar Resultados	Profesor, Asistente BD	Consultar los resultados obtenidos por un Alumno.	El profesor utiliza el Asistente BD para consultar los resultados que ha obtenido un Alumno para los objetivos de cada fase del Tutorial.

Tabla 4-4.- Funcionalidad de cada caso de uso

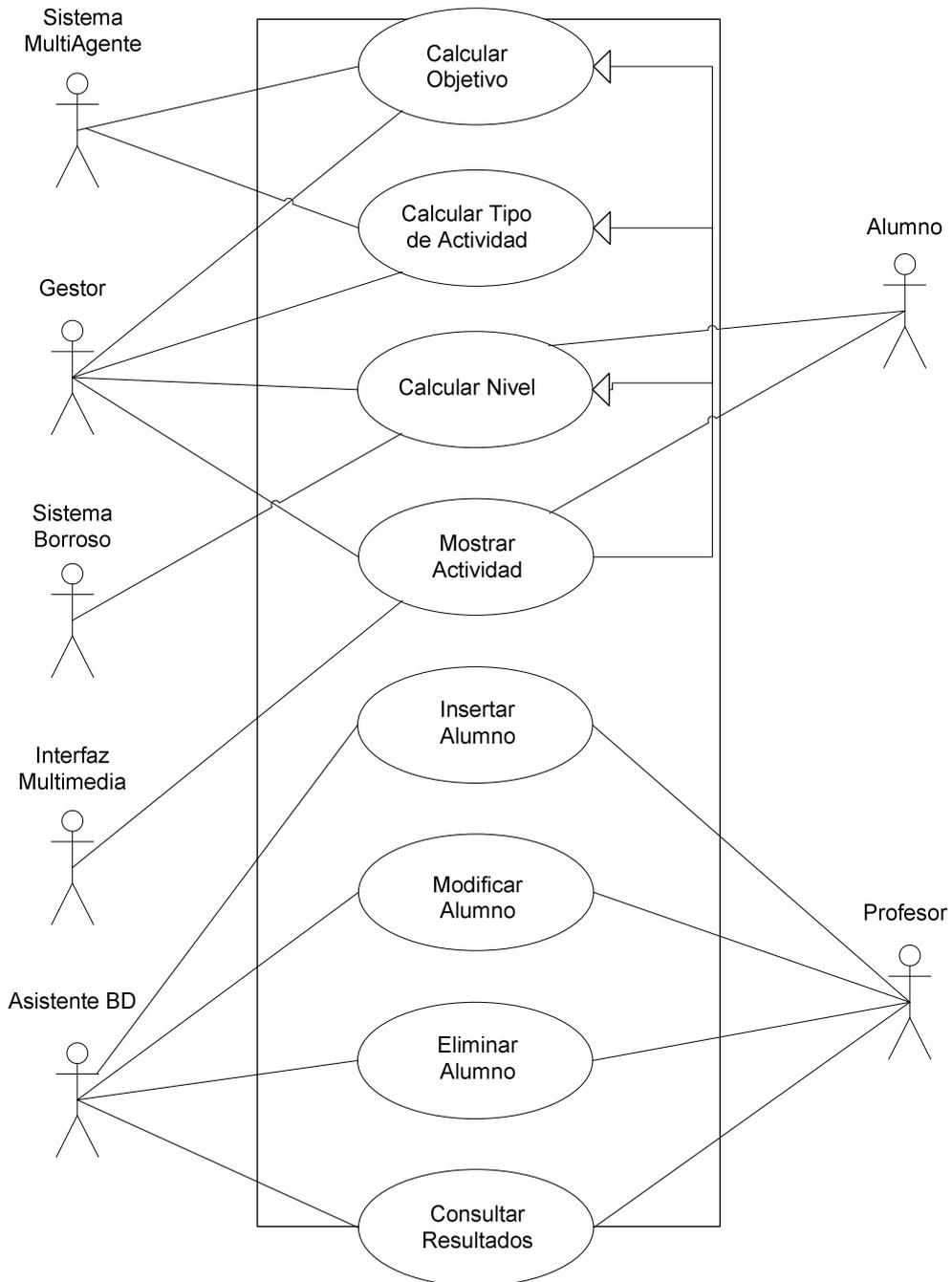


Figura 4-1.- Diagrama de casos de usos del sistema inteligente

4.1 Casos de Uso

4.1.1 CALCULAR NIVEL

Actores: Alumno, Gestor, Sistema Borroso.

Propósito: Calcular el nivel de dificultad para un objetivo.

Resumen: Una vez que el Alumno ha resuelto actividades de un objetivo, el Gestor se encarga de activar el Sistema Borroso y de enviarle la información. Éste calcula cuál es el siguiente nivel en el que tendrá que trabajar el Alumno para ese objetivo (puede ser subir de nivel de dificultad, bajar o mantenerse).

Curso Normal de los Eventos:

1. El Alumno resuelve actividades de un objetivo en un determinado nivel de dificultad.
2. El Gestor actualiza los resultados del Alumno en la Base de Datos (BD) y ejecuta el Sistema Borroso.
3. El Sistema Borroso calcula el nivel en el que debe trabajar el Alumno para ese objetivo y le envía el resultado al Gestor.
4. El Gestor actualiza el nivel del objetivo en la BD.

4.1.2 CALCULAR OBJETIVO

Actores: Gestor, Sistema Multiagente.

Propósito: Calcular el siguiente objetivo en el que trabajará el Alumno.

Resumen: En cada sesión el Alumno trabaja en una serie de objetivos. Para determinar el siguiente objetivo en el que va a trabajar el Alumno, el Gestor se encarga de activar el Sistema Multiagente y enviarle la información del mismo. Dicho Sistema determina en función de estos datos el próximo objetivo a mostrar.

Curso Normal de los Eventos:

1. El Gestor ejecuta el Sistema Multiagente y le envía la información del Alumno.
2. El Sistema Multiagente calcula el siguiente objetivo que se le va a mostrar al Alumno y envía el resultado al Gestor.
3. El Gestor actualiza el objetivo en la BD.

4.1.3 CALCULAR TIPO DE ACTIVIDAD

Actores: Gestor, Sistema Multiagente

Propósito: Calcular el tipo de actividad que se le mostrará al Alumno.

Resumen: Una vez que se determina el siguiente objetivo y el nivel en el que tiene que trabajar el Alumno, el Gestor se encarga de activar el Sistema Multiagente y enviarle la información del alumno. El Sistema Multiagente determina para dicho objetivo cuál es el tipo de actividad que va a tener que resolver el Alumno.

Curso Normal de los Eventos:

1. El Gestor ejecuta el Sistema Multiagente y le envía la información del Alumno.
2. El Sistema Multiagente calcula el tipo de actividad para el objetivo actual que se le va a mostrar al Alumno y envía el resultado al Gestor.
3. El Gestor actualiza el tipo de actividad en la BD.

4.1.4 MOSTRAR ACTIVIDAD

Actores: Alumno, Gestor, Interfaz Multimedia

Propósito: Mostrarle al Alumno la actividad (o actividades) que tiene que resolver.

Resumen: El alumno está a la espera de la actividad que tiene que realizar. El Gestor tiene que conocer el objetivo, el nivel para ese objetivo y el tipo de actividad dentro de ese nivel que le corresponden al Alumno. Una vez que conoce estos datos selecciona una actividad o varias, que se le muestran al Alumno y se almacenan los resultados en la BD.

Curso Normal de los Eventos:

1. El Alumno está esperando que se le muestre una actividad.
2. Se activa el caso de uso: Calcular Objetivo.
3. Se activa el caso de uso: Calcular Nivel.
4. Se activa el caso de uso: Calcular Tipo de Actividad.
5. El Gestor busca en la Base de Datos una o varias actividades que cumplan los requisitos de objetivo, nivel y tipo de actividad calculados anteriormente.
6. El Gestor activa la Interfaz Multimedia y se le muestra la actividad (o actividades) al Alumno.
7. El Alumno resuelve la actividad y el Gestor actualiza la BD con los resultados obtenidos (porcentaje de aciertos, número de actividades realizadas, ...)

Curso Alternativo de los Eventos:

Paso 2: Si es la primera vez que se ejecuta el Tutorial (puesto que todos los objetivos tienen la misma importancia), el primer objetivo se elige de forma aleatoria.

4.1.5 INSERTAR ALUMNO

Actores: Profesor, Asistente BD.

Propósito: Incluir un nuevo Alumno en el Tutorial.

Resumen: El Profesor utiliza el Asistente BD para introducir los datos personales de un nuevo Alumno incluyéndolo en la aplicación.

Curso Normal de los Eventos:

1. El profesor activa el Asistente BD.
2. El Asistente BD muestra en un navegador una pantalla donde aparecen los datos personales de los Alumnos que se encuentran ya incluidos en el Tutorial.
3. El Profesor elige la opción Insertar Alumno.
4. El Asistente muestra una pantalla con una serie de campos vacíos donde hay que rellenar los datos personales del Alumno.
5. El Profesor introduce los datos personales del Alumno y se actualiza la BD insertando el nuevo Alumno.

4.1.6 MODIFICAR ALUMNO

Actores: Profesor, Asistente BD.

Propósito: Modificar los datos de un Alumno.

Resumen: El Profesor utiliza el Asistente BD para modificar los datos personales de un Alumno existente en la aplicación.

Curso Normal de los Eventos:

1. El Profesor activa el Asistente BD.
2. El Asistente BD muestra en un navegador una pantalla donde aparecen los datos personales de todos los Alumnos que se encuentran incluidos en la aplicación.
3. El Profesor busca el Alumno cuyos datos quiere modificar.
4. El Profesor modifica los datos personales del Alumno y se actualiza los nuevos datos en la BD.

4.1.7 ELIMINAR ALUMNO

Actores: Profesor, Asistente BD.

Propósito: Eliminar un Alumno del Tutorial.

Resumen: El Profesor usa el Asistente BD para eliminar un Alumno existente en la aplicación.

Curso Normal de los Eventos:

1. El Profesor activa el Asistente BD.
2. El Asistente BD muestra en un navegador una pantalla donde se pueden ver todos los Alumnos que se encuentran incluidos en la aplicación.
3. El Profesor busca el Alumno que quiere eliminar.
4. El Profesor activa la opción eliminar del Asistente BD y se actualiza la BD eliminando a dicho Alumno.

4.1.8 CONSULTAR RESULTADOS

Actores: Profesor, Asistente BD.

Propósito: Consultar los resultados obtenidos por un Alumno.

Resumen: El profesor utiliza el Asistente BD para consultar los resultados que ha obtenido un alumno para los objetivos de cada fase.

Curso Normal de los Eventos:

1. El profesor activa el Asistente BD.
2. El Asistente BD muestra en un navegador una pantalla donde se pueden ver los datos para cada alumno existente en la aplicación.
3. El Profesor busca el Alumno cuyos resultados quiere consultar.
4. Para el Alumno seleccionado el Asistente BD muestra en que fase está actualmente el Alumno y para cada una de las fases que los

objetivos están asociados a ella. Por cada objetivo se muestra en qué nivel de dificultad se encuentra el alumno y el porcentaje de aciertos que ha obtenido en la resolución de las actividades asociadas a ese objetivo y nivel.

5 VALIDACIÓN DEL ITS

En general nuestro problema puede ser clasificado dentro de los problemas de síntesis ya que los subsistemas que se han definido pretenden resolver problemas de toma de decisiones (decidir el siguiente objetivo a mostrar en el caso del sistema Multiagente y decidir el nivel en el que debe trabajar el alumno en el sistema borroso). Además, en cuanto a la relación con el entorno el sistema tutorial inteligente actúa de forma aislada y en un entorno no crítico, lo que facilita su validación.

Para realizar dicha validación hemos utilizado tanto los métodos de validación cuantitativos como cualitativos. Los cualitativos usan técnicas subjetivas para determinar el rendimiento y usabilidad del sistema, mientras que los cuantitativos usan medidas estadísticas. Así pues la combinación de ambos métodos nos ofrece mejores resultados [Zlatareva 1998].

5.1 Validación cuantitativa

En la validación cuantitativa se ha usado una serie de casos (observaciones) tanto para el sistema borroso como para el multiagente, de forma que nos permitan la comparación de los resultados obtenidos de la simulación del tutorial con los resultados que propondrían los expertos para dichos casos. Esta serie de casos cumplen con los dos criterios esenciales para que la muestra sea significativa: cantidad (que el número de casos sea significativo) y representatividad (que los casos tengan variedad en el dominio de aplicación). Estos casos los hemos validado contra un grupo de expertos, ya que esta técnica tiene la ventaja de que se tiene la opinión de varios expertos (en lugar de la de uno solo o de un consenso de ellos) lo

que provoca una disminución de errores en los resultados y permite además comparar el grado de acuerdo o similitud entre las respuestas de los expertos.

Un ejemplo de caso para la validación del sistema borroso sería el siguiente: Teniendo a un alumno hiperactivo realizando actividades de nivel medio y teniendo en cuenta que su evolución histórica en dichas actividades es muy baja, si a la hora de realizar las nuevas actividades obtiene un porcentaje de acierto del 50% de las mismas, ¿cuál sería el siguiente nivel de dificultad que se le propondría al alumno? ¿Se le mantendría en ese nivel, se le bajaría al nivel inferior, o se le subiría a nivel alto?

Las medidas estadísticas que hemos usado para la validación cuantitativa son medidas de pares en las que se comparan los resultados del tutorial con las interpretaciones de cada uno de los expertos para los casos propuestos.

5.1.1 VALIDACIÓN CUANTITATIVA DEL SISTEMA BORROSO

La variable de salida del sistema borroso es una variable categórica ordinal que consta de tres categorías (regresión, permanencia y progresión). Éstas representan para un alumno dentro de un objetivo concreto el hecho de tener que bajar el nivel de dificultad de las actividades con las que trabaje en ese objetivo (regresión), permanecer en ese nivel (permanencia) o subir el nivel de dificultad (progresión). Las técnicas de validación que vamos a utilizar están dentro de las medidas de acuerdo: el índice de acuerdo dentro de uno y la medida de kappa ponderada. Dentro de las medidas de asociación usaremos las medidas derivadas de la tau de Kendall (la tau b y la gamma) y la rho de Spearman.

En la figura 4-2 se muestra un resumen de los resultados de la validación para el sistema borroso que determina el nivel de la siguiente actividad a presentar al alumno. Se muestran los resultados de la validación realizada con los expertos tanto por aquéllos que han participado en el desarrollo del proyecto (experto1, ... , experto3), como por expertos ajenos al mismo (experto4, ... , experto 15).

Para cada experto se muestra el valor obtenido en: el índice de acuerdo (azul), kappa ponderada (morado), la tau b (amarillo), gamma (verde) y la rho de Spearman (violeta). Es interesante observar que todos los valores obtenidos son superiores en media a 0,80. El valor estadístico Gamma está muy próximo a 1 en

todos los casos, lo que muestra una fuerte relación entre los resultados de los expertos y los obtenidos con el planificador borroso.

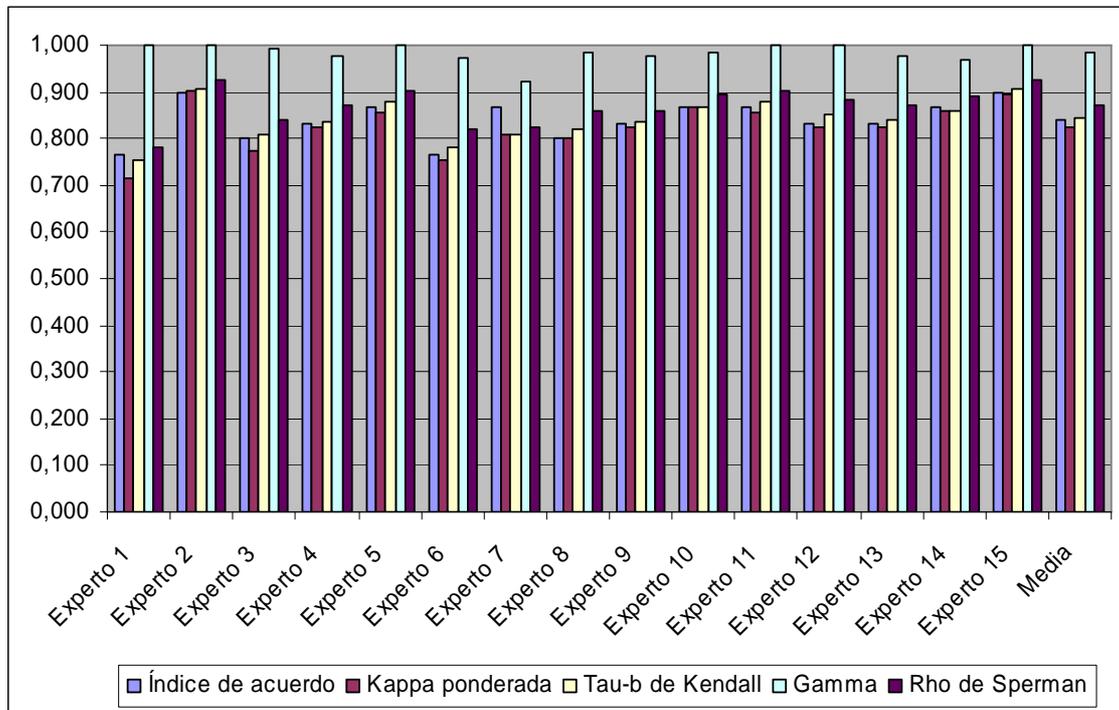


Figura 4-2.- Resultados de la validación del Sistema Borroso

5.1.2 VALIDACIÓN CUANTITATIVA DEL SISTEMA MULTIAGENTE

El sistema Multiagente (MAS) se puede dividir en dos subsistemas:

1. Sistema Multiagente para los Objetivos (MASObj), que determina cuál es el siguiente objetivo con el que tendrá que trabajar el alumno.
2. Sistema Multiagente para las Actividades (MASAct), que indica el tipo de la siguiente actividad que se le mostrará al alumno.

La variable de salida del MASObj es una variable categórica nominal que consta de tantas categorías como objetivos existen dentro de una fase. Y la variable de salida del MASAct también es una variable categórica con tantas categorías como tipos de actividades existen dentro de un objetivo.

Para la validación utilizaremos como medidas de acuerdo los índices de acuerdo y el índice kappa. No se justifica el uso del índice kappa ponderado ya que no existen unas discrepancias más graves que otras entre las categorías. Además estudiaremos las variables de asociación predictivas para determinar en qué grado la interpretación del Sistema Multiagente puede utilizarse para predecir la interpretación del experto. Concretamente utilizaremos la lambda (λ) de Guttman y de Goodman-Kruskal y la tau (τ) de Goodman-Kruskal.

5.1.2.1 Validación Cuantitativa del Sistema MASObj

En la figura 4-3 se muestran los resultados de la validación del sistema frente a los expertos. Se han comparado mediante una serie de casos los resultados que proporciona el Sistema Multiagente a la hora de elegir cuál es el siguiente objetivo a desarrollar por el alumno frente a los resultados que deciden los expertos para esos mismos casos.

Igual que en la validación del sistema borroso, se muestran los resultados de la validación realizada con los expertos tanto por aquéllos que han participado en el desarrollo del proyecto (experto1, ... , experto3), como por expertos ajenos al mismo (experto4, ... , experto 15).

Para cada experto se muestra el valor obtenido en: el índice de acuerdo (azul), kappa ponderada (morado), lambda (amarillo) y tau de Goodman-Kruskal (verde). Se observa que para los índices de acuerdo obtenemos valores superiores a 0,80 siendo un poco inferior este resultado para las medidas de asociación predictiva, aunque en media los resultados son superiores a un 70%.

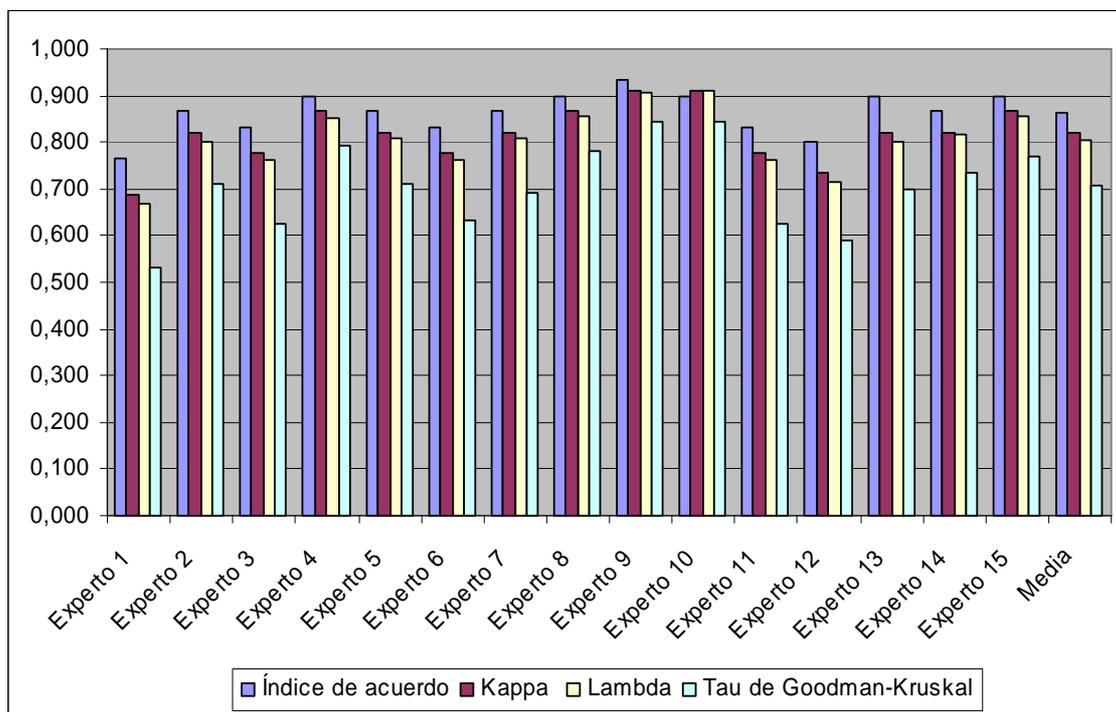


Figura 4-3.- Resultados de la validación del Sistema MASObj

5.1.2.2 Validación Cuantitativa del Sistema MASAct

En la figura 4-4 se observan los resultados de la validación del Sistema Multiagente a la hora de decidir la siguiente actividad a mostrar al alumno (habiéndose elegido previamente el objetivo al que pertenece). Los expertos participantes son los mismos que para la validación cualitativa del Sistema MasObj. Igual que en la validación anterior, en la figura se muestra el valor obtenido en: el índice de acuerdo (azul), kappa ponderada (morado), lambda (amarillo) y tau de Goodman-Kruskal (verde). Se puede observar que en término medio todos los resultados obtenidos son superiores al 75%.

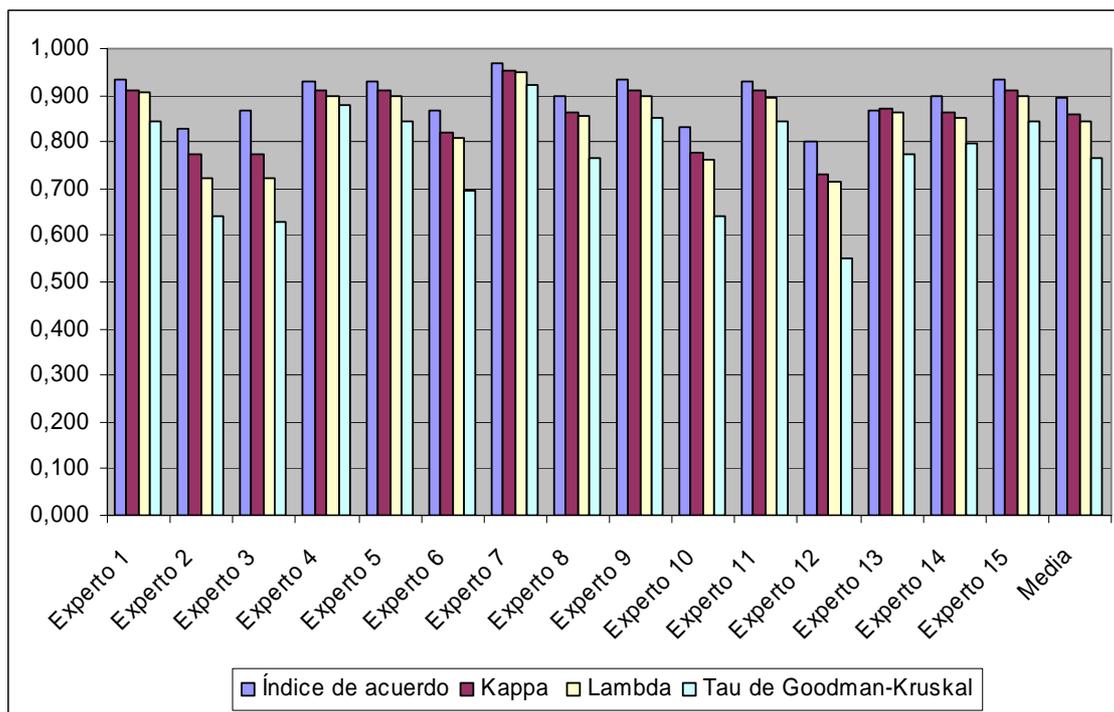


Figura 4-4.- Resultados de la validación del Sistema MASAct

5.2 Validación cualitativa

Como ya se ha comentado, en la validación cualitativa se usan técnicas subjetivas para determinar el rendimiento y el uso del sistema. En nuestro caso ha sido realizada por los dos tipos principales de usuarios:

- Los profesores de los centros que van utilizar el tutorial inteligente que son expertos en el dominio y por lo tanto usuarios válidos a la hora de realizar las pruebas con la aplicación (test de campo, puesto que el dominio no es crítico y usabilidad).
- Los alumnos de infantil y primaria a los que va destinado el sistema inteligente, que obviamente no son expertos en el dominio y por lo tanto colaborarán en la parte de validación correspondiente al uso del sistema.

5.2.1 USABILIDAD DE LA INTERFAZ GRÁFICA

En primer lugar para comprobar la usabilidad del sistema se hicieron cuestionarios ergonómicos del ITS para los expertos. La interfaz Multimedia del ITS fue verificada en función de las siguientes preguntas:

- a. ¿Los menus son lógicos y fáciles de usar? Se confirmó que un entorno web facilitaba el uso del tutorial puesto que era conocido por los profesores.
- b. ¿Es la interacción con el sistema correcta desde el punto de vista del usuario al que va dirigido? Como se comentó en el capítulo 2, en este punto hubo que cambiar el mecanismo en el que se producía el audio. Inicialmente se utilizó text-to-speech pero los alumnos no entendían el texto. Por ello se decidió grabar con voces de profesores las distintas informaciones.
- c. ¿Son fácilmente comprensibles los feedbacks del sistema? Una vez cambiado el audio en el sentido del apartado anterior, no hubo problemas con los mensajes.
- d. ¿La imagen conceptual del sistema es la misma que la del usuario? Determinadas imágenes eran fácilmente entendibles por los alumnos de infantil, así que se contextualizaron las tareas y se eliminaron aquellos gráficos complicados.

5.2.2 VALIDACIÓN CUALITATIVA DEL ITS

La validación cualitativa se realizó con alumnos de la Asociación Tinerfeña de Trisómicos 21 (ATT21) que están integrados en centros de infantil y primaria siguiendo el currículo de diferentes niveles educativos. En dicha asociación reciben clases de apoyo escolar sobre diferentes disciplinas. Los datos se recogieron en varias sesiones que fueron grabadas en videocámara durante la clase de apoyo escolar. El profesor permanece sentado junto al alumno durante las sesiones por si necesita ayuda con el ordenador. La validación cualitativa se realizó en tres etapas: en primer lugar los alumnos trabajaron con las actividades de forma independiente,

luego con secuencias de actividades fijas para poder realizar una comparación entre los alumnos sin que intervenga el grado de dificultad de las actividades y posteriormente con el ITS. La validación ha sido llevada a cabo en colaboración con miembros del Departamento de Análisis Matemático (área de Didáctica de las Matemáticas) de la Universidad de La Laguna [Aguilar 2003c].

En la fase 1 del tutorial los datos a analizar fueron obtenidos por las respuestas de 13 alumnos con Síndrome de Down a actividades de Clasificación, Correspondencia uno a uno, Seriaciones y Cuantificadores. Siete de los alumnos (A1, ..., A7), aunque están integrados en aulas de infantil y primaria, siguen el currículo de infantil (Inf). Los otros seis son alumnos (A8, ..., A13) no escolarizados que reciben clases en diversas disciplinas son llamados alumnos de alfabetización (Alf) siendo el currículo de Matemáticas que se les imparte también correspondiente a infantil. Las edades de dichos alumnos oscilan entre los 5 y los 26 años [Bruno 2006].

En total los alumnos trabajaron con 174 actividades: 31 de Clasificación (CL), 51 de Relaciones de Orden (RO), 36 de Correspondencia (CO) y 56 de cuantificadores (CU). Recordar que el tutorial muestra a los alumnos actividades diferentes en función de sus respuestas, por lo que el análisis de los resultados se hará de forma global.

El porcentaje de aciertos de los alumnos al trabajar con estas actividades se puede ver en la figura 4-5. Todos los conceptos tienen porcentaje de aciertos por encima del 60% menos las actividades de Relaciones de Orden que presentan para este tipo de alumnos una mayor dificultad [Aguilar 2005c].

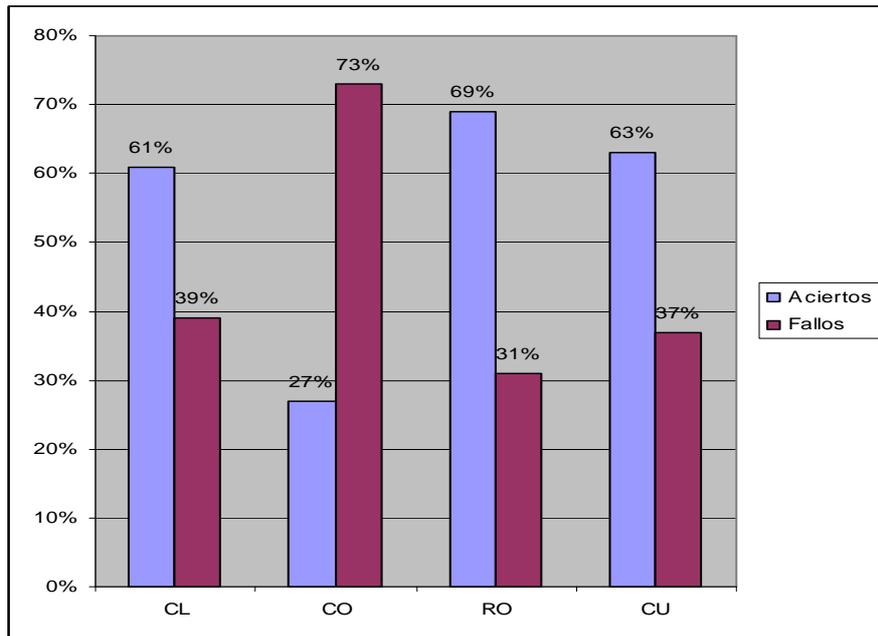


Figura 4-5. Porcentajes de aciertos y fallos en la resolución de las actividades por parte de los alumnos.

Para realizar la validación se prepararon actividades de dos tipos dentro de cada objetivo, siendo la tipo 2, de mayor dificultad que la tipo 1 (tabla 4-5). Si analizamos los resultados en función del tipo de actividad comprobamos como era de esperar que, los alumnos obtuvieron para todos los objetivos mejores resultados en la resolución de las actividades de tipo 1 que en las de tipo 2 (figura 4-6).

En cuanto al manejo del tutorial por parte de los alumnos tenemos interés en comprobar el comportamiento de los alumnos frente al tutorial, saber si son autónomos, si usan correctamente el ratón, etc.

En primer lugar se considera un éxito que el alumno sea capaz de realizar dos acciones fundamentales al resolver las actividades como son pinchar con el ratón y pinchar y mover sin la ayuda del profesor (recordar que la interacción de pinchar y mover sustituye a la de arrastrar para que fuera más fácil su utilización para los alumnos).

Tipos de Actividades	Total de Actividades
CL Tipo 1 Elegir objetos idénticos entre sí.	13
CL Tipo 2 Reconocer las características de un conjunto y separar elementos que no pertenezcan a él.	18
CO Tipo1 Relacionar dos conjuntos de objetos iguales.	23
CO Tipo 2 Relacionar dos conjuntos de objetos con una relación pero no iguales.	13
RO Tipo 1 Hacer seriaciones simples.	25
RO Tipo 2 Hacer seriaciones con alternancia de elementos y una o más variables.	26
CU Tipo 1 Seleccionar todos los objetos de un conjunto que posean, o no, una determinada característica.	30
CU Tipo 2 Extraer elementos de un conjunto, hasta que no quede ninguno.	26

Tabla 4-5. Tipos de actividades fase 1.

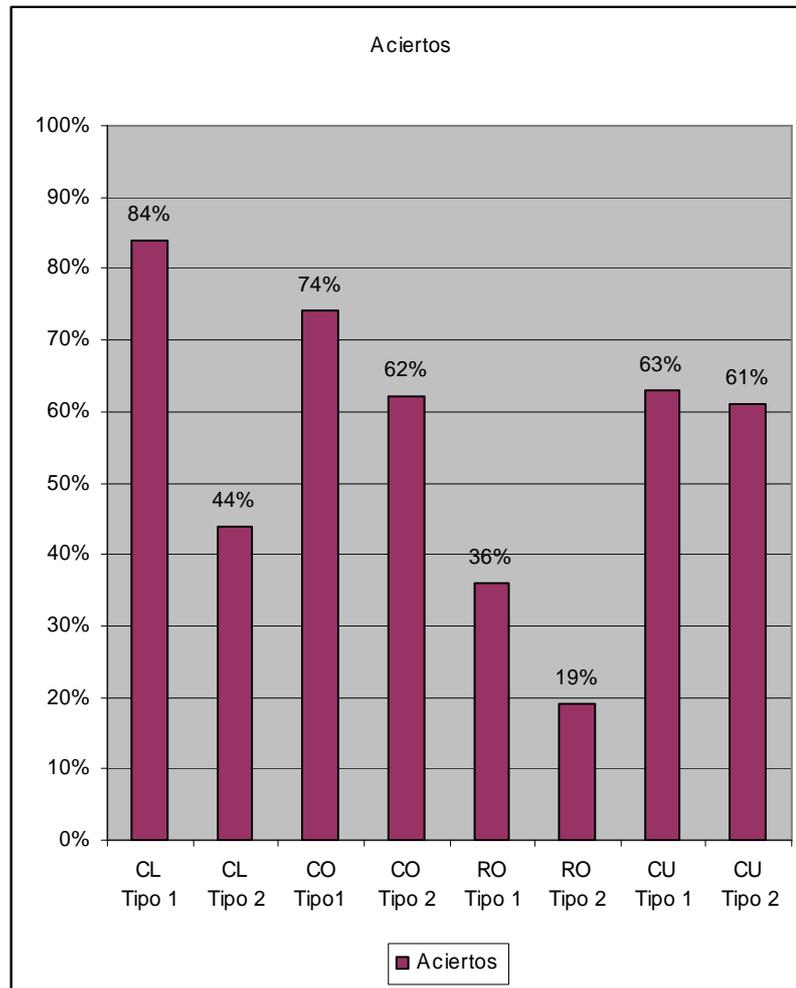


Figura 4-6. Porcentaje de aciertos de los alumnos por tipo de actividad.

Los resultados se pueden ver en la figura 7. Se observan diferencias considerables entre los alumnos en cuanto al manejo del ratón. Algunos consiguieron muy buenos resultados frente a otros que necesitaron la ayuda del profesor de forma casi constante. Se comprobó que en algunos casos los alumnos decían en voz alta la respuesta correcta pero les costaba la interacción con el ratón por lo que tendían a señalar con el dedo la pantalla del ordenador (en algunos casos ni siquiera intentaban utilizar el ratón si no que directamente tocaban la pantalla). Estos resultados hicieron que nos planteáramos la conveniencia del uso de una pantalla táctil. Además en algunos casos el escaso uso del ordenador en el ambiente familiar y escolar (de hecho para algunos era la primera vez que

manejaban un ordenador) y las dificultades motrices de los niños con Síndrome de Down también representan un problema en su interacción con el ordenador.

Respecto a la autonomía de los alumnos, es decir, si los alumnos pueden resolver solos las actividades sin la ayuda del profesor, se puede observar en los resultados que ningún alumno fue totalmente autónomo frente al ordenador aunque la mayoría superó el 50% (figura 7). Si embargo, hay que tener en cuenta que los alumnos con Síndrome de Down están acostumbrados a trabajar dependiendo de un profesor lo que se vio incrementado si tenían problemas con el manejo del ratón o si no entendían lo que el agente pedagógico les estaba pidiendo que hicieran para resolver la actividad [Aguilar 2005d] .

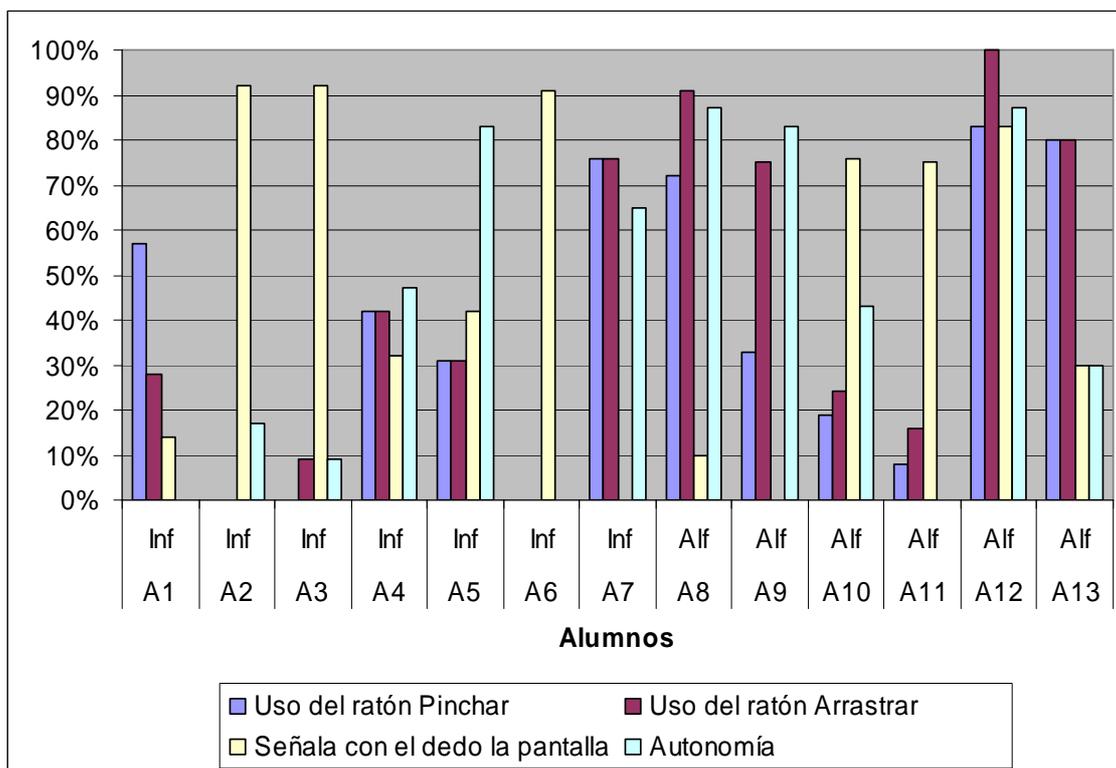


Figura 4-7. Resultados de los alumnos en el uso del ratón.

Para realizar la validación de la fase 2 del Tutorial se seleccionó una secuencia de actividades correspondientes a diferentes objetivos numéricos. De esta forma, se seleccionaron 37 actividades del Tutorial repartidas como se indica en la Tabla 4-6. La descripción de estas actividades se puede ver en el Apéndice A.

Tipos de Actividades	Nivel Poco	Nivel Medio	Nivel Alto	Total de Actividades
Contar y Reconocimiento del Número	2	2	2	6
Cardinalidad	5	5	6	16
Orden y Ordinalidad	1	4	2	7
Problemas	-	4	4	8

Tabla 4-5. Tipos de actividades fase 2.

De los 10 alumnos participantes, 7 de ellos estaban integrados en aulas de Infantil, Primaria y Secundaria, según su edad, y 3 eran alumnos de Alfabetización. Para poder analizar el grado de conocimiento que muestran con el tutorial, se dividieron los alumnos en tres grupos:

- Inf: Alumnos de infantil: A1, A2, A3 y A4, siguen sólo currículo de infantil y están escolarizados.
- Prim: Alumnos de primaria: A5, A6 y A7, siguen el currículo de infantil-primaria y están escolarizados.
- Alf: Alumnos de alfabetización: A8, A9 y A10, son jóvenes no escolarizados, cuya trayectoria educativa ha sido irregular, ya que no han estado escolarizados todos los años de su infancia, siguen el currículo de infantil- primaria o sólo infantil.

En la Figura 4-8 se recogen los porcentajes de éxito en el uso del ratón, tanto al pinchar como al pinchar y mover (se considera un éxito, si el alumno usa el ratón sin la ayuda del profesor en ningún momento de la actividad). También se indica el porcentaje de actividades en las que los alumnos señalan en la pantalla para dar la respuesta a las actividades. Otro aspecto analizado es si el alumno entiende la acción a realizar para resolver las actividades. Por ejemplo, si entiende que debe unir elementos de diferentes columnas, arrastrar elementos para colocarlos en una serie de recuadros, o si lo que tiene que hacer es pinchar en un elemento. Este

aspecto sobre la comprensión de la acción a realizar, es fundamental, ya que los alumnos con Síndrome de Down, suelen repetir automáticamente procesos aprendidos de memoria [Aguilar 2006]. Hemos decidido valorar este aspecto en la fase 2, ya que el hecho de que la acción a realizar para resolver las actividades cambie de una a otra, podría ser, a priori, un condicionante en el éxito del Tutorial para este tipo de alumnos (figura 4-8).

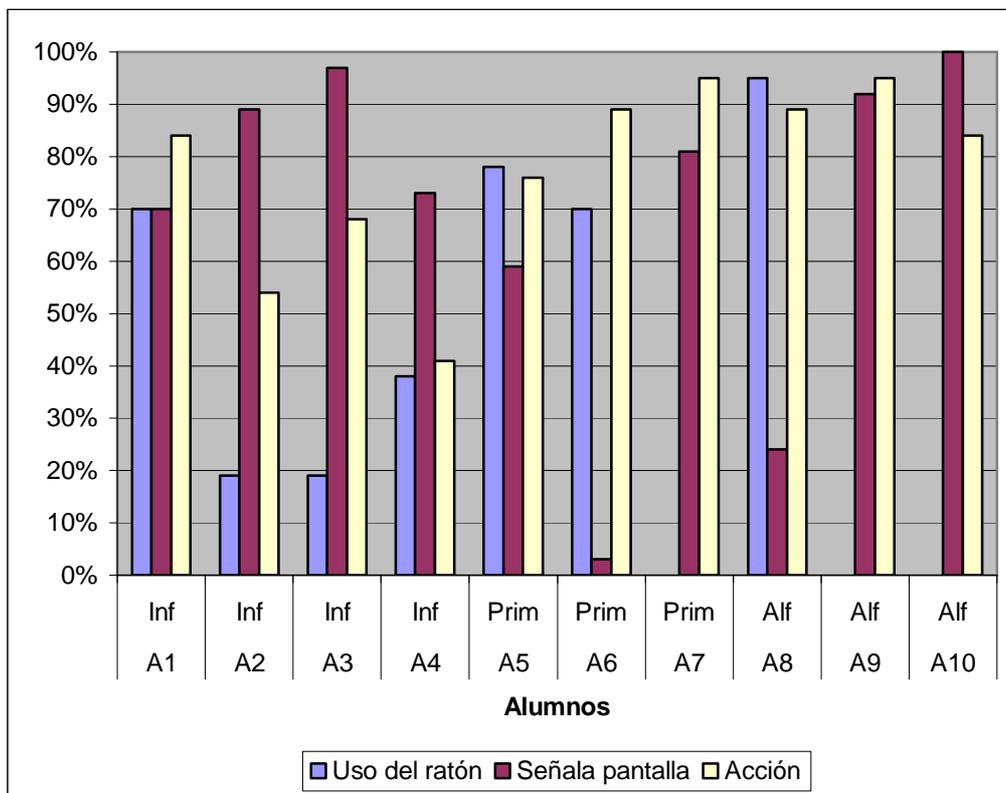


Figura 4-8. Resultados de los alumnos en el uso del ratón y comprensión de la actividad.

Los resultados sobre el uso del ratón muestran que ningún alumno tuvo en la fase 2 un 100% de éxito al pinchar y mover: todos han requerido en algún momento de las actividades la ayuda del profesor. Sin embargo se observan grandes diferencias entre los alumnos. Algunos alumnos mostraron un alto grado de precisión y autonomía, otros tuvieron problemas en algunas actividades y otros en cambio necesitaron ayuda del profesor en todo momento, el cual le facilitaba el movimiento del ratón o le ayudaba a pinchar, una vez que el alumno había indicado verbalmente cuál era el movimiento que quería realizar. Como ya comentamos en

la fase 1, la falta de éxito en el uso del ratón se puede deber al poco uso del ordenador en el ambiente familiar o escolar, y a las dificultades motrices propias de las personas con Síndrome de Down. Al igual que en la fase anterior, otro aspecto observado es la tendencia general de señalar en la pantalla del ordenador, tanto si dominan el ratón, como si no. Como era de esperar, los alumnos que tienen menos tendencia a señalar sobre la pantalla son los que tienen mayor control en el uso del ratón.

Con respecto a la comprensión de la acción, los resultados son óptimos, pues la mayoría de los alumnos han comprendido la acción a realizar en casi todas las actividades propuestas, manifestando más dudas con respecto a los conceptos matemáticos que a los tecnológicos. De hecho, los resultados más bajos corresponden a dos alumnos del grupo de infantil (A2 y A4) con un 54% y 41 % de éxito, respectivamente. El resto de los alumnos tienen altos porcentajes de éxito.

Otro aspecto no analizado en la validación de la fase 1 y que si se ha valorado en esta fase 2, fue la concentración de los alumnos frente al Tutorial, ya que es evidente que el interés y la atención que los alumnos muestran al realizar las actividades pueden hacer que sus respuestas tengan mayor o menor éxito. En la figura 4-9 se muestran los porcentajes de actividades en las que los alumnos se mostraron concentrados y atentos en todo momento de la realización de la actividad. El interés y grado de atención al realizar las actividades del Tutorial muestran una motivación importante por parte de los alumnos, hecho observado en la concentración al realizar las actividades, las expresiones de satisfacción cuando el agente los felicitaba, y el interés en seguir realizando actividades una vez concluidas las sesiones de trabajo. De hecho, los alumnos de los grupos de Primaria y Alfabetización mostraron una alta concentración y mucho interés en las sesiones de trabajo. Frente a ellos, los alumnos del grupo de Infantil se mostraron, en algunos momentos de las actividades, distraídos o cansados, debido a que el esfuerzo para resolver las actividades para ellos era superior, bien por los problemas motrices o conceptuales.

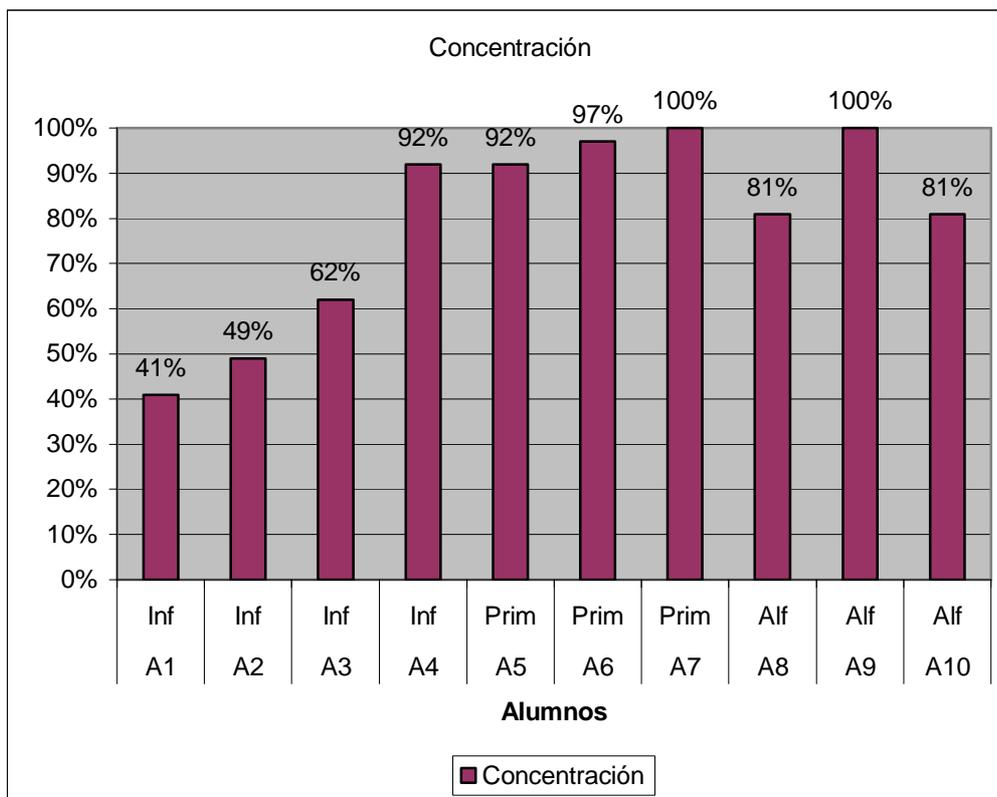


Figura 4-9. Resultados de los alumnos en concentración.

En definitiva, el análisis de la fase 2 muestra algunas cuestiones ya observadas en la fase 1. En primer lugar, hay una gran variabilidad entre los alumnos en el manejo del ordenador en función de si lo usan en mayor o menor medida en otros ambientes, escolares o familiares. De nuevo, encontramos las dificultades motrices que provoca el manejo del ratón, lo cual hay que entenderlo no como un problema del propio Tutorial, sino como una dificultad de las personas con esta discapacidad.

Es importante resaltar dos aspectos positivos en la validación de la fase 2 que no se observaron en la fase 1. Por un lado, los altos porcentajes de éxito en cuanto a la comprensión de la acción para realizar las actividades, lo que demuestra que éstas son intuitivas para los alumnos y se relacionan con las que habitualmente hacen en papel y lápiz, lo cual lleva a que no se producen bloqueos frente al ordenador. Por otro lado, un aspecto muy valorado por los tutores de los alumnos, es la alta motivación frente a la realización de las actividades del Tutorial.

Conclusiones

Se ha presentado en esta Tesis Doctoral el diseño e implementación de un Sistema Tutorial Inteligente (ITS) que combina el uso de metodologías Borrosa y Multiagente para el refuerzo de los conceptos lógicos de número, suma y resta, en el contexto de un currículo español de infantil, aunque es posible generalizarlo a cualquier dominio. Para ello contamos con la colaboración de miembros del Departamento de Análisis Matemático (área de Didáctica de las Matemáticas) de la Universidad de La Laguna que han aportado sus conocimientos para hacer posible la adquisición, análisis y modelado del conocimiento experto necesario para la realización del ITS.

En primer lugar se ha realizado un estudio de la influencia de las nuevas tecnologías en la educación, especialmente de los beneficios de la inclusión del ordenador en el aula.

El ITS que se ha diseñado consta de tres partes fundamentales: La Interfaz Gráfica, El Módulo Tutor y bases de datos.

La **Interfaz Gráfica** es la encargada de controlar el flujo de comunicación entre el ITS y el alumno. Para atraer, motivar y retener la atención del estudiante se han creado plantillas usando tecnología XML con Javascript. Esto permite que las actividades que se le presentan al alumno se visualicen en un navegador y no sea necesaria la instalación ni el aprendizaje de software adicional por parte de profesores ni alumnos. Dichas plantillas incorporan agentes pedagógicos que se encargan de interactuar con los alumnos y explicarles la actividad que tienen que realizar. De esta forma, se encargan de presentar el problema, guiar la ejecución de la actividad y presentar estímulos positivos y/o negativos (feedbacks) de acuerdo a los resultados obtenidos por el alumno en la resolución de la actividad. Esto provoca que la motivación del alumno se vea incrementada. Se ha desarrollado un Generador de Actividades para que los profesores puedan crear sus propias actividades para el ITS. El programa generador, que utiliza tecnología Apache Cocoon, sirve como asistente en la creación de plantillas multimedia de forma fácil y amigable. Esta tecnología es un sistema de publicación web basado en XML/XSL que cuenta con un desarrollo en Java. El uso de dicha tecnología permite que la generación de actividades para la inclusión en el ITS sea accesible vía web.

Así cualquier profesor se puede conectar a un servidor y generar las actividades que desee de forma remota o hacerlo localmente en su máquina si lo prefiere.

El Módulo Tutor: El uso de la combinación de la Lógica Borrosa más los Sistemas Multiagente para la realización del Planificador Instruccional es otra aportación novedosa de esta Tesis Doctoral. Dicho Planificador Instruccional permite tomar decisiones en el ITS sobre el nivel de complejidad en el que debe trabajar el alumno, en base a los resultados obtenidos por el mismo en todos los objetivos en los que se está trabajando de forma simultánea. Para el desarrollo del mismo se ha utilizado lógica borrosa teniendo en cuenta las características de incertidumbre del conocimiento pedagógico a modelar. Se basa en reglas del tipo: *"Si los resultados del alumno son muy bajos y su evolución es lenta entonces hay que bajar el nivel de dificultad"*, puesto que la mayoría de los expertos no piensan normalmente en términos de probabilidad sino en términos tales como mucho, poco, bien, mal, etc. Debido a los resultados obtenidos en las primeras pruebas realizadas sobre un prototipo, se vio la necesidad de utilizar técnicas adicionales en la elección del siguiente objetivo a tratar. Para ello se estudió la metodología basada en Sistemas Multiagente y se adaptó la misma a nuestras necesidades. Dicho Sistema Multiagente determina cuál es el objetivo de enseñanza (y tipo de actividad dentro del objetivo) en el que tiene que trabajar el alumno. Gracias a la incorporación de esta técnica conseguimos mejorar la estrategia de enseñanza de los objetivos y a la vez disminuir el tiempo de espera en la interacción con el usuario.

Las **bases de datos** del modelo del estudiante y del conocimiento pedagógico: Los objetivos de aprendizaje se almacenan en la base de datos de actividades que cuenta con un gran número de actividades generadas con las plantillas para que haya suficiente variedad y mantener la atención y motivación de los alumnos. En la base de datos del modelo del estudiante se guardan los datos del mismo (datos personales y del tipo al que pertenece) y el historial del alumno que le sirve a los profesores para tener un seguimiento de los trabajos realizados.

Se ha llevado a cabo la **verificación y validación del ITS** que ha sido realizada tanto con expertos en el dominio matemático como por profesores de otras áreas de conocimiento. Puesto que en el diseño de nuestro ITS se ha usado una metodología incremental el sistema se dividió en tres subsistemas principales: el Sistema Borroso, el Sistema Multiagente y la Interfaz Gráfica. Cada uno de estos

subsistemas fue verificado por separado para posteriormente realizar una verificación del sistema completo y comprobar el funcionamiento correcto de la unión de todas las partes. Se ha realizado también la verificación del modelo lógico del sistema mediante la definición y comprobación de los casos de uso del mismo. La validación del ITS ha sido realizada utilizando tanto los métodos de validación cuantitativos como cualitativos y de usabilidad de la interfaz gráfica. La validación cualitativa se realizó con alumnos de la Asociación de Trisómicos 21 (ATT21) que siguen el currículo de diferentes niveles educativos gracias a un "Convenio Marco entre la Asociación Tinerfeña de Trisómicos 21 y la Universidad de La Laguna para la realización de actividades de interés común".

Actualmente el ITS está siendo traducido al portugués y usado en la Universidad Luterana de Brasil (ULBRA) en función a los convenios de colaboración surgido entre dicha Universidad y el departamento de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadores de la Universidad de La Laguna. Estos convenios son: un "Convenio Marco entre la Universidad Luterana de Brasil y la Universidad de La Laguna" y un Proyecto de Cooperación Iberoamericana (PCI) denominado: "Formación Profesional a distancia en Matemáticas y Nuevas Tecnologías para Enseñanzas Técnicas con desarrollo de Nuevas Herramientas Tecnológicas (D/6185/06)". Además se ha firmado un "Convenio Marco entre la Federación Iberoamericana de Sociedades de Educación Matemática y La Universidad de la Laguna para la realización de actividades de investigación y otras actividades de interés común".

Igualmente el Generador de Actividades se utilizará como herramienta en asignaturas de la Facultad de Educación de la Universidad de La Laguna gracias a la colaboración con el Departamento de Análisis Matemático (área de Didáctica de las Matemáticas) de dicha Universidad. Con ello se conseguirá tener un mayor número de actividades que enriquecerán el abanico de actividades a presentar al alumno en el ITS.

Un trabajo futuro consistirá en la generación automática de actividades. Esto es, dada una base de datos con distintos elementos multimedia, crear un sistema inteligente capaz de crear automáticamente actividades siguiendo las plantillas ya definidas.

APÉNDICE A. TIPOS DE ACTIVIDADES

En este apéndice se presenta la descripción y ejemplos de todos los tipos de actividades que se han creado para el ITS.

1 EJERCICIOS FASE 1 (LÓGICA)

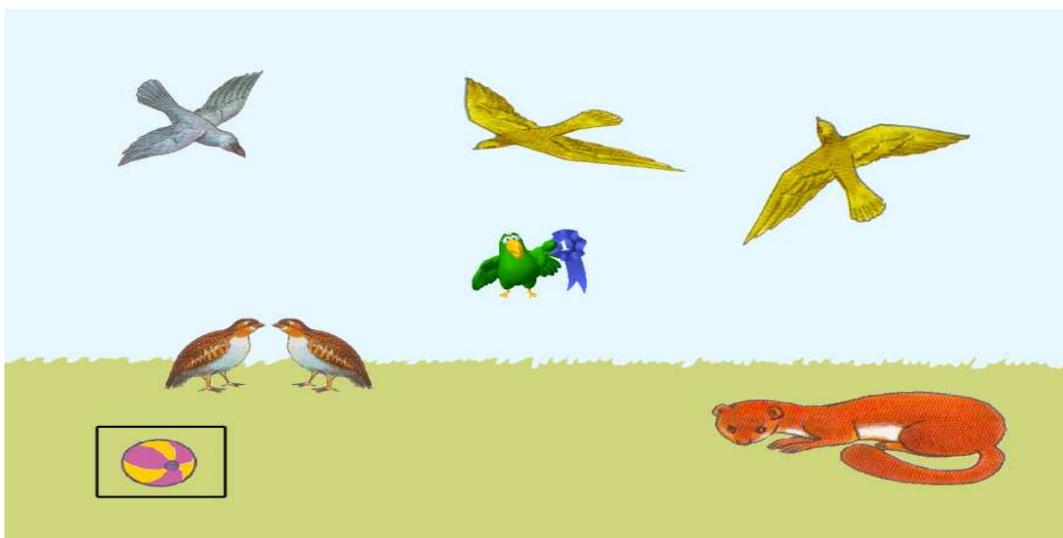
1.1 Clasificación

1.1.1 NIVEL BAJO

1.1.1.1 Actividad Tipo 1

Objetivo: Reconocer las características de un conjunto y separar elementos que no pertenezcan a él.

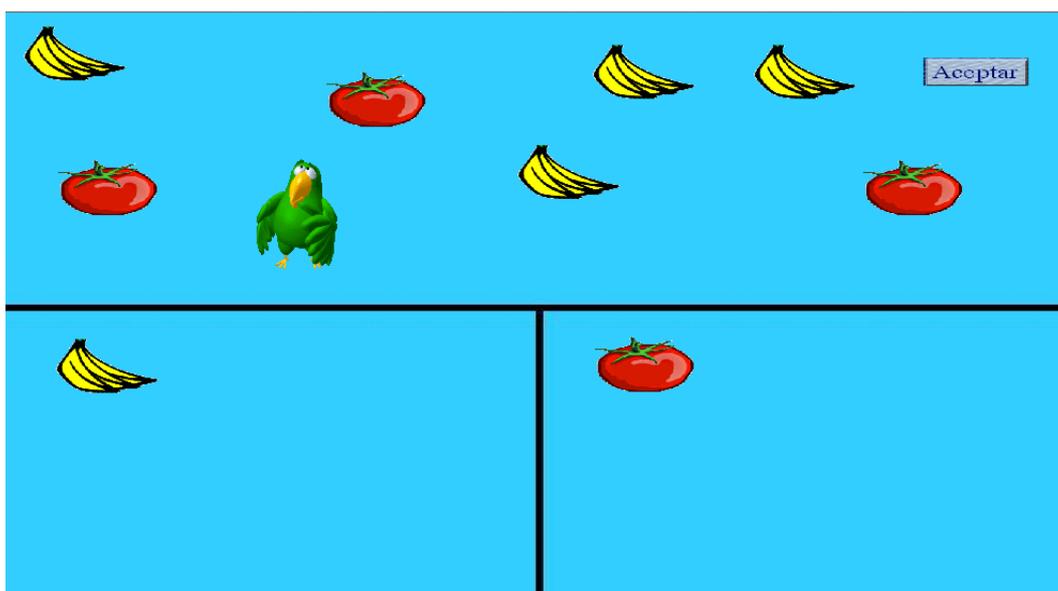
Ejemplo: En la figura vemos un grupo de animales y una pelota, por lo que el elemento que no pertenece al conjunto es la pelota. Al pinchar sobre ella, se marca con un recuadro y el agente pedagógico nos felicita, indicándonos que hemos realizado correctamente el ejercicio.



1.1.1.2 Actividad Tipo 2

Objetivo: Agrupar objetos idénticos entre sí.

Ejemplo: Esta actividad, como se puede observar en la figura, tiene como objetivo agrupar en cada uno de los recuadros de la parte inferior de la pantalla aquellas imágenes que sean iguales a las indicadas en estos recuadros. En el ejemplo, los plátanos se colocarían en la parte inferior izquierda de la pantalla y los tomates en la inferior derecha. Una vez que se han colocado todos, hay que pulsar en Aceptar para que se realice la validación del ejercicio. Si es correcto, se pasa al siguiente ejercicio, y si no, se vuelven a colocar los objetos en su posición inicial y se da una nueva oportunidad para realizar el ejercicio.

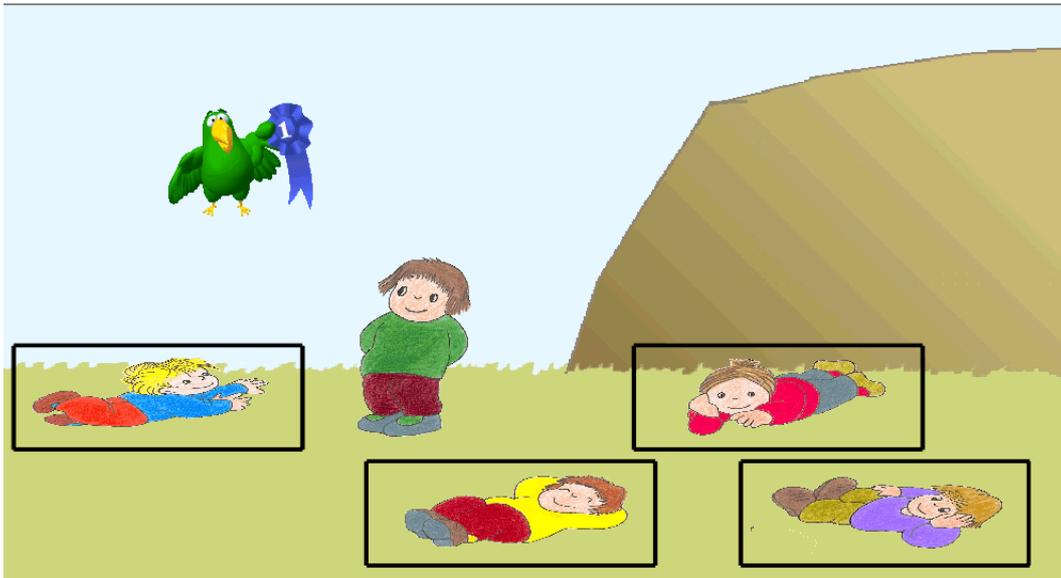


1.1.2 NIVEL ALTO

1.1.2.1 Actividad Tipo 3

Objetivo: Elegir objetos opuestos a otros.

Ejemplo: Como se puede ver en la figura hay que elegir a los niños que están tumbados frente al que está de pie. Para ello, se pincha sobre los elementos a seleccionar, los cuales quedan marcados mediante un recuadro de color negro.



1.1.2.2 Actividad Tipo 4

Objetivo: Elegir objetos con alguna relación y que no sean idénticos.

Ejemplo: La realización es igual que las actividades de clasificación nivel poco tipo 2, pero los objetos se agrupan en función de clases lógicas. En la figura hay que agrupar los objetos que sean herramientas en el recuadro de la parte inferior izquierda de la pantalla, y los muebles en la parte inferior derecha.



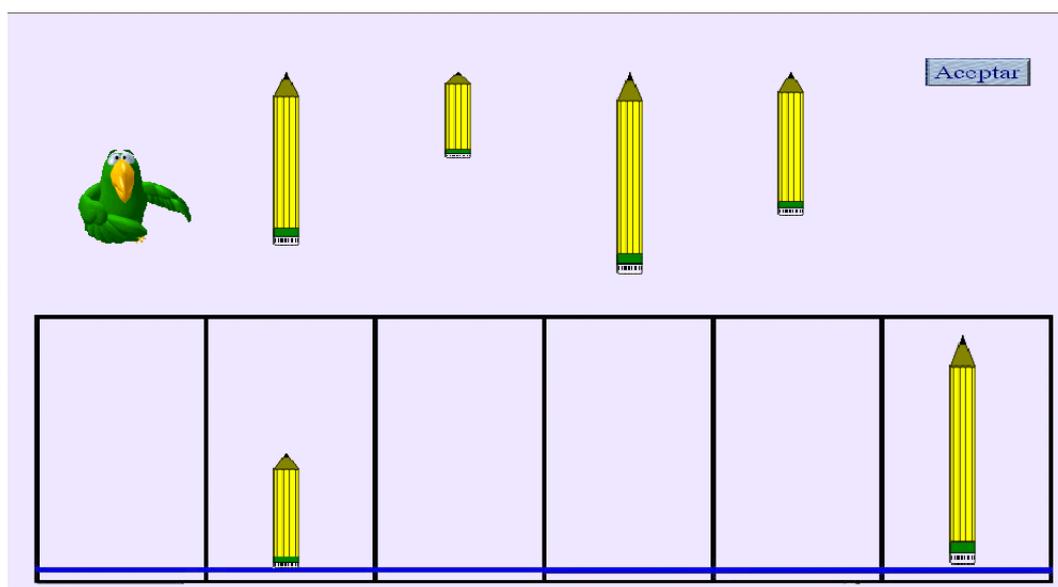
1.2 Relaciones de Orden

1.2.1 NIVEL BAJO

1.2.1.1 Actividad Tipo 1

Objetivo: Hacer seriaciones simples de mayor a menor, y viceversa.

Ejemplo: En la actividad que se ve en la figura hay que ordenar los lápices, colocándolos en el lugar que les corresponda de menor a mayor. La marca azul que se puede observar en la figura se ha puesto como referencia base para que el niño pueda medir y comparar las alturas.

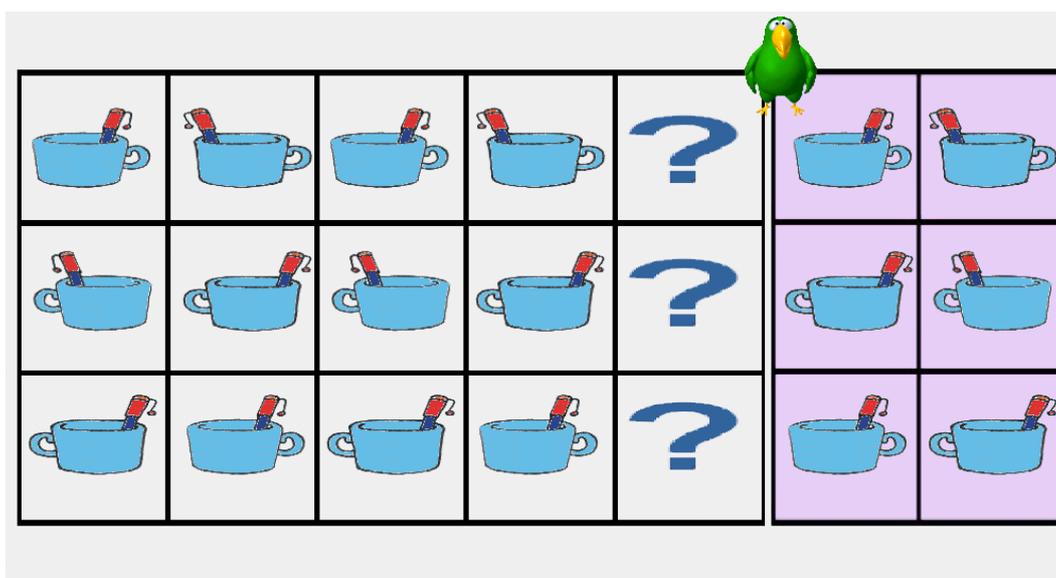


1.2.1.2 Actividad Tipo 2

Objetivo: Hacer seriaciones con alternancia de dos elementos y una incógnita.

Ejemplo: Como se ve en la figura el niño tiene que completar la serie colocando en el lugar de la incógnita el dibujo en el que la taza y el bolígrafo estén en el lugar correcto según la serie. Para ello, tiene que pinchar primero en el interrogante, y luego en uno de los dibujos situados en la parte derecha de la

pantalla (sobre fondo malva). Si el resultado es correcto, el interrogante es sustituido por la figura en cuestión, y el agente felicita al niño. En el caso contrario, no se produce ninguna modificación en las imágenes de la pantalla, pero el agente realiza un feedback correctivo y se le da al niño una segunda oportunidad.

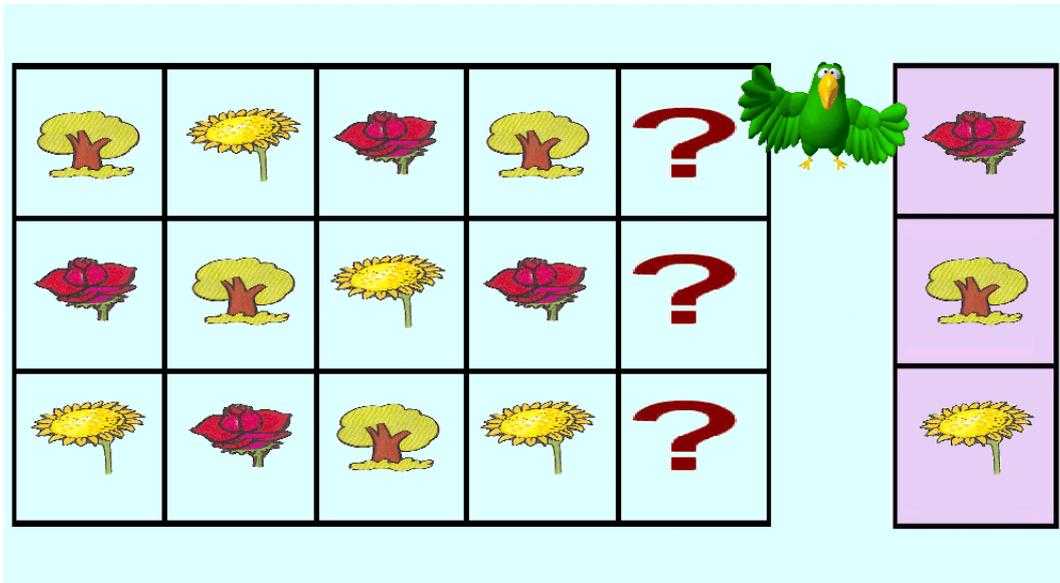


1.2.2 NIVEL ALTO

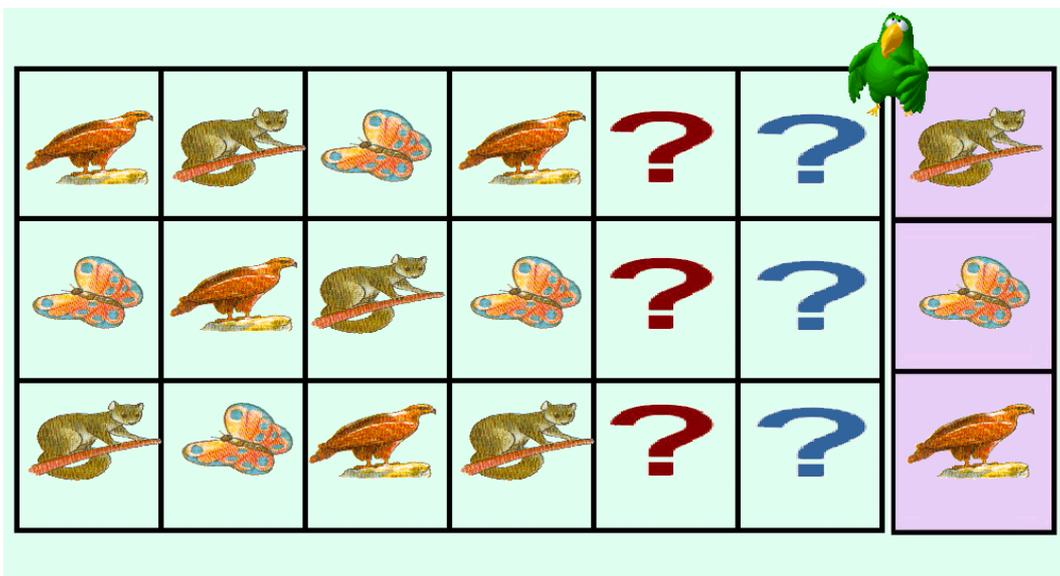
1.2.2.1 Actividad Tipo 3

Objetivo: Hacer seriaciones con alternancia de más de dos elementos y varias incógnitas.

Ejemplos: Para este tipo de actividad, existen dos variantes: la primera es hacer seriaciones con alternancia de tres elementos y una incógnita como se puede observar en la siguiente imagen.



La otra variación es hacer seriaciones con alternancia de tres elementos, pero teniendo dos incógnitas. La realización de ambas actividades es igual a la de nivel bajo, es decir, primero se tiene que pinchar en el interrogante, y luego en uno de los dibujos situados en la parte derecha de la pantalla (sobre fondo malva).



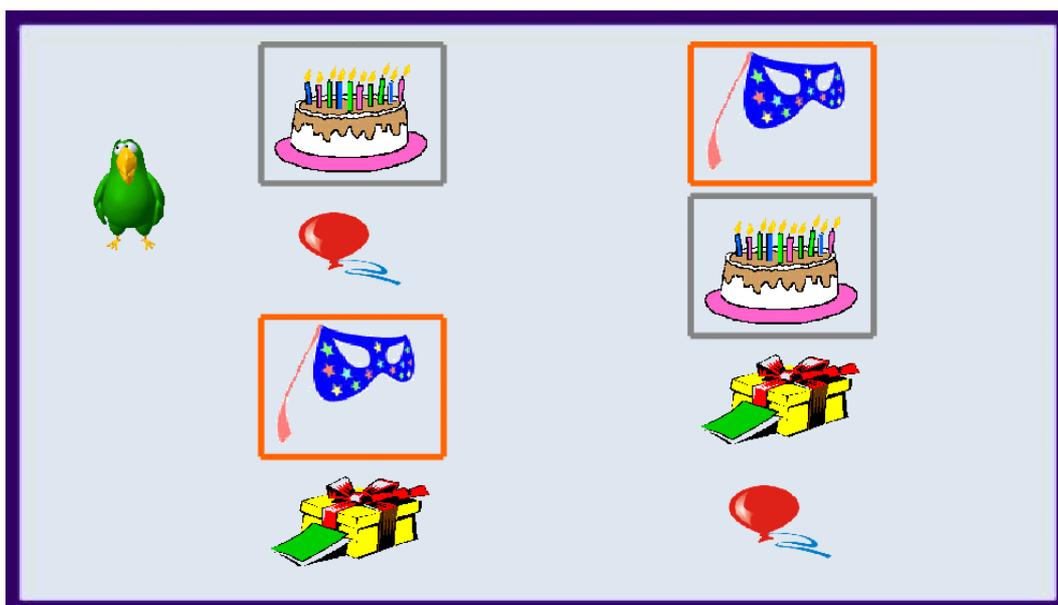
1.3 Correspondencia término a término

1.3.1 NIVEL BAJO

1.3.1.1 Actividad Tipo 1

Objetivo: Relacionar dos conjuntos de objetos con una relación de igualdad.

Ejemplo: Como se puede ver en la imagen, hay dos columnas con imágenes que se repiten, pero en distintas posiciones. El alumno tiene que emparejar las de la columna izquierda con las de la derecha. Si el alumno realiza correctamente el emparejamiento, se recuadran las parejas con un color y se da un feedback positivo. En caso contrario, no se marcan, se da un feedback correctivo y otra oportunidad.

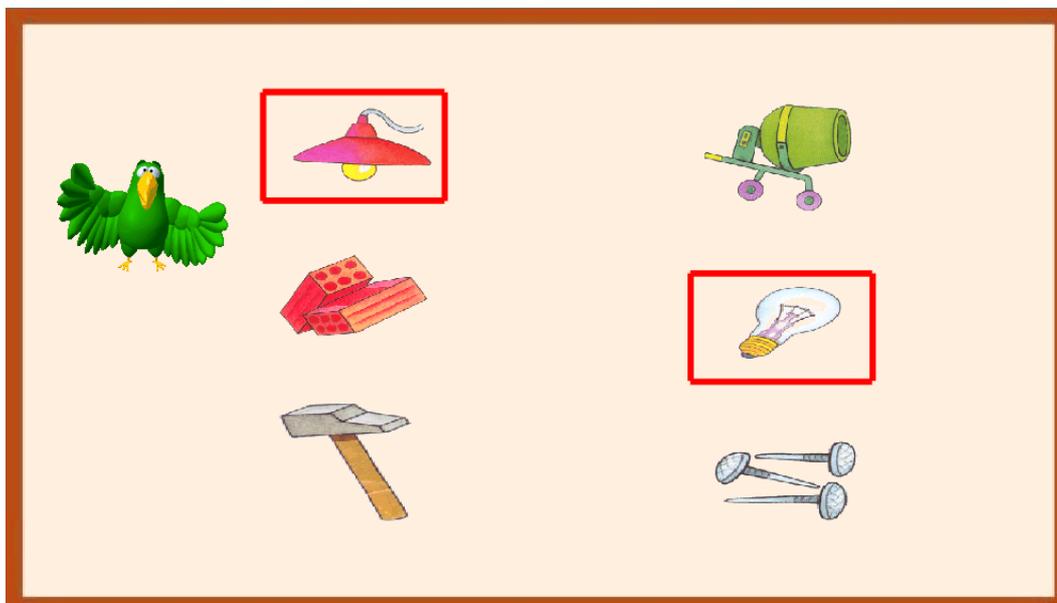


1.3.2 NIVEL ALTO

1.3.2.1 Actividad Tipo 2

Objetivo: Relacionar dos conjuntos de objetos con una relación que no implique igualdad.

Ejemplo: Las actividades son iguales a las del apartado anterior, pero las parejas se establecen entre objetos que no son iguales, pero que tienen alguna relación que los une.



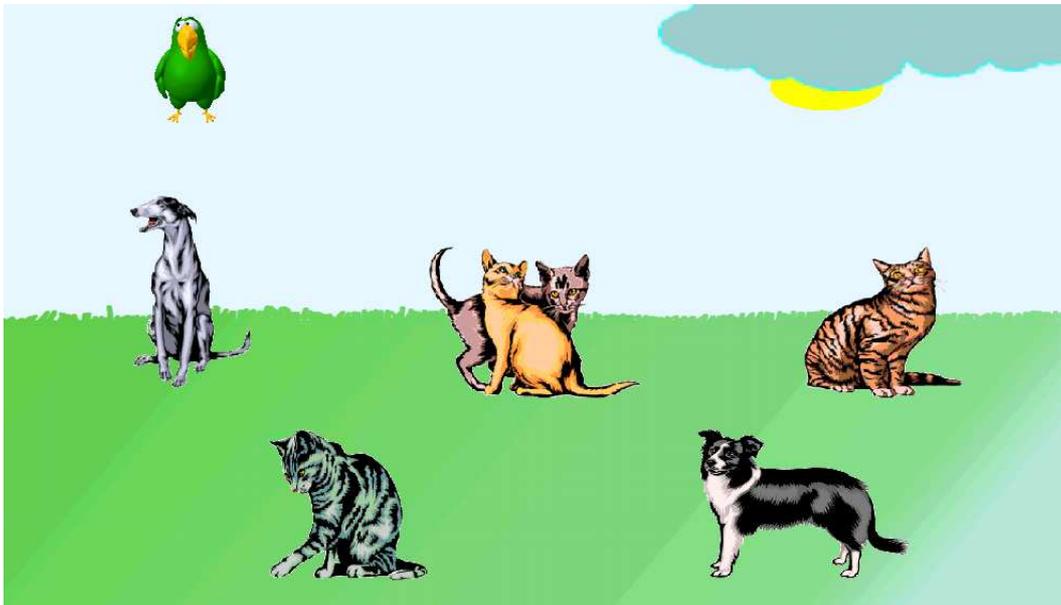
1.4 Cuantificadores

1.4.1 NIVEL BAJO

1.4.1.1 Actividad Tipo 1

Objetivo: Seleccionar todos los objetos de un conjunto que posean, o no, una determinada característica.

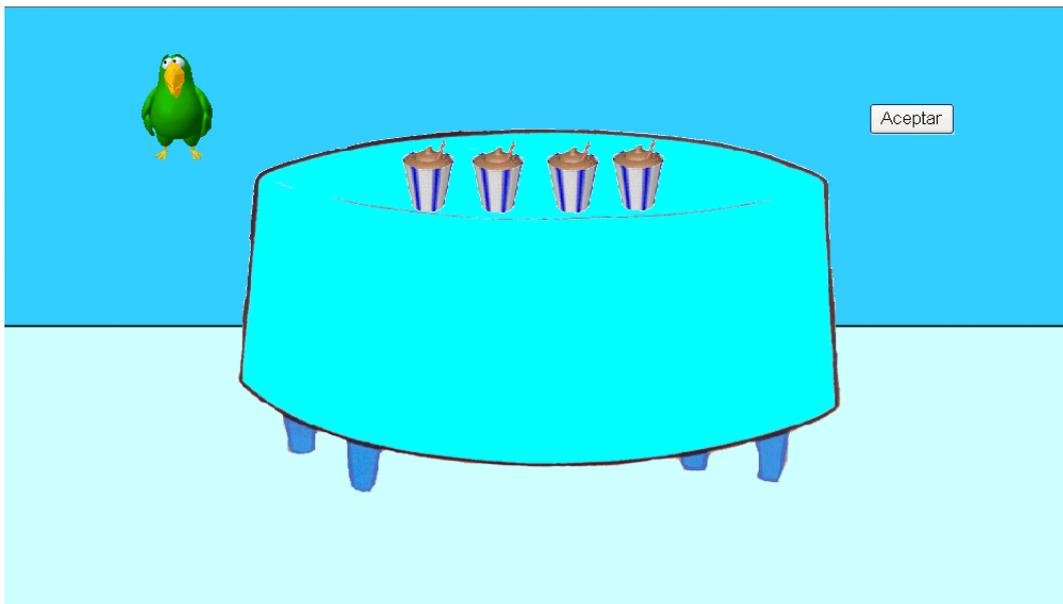
Ejemplo: En la figura podemos ver una actividad de este tipo, en la que hay que señalar todos los gatos. La misma actividad podría servir, en el caso de que así se indique, para señalar el caso contrario, es decir, todos los que no son gatos (o sea, los perros).



1.4.1.2 Actividad Tipo 2

Objetivo: Extraer elementos de un conjunto, hasta que no quede ninguno.

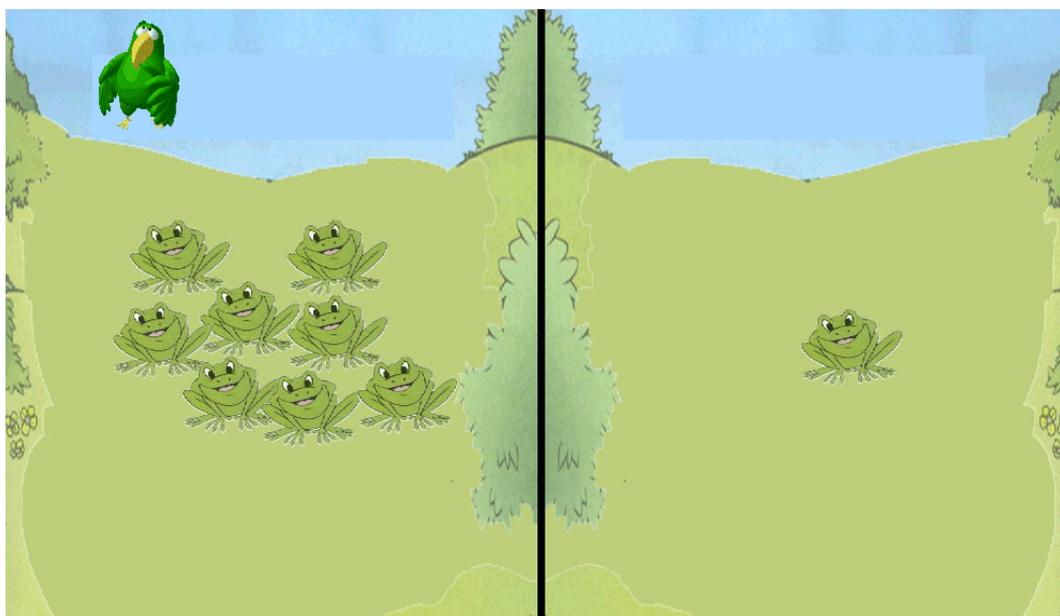
Un ejemplo lo tenemos en la figura en la que hay que quitar todos los helados de encima de la mesa hasta que no quede ninguno, y luego pulsar en el botón de Aceptar para validar el ejercicio.



1.4.1.3 Actividad Tipo 3

Objetivo: Indicar qué conjunto tiene muchos y/o pocos elementos, siendo éstos iguales.

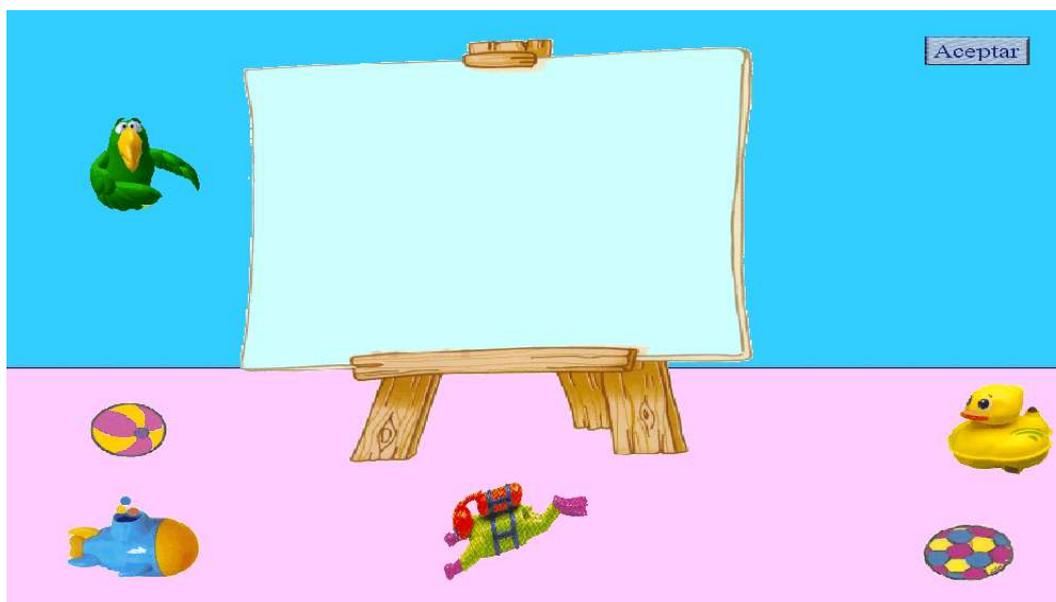
Ejemplo: Vemos en la figura una actividad de este tipo, en la que habría que señalar, por ejemplo en que parte de la pantalla hay muchas ranas. La misma actividad sirve para señalar por ejemplo, en qué parte hay pocas ranas.



1.4.1.4 Actividad Tipo 4

Objetivo: Seleccionar algunos objetos de un conjunto.

Ejemplo: En esta figura se puede observar una pizarra y unos cuantos juguetes distribuidos por el suelo. El objetivo es poner en la pizarra algunos de los juguetes que se encuentran en el suelo. Para que el ejercicio sea correcto, se tiene que colocar en la pizarra más de un juguete y menos que el total, es decir, en este caso se podrían colocar dos, tres o cuatro juguetes.

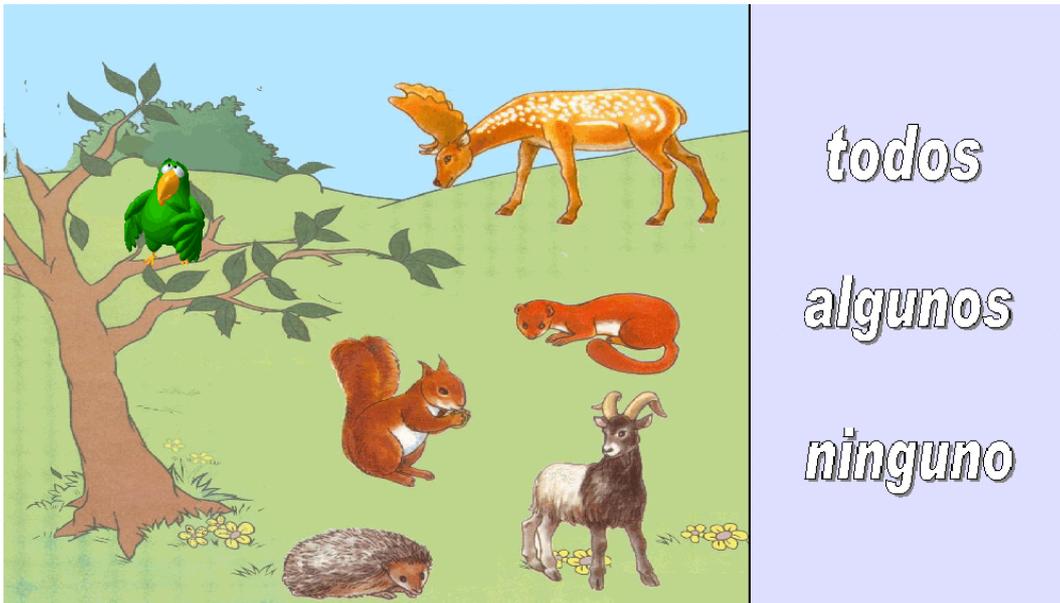


1.4.2 NIVEL ALTO

1.4.2.1 Actividad Tipo 1

Objetivo: Dado un conjunto de elementos, ver si todos los elementos poseen una determinada característica.

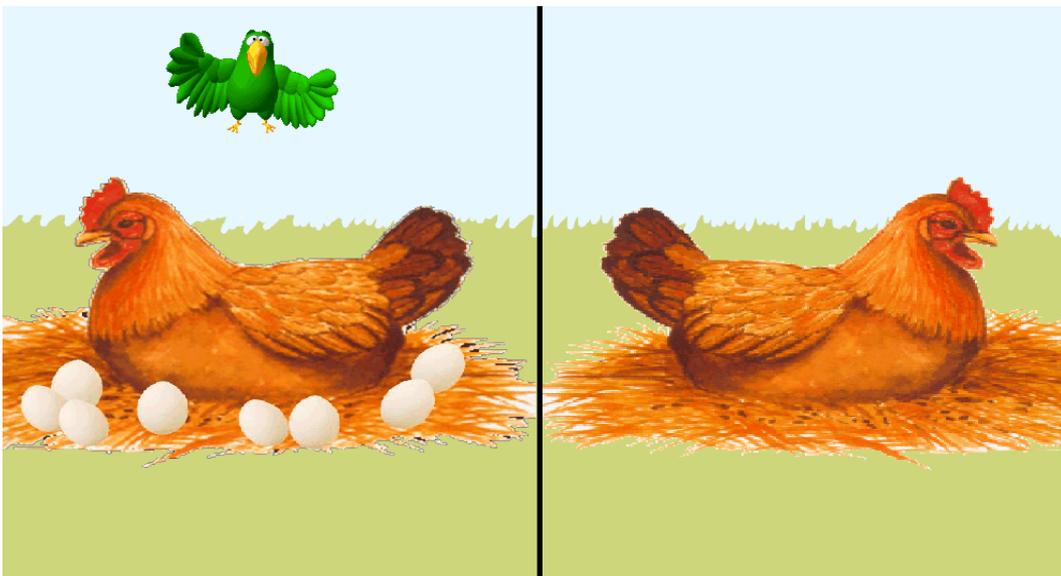
Ejemplo: Este tipo de actividad es igual que la de nivel bajo, pero con un nivel de complejidad mayor. Por ejemplo, en la figura se pregunta cuántos animales de los que aparecen en el dibujo tienen cuatro patas, y se dan tres opciones: todos, algunos y ninguno. El niño tendría que pinchar encima de la palabra que considere la respuesta correcta.



1.4.2.2 Actividad Tipo 2

Objetivo: Señalar en que imágenes no existe ningún elemento de los que se indica.

Ejemplo: En la actividad que se ve en la figura tenemos un ejemplo en el que hay que señalar cuál es la gallina que no tiene ningún huevo. Para ello, en este caso, habría que pinchar en la imagen correspondiente al lado derecho de la pantalla.



1.4.2.3 Actividad Tipo 3

Objetivo: Indicar qué conjunto tiene muchos y/o pocos elementos, estando éstos relacionados de alguna forma, pero no siendo iguales.

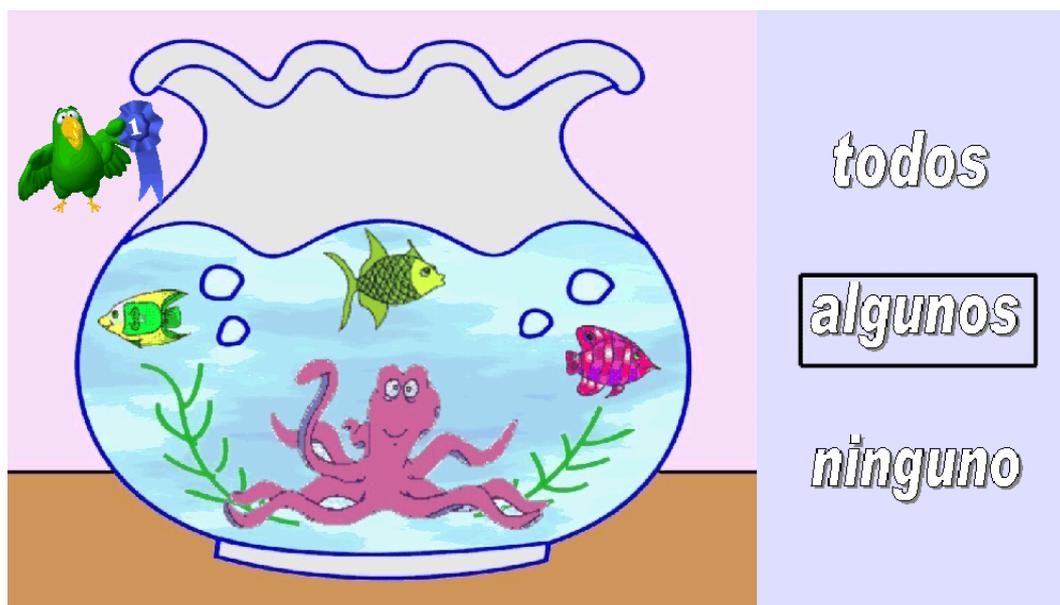
Ejemplo: Este tipo de actividad es igual que la de nivel bajo, pero aumenta el nivel de complejidad, puesto que ahora los elementos que hay que comparar no son iguales. Vemos en la figura una actividad de este tipo, en la que habría que señalar la parte de la pantalla con pocos juguetes.



1.4.2.4 Actividad Tipo 4

Objetivo: Dado un conjunto de elementos, ver si algunos de estos elementos poseen una determinada característica (no todos).

Ejemplo: Por ejemplo en la figura se pregunta cuántos de los animales que se encuentran en la pecera tienen aletas. Como en el caso de la actividad de tipo 1, de nivel Alto se le dan tres opciones: todos, algunos y ninguno, y el niño tendrá que pinchar encima de la palabra que considere que es la respuesta correcta.



2 EJERCICIOS FASE 2 (NÚMEROS Y OPERACIONES)

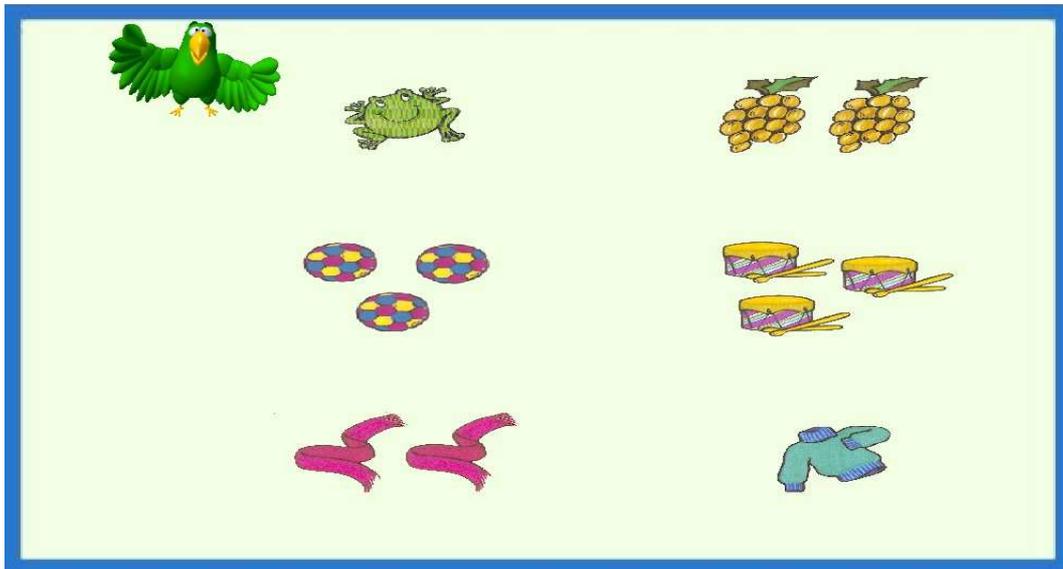
2.1 Contar

2.1.1 NIVEL BAJO

2.1.1.1 Actividad Tipo 1

Objetivo: Contar objetos hasta el número tres.

Ejemplo: Como se puede observar en la figura inferior, hay dos columnas con colecciones que tienen un número variable de elementos (hasta tres). El alumno tiene que emparejar, las de la columna izquierda, con las de la derecha. Si el alumno realiza correctamente el emparejamiento se recuadran las parejas con un color, y se da un feedback positivo. En caso contrario, no se marcan, se da un feedback correctivo y otra oportunidad.



2.1.1.2 Actividad Tipo 2

Objetivo: Completar una serie en la que faltan números (con números hasta el número tres).

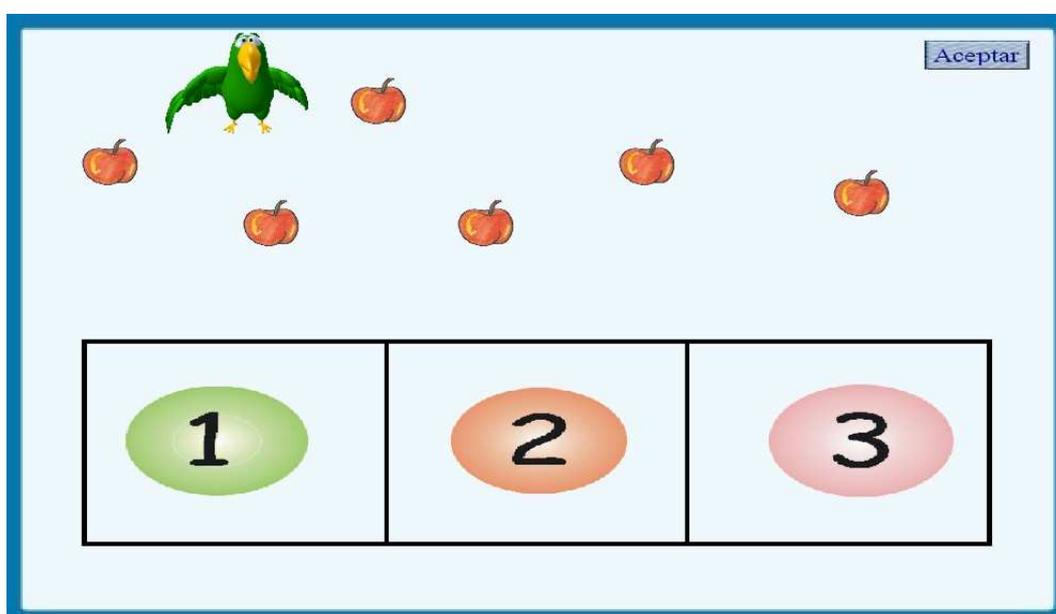
Ejemplo: En la actividad que se ve en la figura hay que ordenar los números de menor a mayor colocándolos en el lugar que les corresponda, dentro del recuadro negro.

An interactive activity interface with a light purple background. In the top left corner is a green parrot icon. In the top right corner is a button labeled "Aceptar". In the center, there are three large numbers: a yellow "1", a pink "2", and a blue "3". Below the numbers is a horizontal row of three rectangular boxes. The first two boxes are empty, and the third box contains the number "3".

2.1.1.3 Actividad Tipo 4

Objetivo: Construir la serie numérica mediante la adición de la unidad (hasta el número tres).

Ejemplo: En la actividad que nos sirve de ejemplo, hay que llevar a cada recuadro tantos melocotones como indique el número. Es decir, habría que colocar un melocotón sobre la figura del número 1, dos sobre la del número 2, y tres sobre la del 3. Como la dificultad del ejercicio es baja, el número de melocotones es el exacto para la resolución correcta del ejercicio (no sobran melocotones).

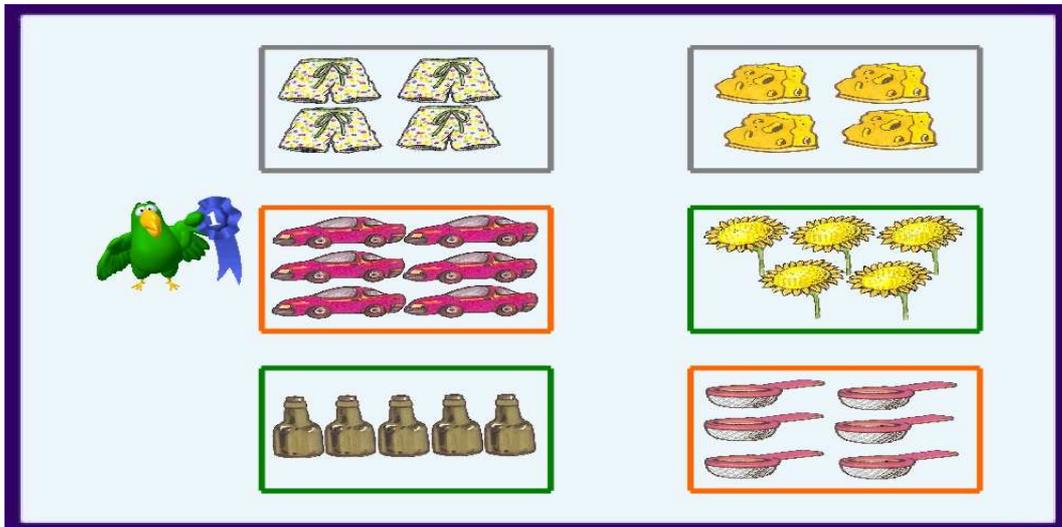


2.1.2 NIVEL MEDIO

2.1.2.1 Actividad Tipo 1

Objetivo: Contar objetos hasta el número seis.

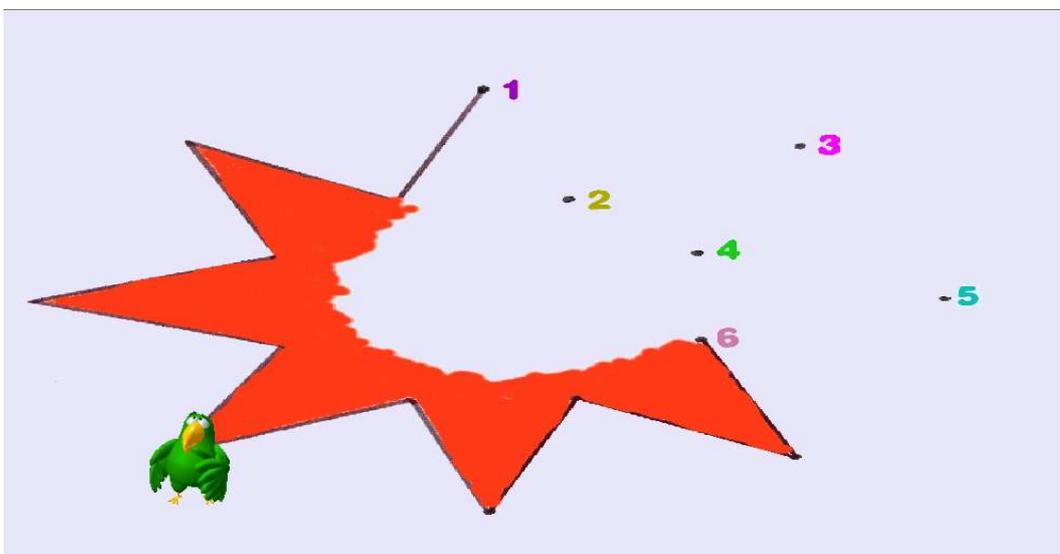
Ejemplo: Es igual que el ejercicio de nivel bajo, pero el número de objetos en las colecciones puede llegar hasta seis.



2.1.2.2 Actividad Tipo 2

Objetivo: Unir puntos de una serie (con números hasta el número seis).

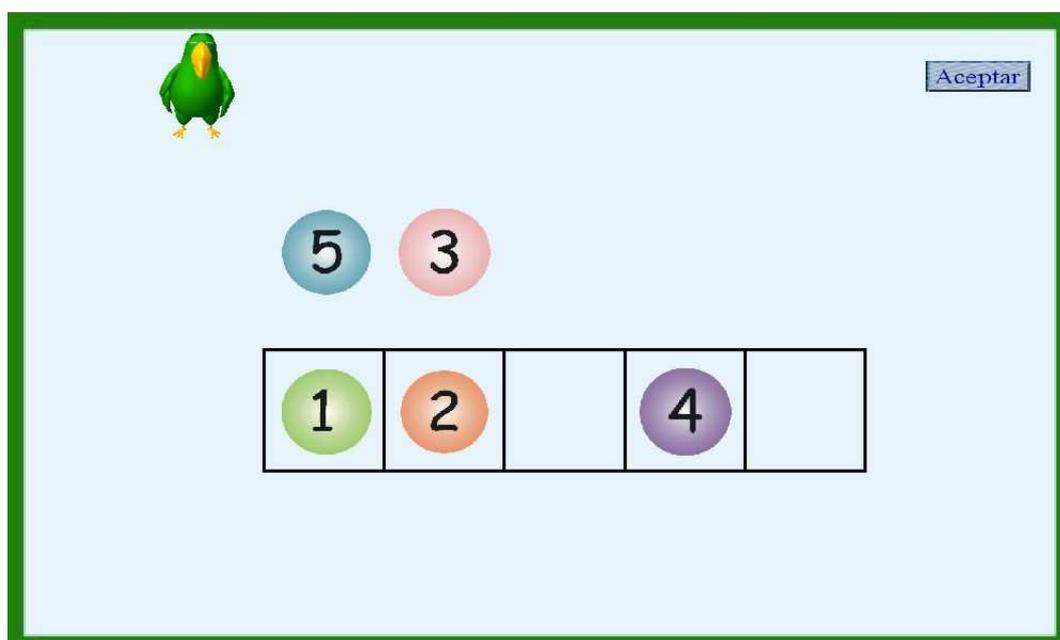
Ejemplo: En este ejercicio hay que ir pinchando en orden los números para completar el dibujo. La resolución correcta es pinchar, en primer lugar, sobre el número 1 (que es el que indica el comienzo de la parte del dibujo que falta), luego en el 2, y así sucesivamente, hasta el 6 para cerrar el dibujo. Las líneas irán apareciendo a medida que se pinchen en los números. Cuando esté completo, el agente felicita y se pasa al siguiente.



2.1.2.3 Actividad Tipo 3

Objetivo: Completar una serie en la que faltan números (con números hasta el seis, incluido el cero).

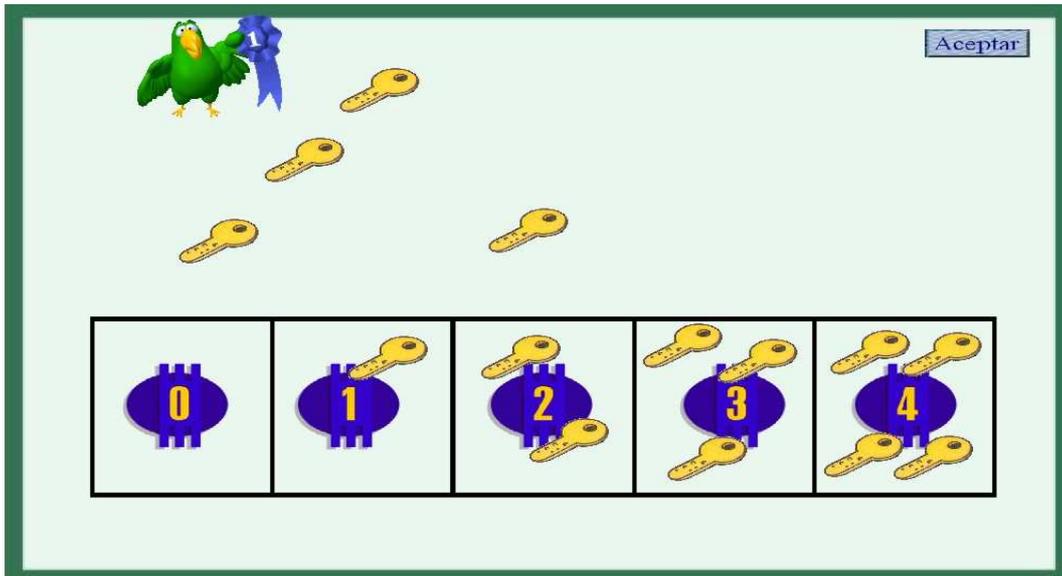
Ejemplo: Las actividades son iguales que las de nivel bajo, pero hasta el número seis, y pudiendo incluir el cero.



2.1.2.4 Actividad Tipo 4

Objetivo: Construir la serie numérica mediante la adición de la unidad (con números hasta seis, incluido el cero).

Ejemplo: En la actividad que nos sirve de ejemplo, hay que llevar a cada recuadro tantas llaves como indique el número, como se ve en la figura. El número de llaves es mayor que las necesarias para aumentar la dificultad.

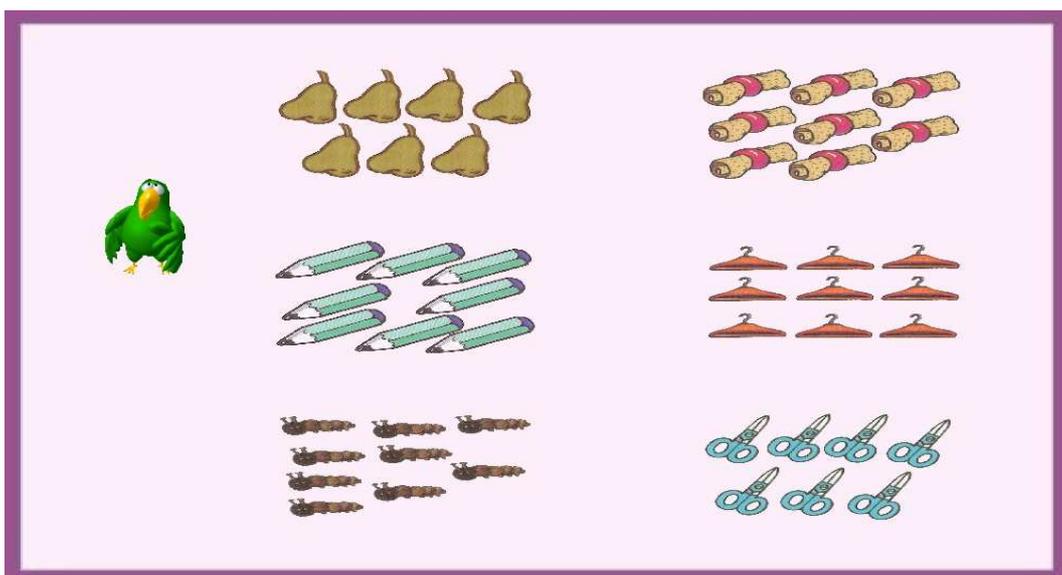


2.1.3 NIVEL ALTO

2.1.3.1 Actividad Tipo 1

Objetivo: Contar objetos hasta el número nueve.

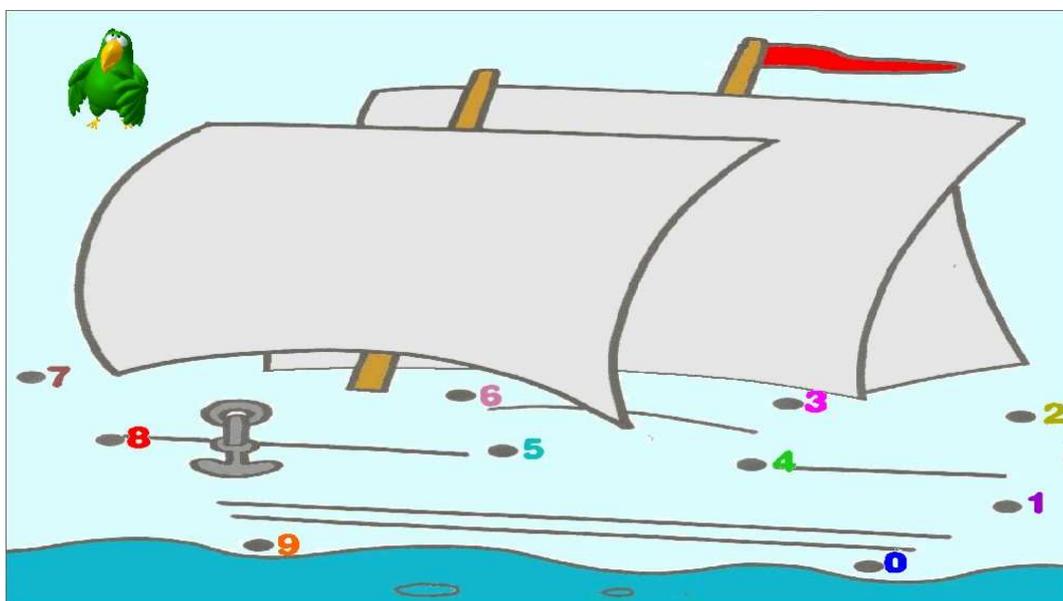
Ejemplo: Igual que en los niveles bajo y medio, pero el número de objetos en las colecciones llega hasta nueve.



2.1.3.2 Actividad Tipo 2

Objetivo: Unir puntos de una serie (con números hasta el nueve, incluyendo el cero).

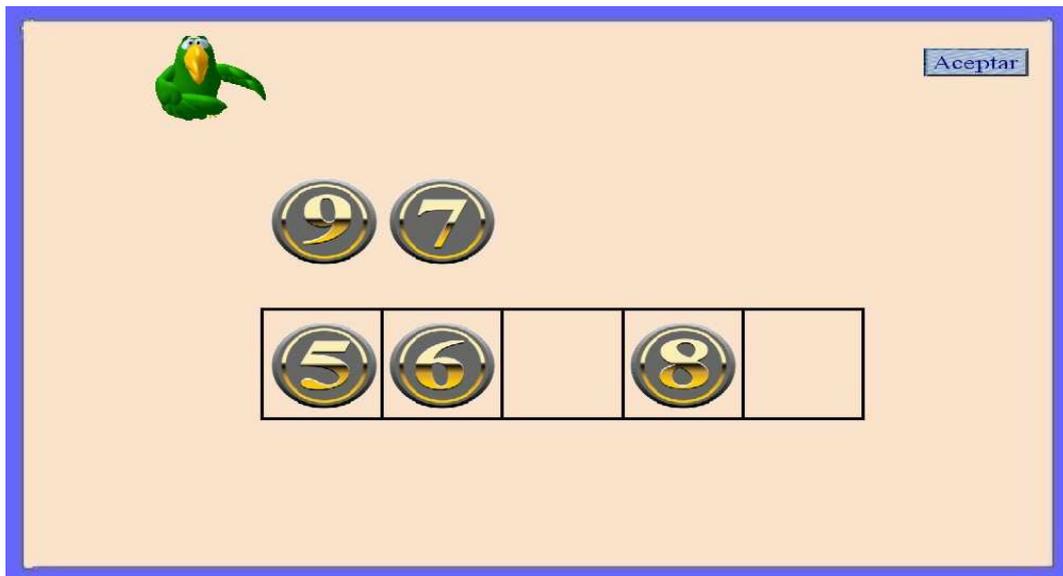
Ejemplo: Igual que en el ejercicio de nivel medio hay que ir pinchando en orden los números para completar el dibujo, pero ahora hay que pinchar en primer lugar sobre el número 0 (que es el que indica el comienzo de la parte del dibujo que falta) y llegar hasta el 9.



2.1.3.3 Actividad Tipo 3

Objetivo: Completar una serie en la que faltan números (con números hasta el nueve, incluido el cero).

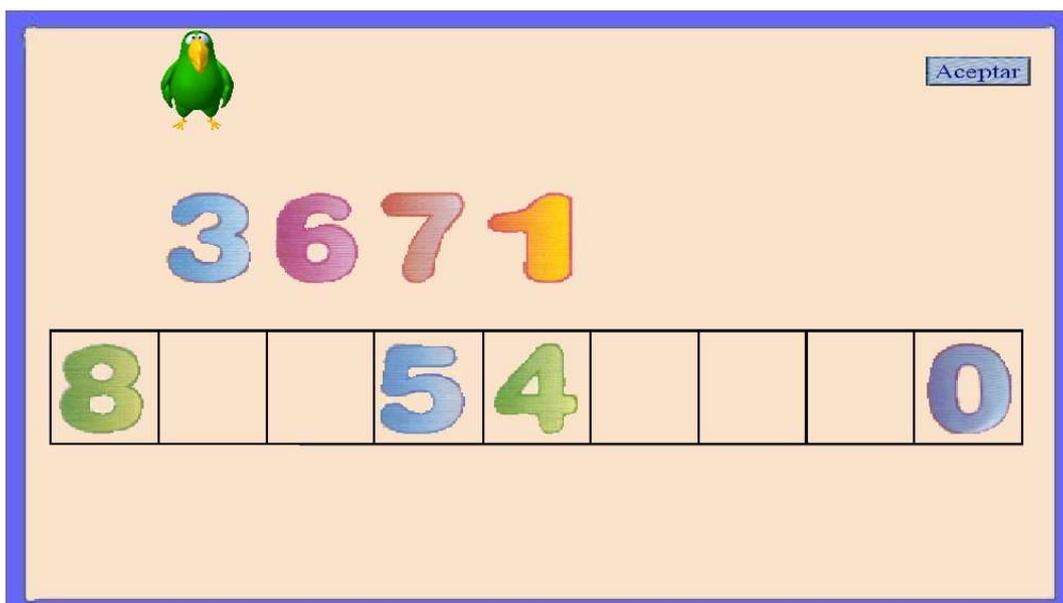
Ejemplo: Las actividades son iguales que las de nivel bajo y medio, pero hasta el número nueve.



2.1.3.4 Actividad Tipo 5

Objetivo: Contar hacia atrás con la serie numérica.

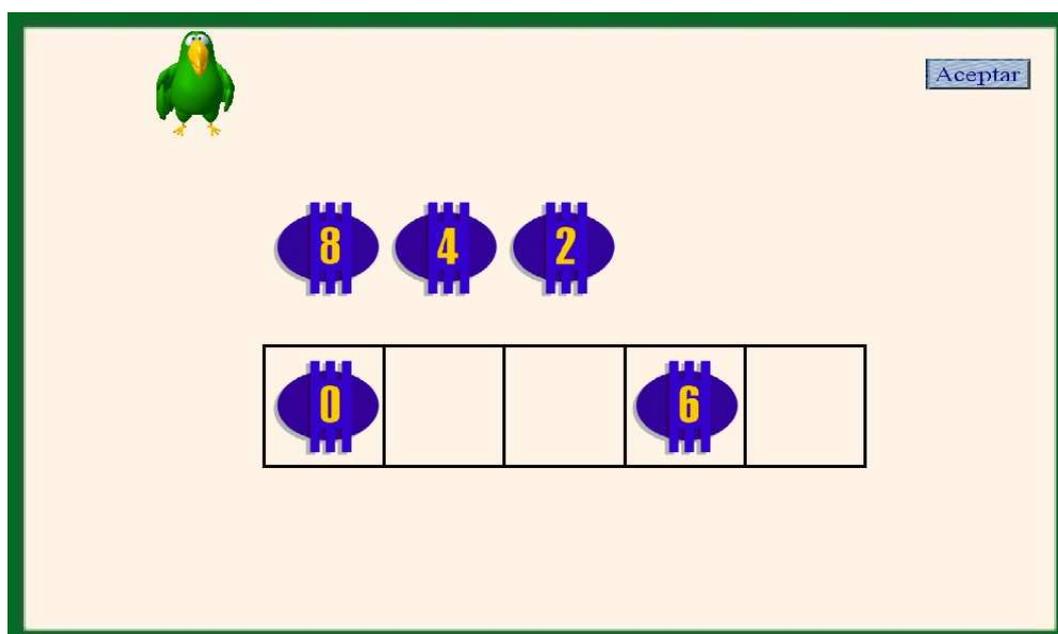
Ejemplo: En la actividad que se ve en la figura hay que ordenar los números de mayor a menor, colocándolos en el lugar que les corresponda dentro del recuadro negro.



2.1.3.5 Actividad Tipo 6

Objetivo: Contar de dos en dos.

Ejemplo: En la figura hay que ordenar los números contando de dos en dos de menor a mayor (es decir, el 0, el 2, el 4, etc), colocando los que se encuentran en la parte superior en el lugar que les corresponda dentro del recuadro negro.



2.2 Reconocimiento del número

2.2.1 NIVEL BAJO

2.2.1.1 Actividad Tipo 2

Objetivo: Al oír el número, saber identificarlo con el símbolo correspondiente (con números hasta el tres).

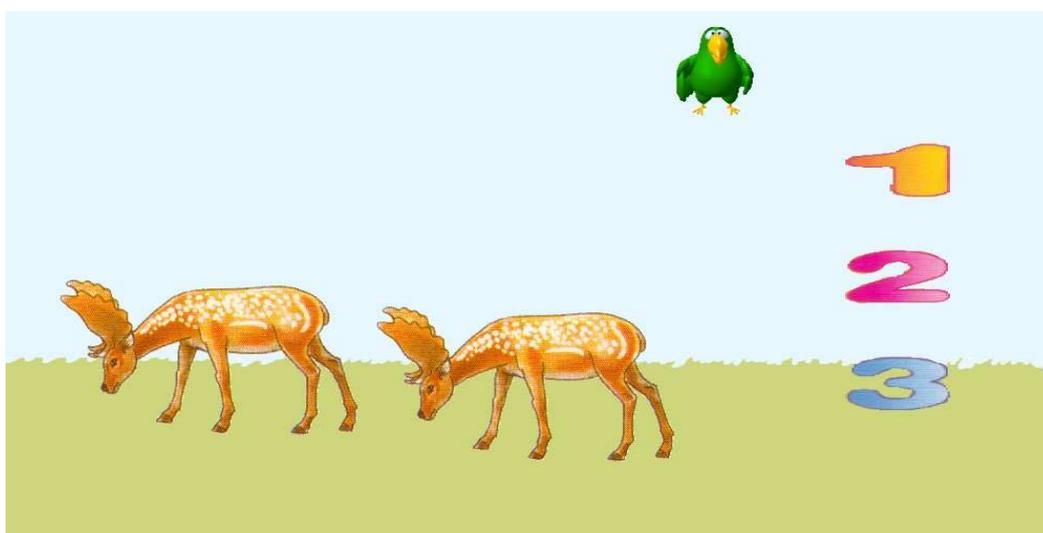
Ejemplo: En este ejercicio el agente pedagógico dirá un número y el alumno tiene que pinchar en el símbolo que indique ese número, eligiendo entre los que se encuentren en pantalla. En este caso, si por ejemplo, el agente pedagógico dice el número dos, habría que pinchar en la bolita que contiene el símbolo del número dos.



2.2.1.2 Actividad Tipo 3

Objetivo: Al ver un conjunto, saber relacionarlo con el símbolo correspondiente (con números hasta el tres).

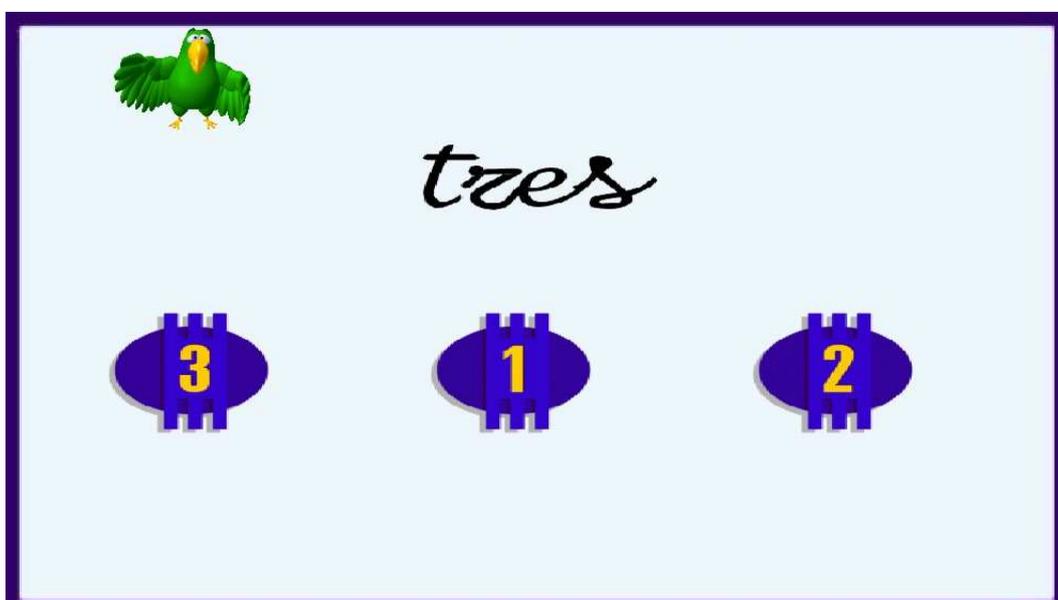
Ejemplo: En estas actividades se muestra una figura que contiene una colección de objetos y el alumno tendrá que pinchar en los números que hay en el lado derecho de la pantalla para indicar cuántos objetos componen la colección mostrada. Para una resolución correcta del ejercicio habría que pinchar en el número 2.



2.2.1.3 Actividad Tipo 4

Objetivo: Al ver el número escrito, saber relacionarlo con su símbolo (con números hasta el tres).

Ejemplo: En la parte superior de la pantalla se escribe el número, que en este caso es el "tres" y hay que pinchar en el símbolo que representa dicho número, de entre los que se encuentran en la parte inferior de la pantalla, es decir en el "3".

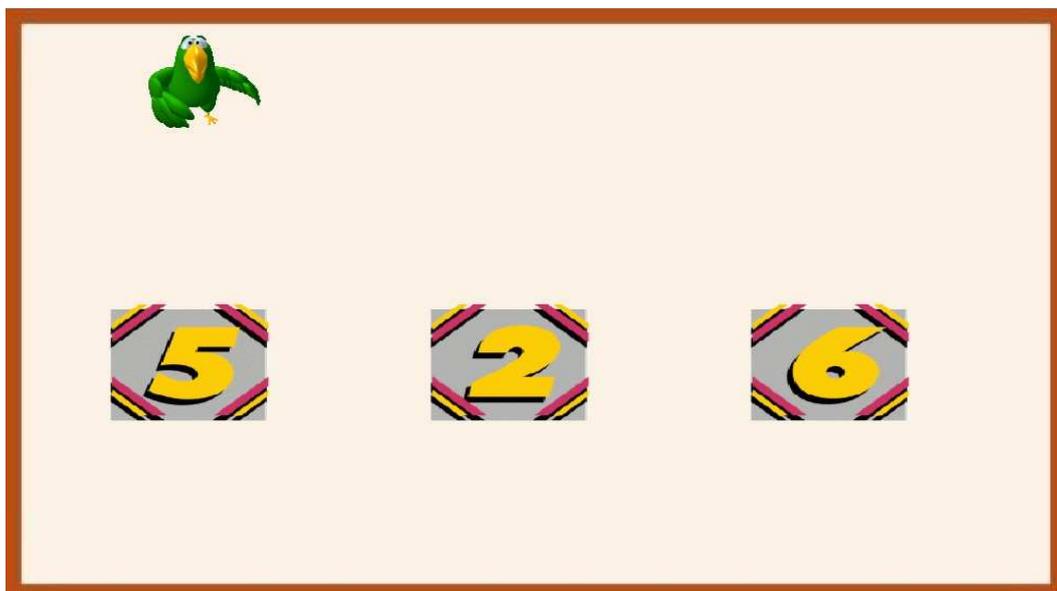


2.2.2 NIVEL MEDIO

2.2.2.1 Actividad Tipo 2

Objetivo: Al oír el número, saber identificarlo con el símbolo correspondiente (con números hasta el seis, incluido el cero).

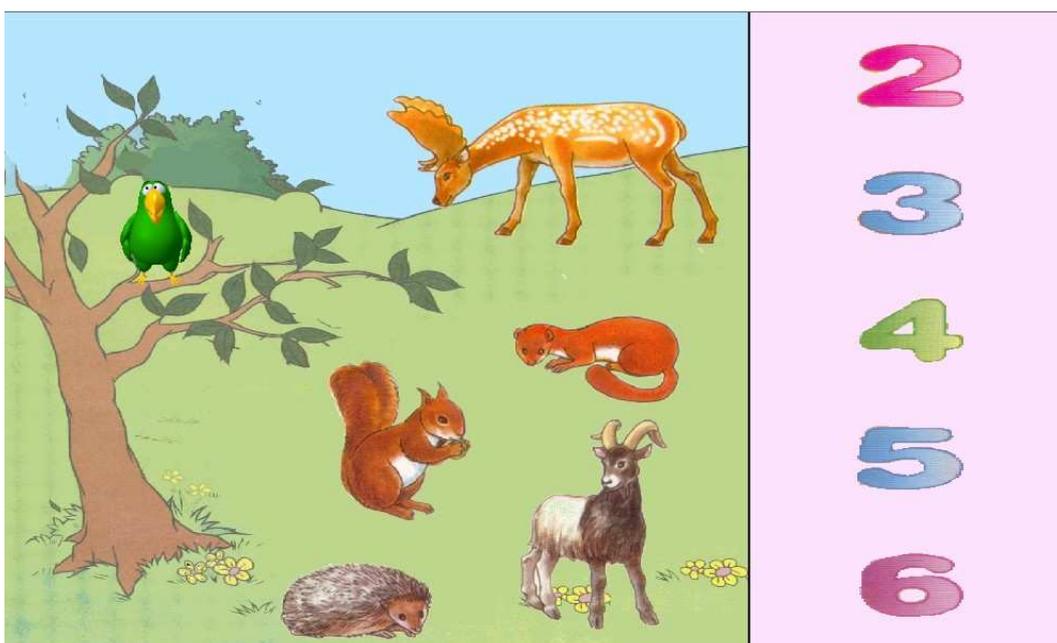
Ejemplo: Igual que en la actividad de nivel bajo pero incluyendo el cero y pudiendo utilizarse los números hasta el seis, incluyendo el cero.



2.2.2.2 Actividad Tipo 3

Objetivo: Al ver un conjunto, saber relacionarlo con el símbolo correspondiente (con números hasta el seis).

Ejemplo: La resolución es igual que en las actividades de este mismo tipo en nivel bajo, pero con números hasta el seis (sin incluir el cero).



2.2.2.3 Actividad Tipo 4

Objetivo: Al ver el número escrito, saber relacionarlo con su símbolo (con números hasta el seis, incluyendo el cero).

Ejemplo: Igual que en el ejercicio de nivel bajo, pero añadiendo los números hasta el seis y el cero.

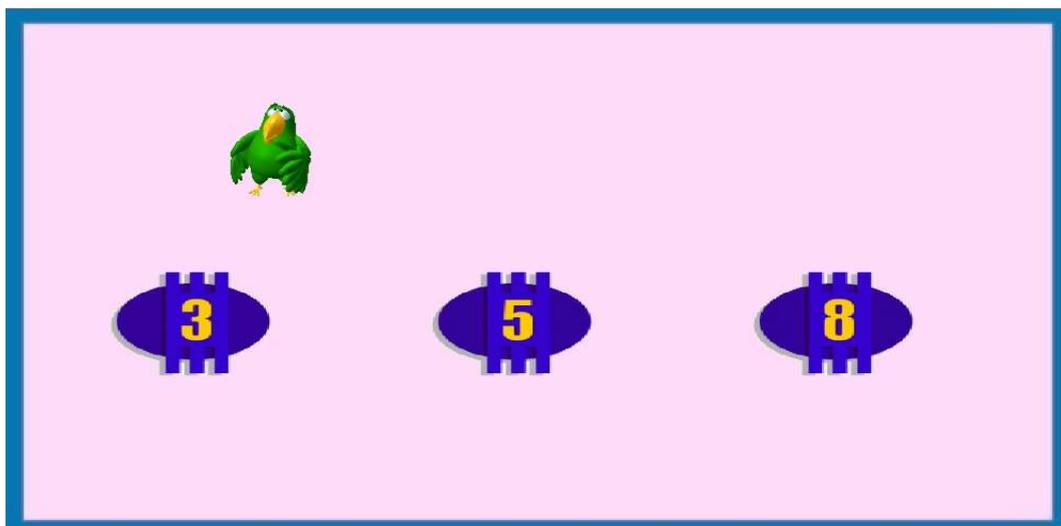


2.2.3 NIVEL ALTO

2.2.3.1 Actividad Tipo 2

Objetivo: Al oír el número saber identificarlo con el símbolo correspondiente (con números hasta el nueve, incluido el cero).

Ejemplo: Igual que en las actividades de niveles bajo y medio, pero con números hasta el nueve, incluyendo el cero.



2.2.3.2 Actividad Tipo 3

Objetivo: Al ver un conjunto, saber relacionarlo con el símbolo correspondiente (con números hasta el nueve).

Ejemplo: Igual que los ejercicios de nivel bajo y medio, pero con números hasta el nueve (sin incluir el cero). El alumno tiene que pinchar en el ocho para la correcta resolución del ejercicio.



2.2.3.3 Actividad Tipo 4

Objetivo: Al ver el número escrito, saber relacionarlo con su símbolo (con números hasta el nueve, incluyendo el cero).

Ejemplo: Igual que en los ejercicio de nivel bajo y medio, pero añadiendo los números hasta el nueve.



2.3 Cardinalidad

2.3.1 NIVEL BAJO

2.3.1.1 Actividad Tipo 2

Objetivo: Al oír el cardinal de una determinada colección de objetos, saber relacionar de entre un conjunto de colecciones de diferentes tamaños, el (los) correspondiente (s) con dicho cardinal (hasta el número tres).

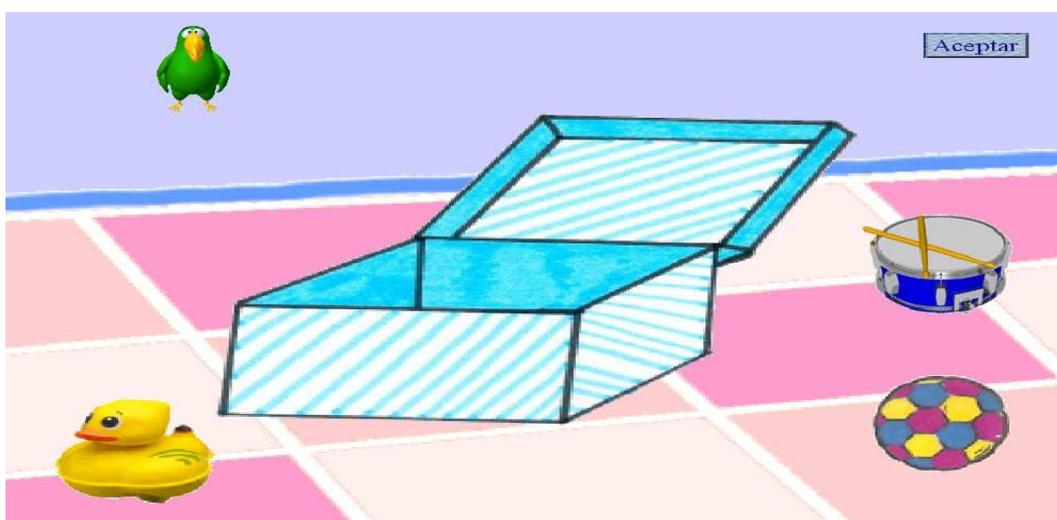
Ejemplo: En este ejercicio, después de oír el número que diga el agente habría que pinchar en el (los) dibujo (s) de la zona inferior, que tenga tantos objetos como el número que haya indicado el agente. Si el agente dice el número uno, en los pantalones, si dice el dos, en los tenedores y el tres, en los niños.



2.3.1.2 Actividad Tipo 3

Objetivo: Al oír el cardinal de una determinada colección de objetos, saber construir o completar dicha colección (hasta el número tres).

Ejemplo: El agente en este ejemplo le dirá al niño que coloque uno, dos o tres juguetes dentro de la caja. Para resolver el ejercicio habrá que mover tantos juguetes como haya indicado el agente al interior de la caja y pinchar en el botón de Aceptar.



2.3.1.3 Actividad Tipo 4

Objetivo: Al mostrar una colección de un número determinado de objetos, saber seleccionar, de entre varios números, el símbolo correspondiente (hasta el número tres).

Ejemplo: En esta actividad se muestra un figura que contiene varios objetos teniendo algunos de ellos características comunes. El alumno tendrá que pinchar en los números que hay en el lado derecho de la pantalla para indicar cuál es el cardinal que plantea el agente. Por ejemplo, en este caso el agente pregunta: ¿Cuántos pájaros hay?, y para una correcta resolución del ejercicio habría que pinchar en el 3.



2.3.1.4 Actividad Tipo 5

Objetivo: Dado el símbolo, saber seleccionar de entre una colección de conjuntos de objetos, aquél, o aquellos, que tienen, precisamente, esos elementos (hasta el número tres).

Ejemplo: Como se ve en la siguiente figura, en el centro de la pantalla se muestra el símbolo, y alrededor varias colecciones de objetos. En este caso puesto

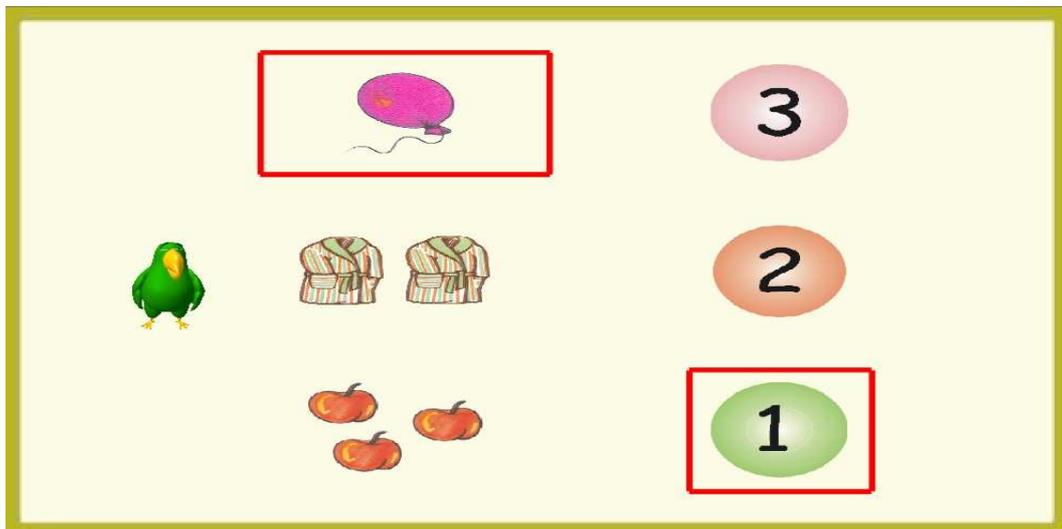
que se están buscando colecciones que consten de tres elementos, habría que pinchar en los niños y en las manos de colores.



2.3.1.5 Actividad Tipo 6

Objetivo: Relacionar cada una de las colecciones de un conjunto de colecciones de diferentes tamaños, con los dígitos que indican su cardinal (igual número de dígitos que de colecciones y hasta el número tres).

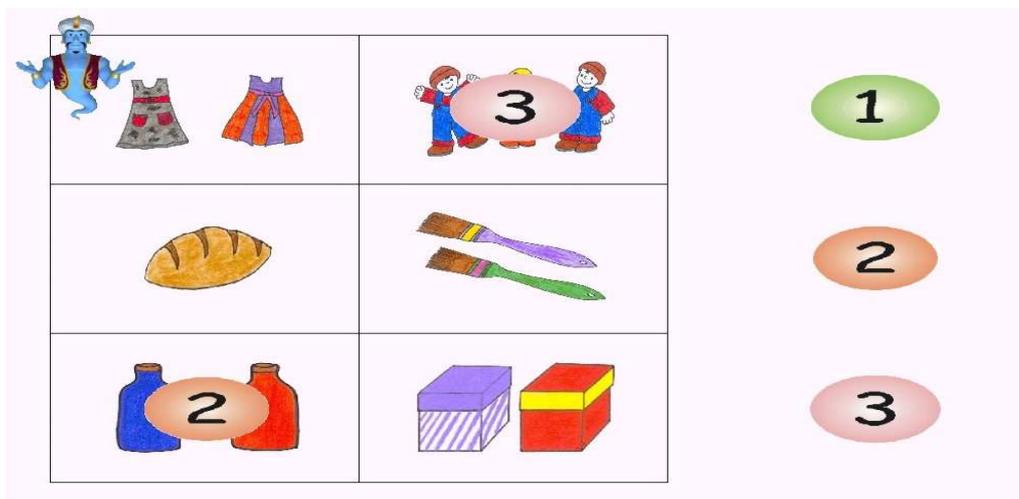
Ejemplo: Como se puede observar en la figura inferior, hay dos columnas. La primera contiene colecciones que tienen un número variable de elementos (hasta tres) y la segunda contiene los símbolos que se corresponden con esas colecciones. El alumno tiene que emparejar las colecciones de la columna de la izquierda con los dígitos de la de la derecha. Si el alumno realiza correctamente el emparejamiento, se recuadran las parejas con un color.



2.3.1.6 Actividad Tipo 7

Objetivo: Relacionar cada una de las colecciones de un conjunto de colecciones de diferentes tamaños con los dígitos que indican su cardinal (menor número de dígitos que de colecciones y hasta el número tres).

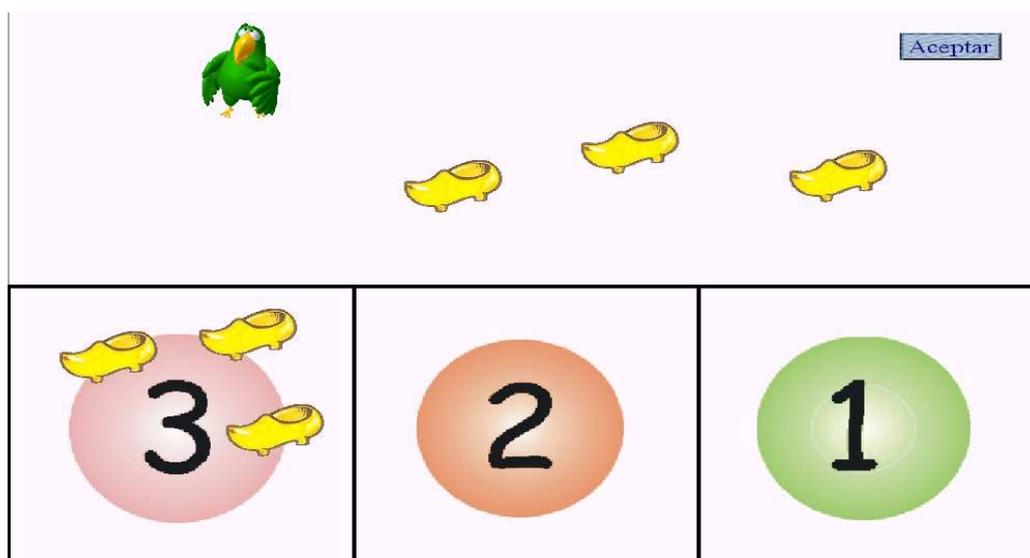
Ejemplo: En este tipo de actividades hay que pinchar primero en una figura de la izquierda y luego en el número de la derecha, que indica cuántos elementos hay en la figura que se ha pinchado. Como se puede ver en la imagen, cuando se acierta se coloca el número encima de la figura para indicar que esa figura ya está resuelta. Cuando se resuelven todas, se pasa a la siguiente actividad.



2.3.1.7 Actividad Tipo 8

Objetivo: Al ver el cardinal (símbolo) de una determinada colección de objetos, saber construir o completar dicha colección (hasta el número tres).

Ejemplo: En la actividad que nos sirve de ejemplo, hay que llevar a cada recuadro tantos zapatos como indique el número. Es decir, habría que colocar tres zapatos sobre la figura del número 3, dos, sobre la del número 2, y uno, sobre la del 1. Como la dificultad del ejercicio es baja, el número de zapatos es el exacto para la resolución del ejercicio (no sobran zapatos). Hay que tener en cuenta que los números no tienen que ser correlativos.

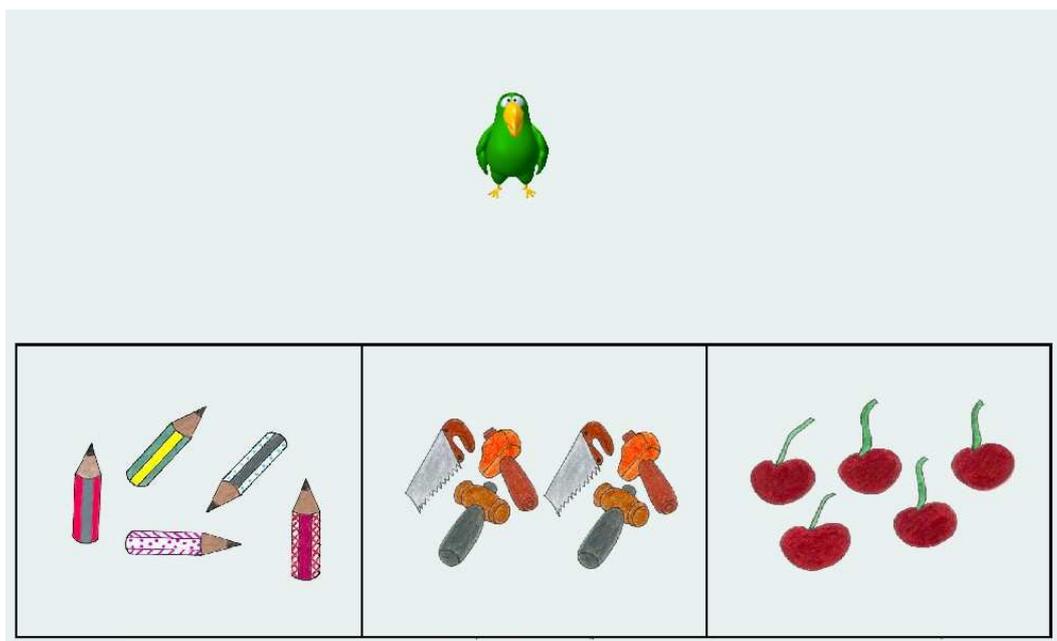


2.3.2 NIVEL MEDIO

2.3.2.1 Actividad Tipo 2

Objetivo: Al oír el cardinal de una determinada colección de objetos, saber relacionar de entre un conjunto de colecciones de diferentes tamaños, el (los) correspondiente (s) con dicho cardinal (hasta el número seis).

Ejemplo: Igual en la actividad de nivel bajo. Si el agente dice el número cinco habría que pinchar en los lápices y en las cerezas, y si dice seis, en las herramientas.



2.3.2.2 Actividad Tipo 3

Objetivo: Al oír el cardinal de una determinada colección de objetos, saber construir o completar dicha colección (hasta el número seis).

Ejemplo: La actividad es igual que la de nivel bajo pero hasta el número seis. Por ejemplo, en este ejercicio el agente le pedirá al alumno que coloque cuatro yogures en la nevera. Una vez que lo haya hecho, hay que pinchar en Aceptar.



2.3.2.3 Actividad Tipo 4

Objetivo: Al mostrar una colección de un número determinado de objetos, saber seleccionar, de entre varios números, el símbolo correspondiente (hasta el número seis).

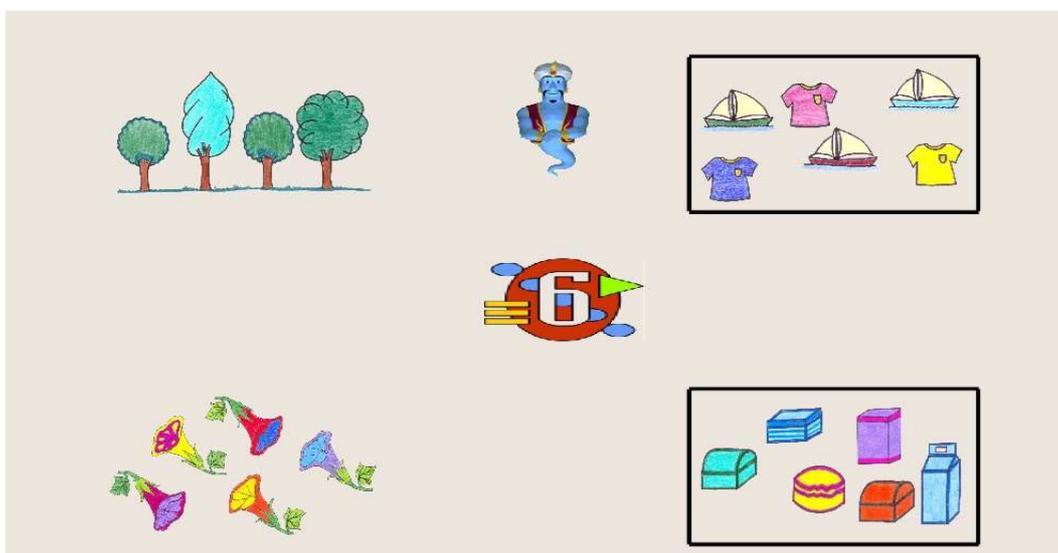
Ejemplo: Al igual que en los ejercicios de nivel bajo, el alumno tendrá que pinchar en los números que hay en el lado derecho de la pantalla para indicar cual es el que soluciona la pregunta planteada por el agente. Por ejemplo, en este caso el agente pregunta: ¿Cuántos juguetes hay?, y para la correcta resolución del ejercicio, habría que pinchar en el número 6.



2.3.2.4 Actividad Tipo 5

Objetivo: Dado el símbolo, saber seleccionar de entre una colección de conjuntos de objetos aquél o aquellos que tienen precisamente esos elementos (hasta el número seis).

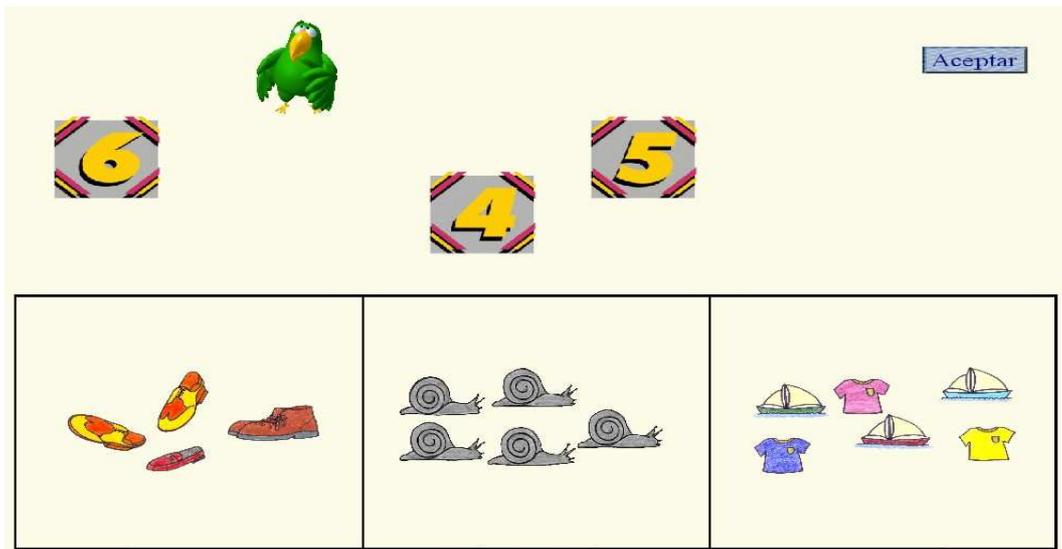
Ejemplo: Como se puede ver en la siguiente figura, la resolución es igual que en la actividad de tipo bajo. En el centro de la pantalla se muestra en este caso el símbolo 6, y al pinchar en las colecciones que tienen seis elementos, estas se marcan con un recuadro (cuando se hayan señalado todas las que tienen el número de elementos que se pide, se considera correcto el ejercicio).



2.3.2.5 Actividad Tipo 6

Objetivo: Relacionar cada una de las colecciones de un conjunto de colecciones de diferentes tamaños con los dígitos que indican su cardinal (igual número de dígitos que de colecciones y hasta el número seis).

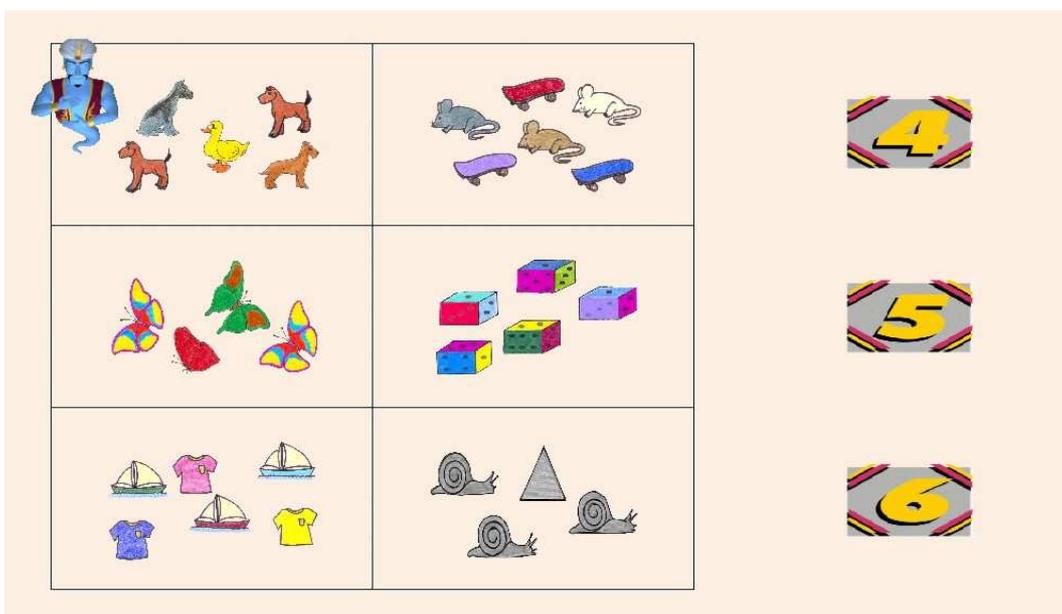
Ejemplo: Se puede observar en la figura que en la parte superior de la pantalla hay tres símbolos numéricos, y en la parte inferior, hay igual número de colecciones de objetos. Hay que colocar cada símbolo encima de la colección que tiene tantos objetos como indica el símbolo.



2.3.2.6 Actividad Tipo 7

Objetivo: Relacionar cada una de las colecciones de un conjunto de colecciones de diferentes tamaños, con los dígitos que indican su cardinal (menor número de dígitos que de colecciones y hasta el número seis).

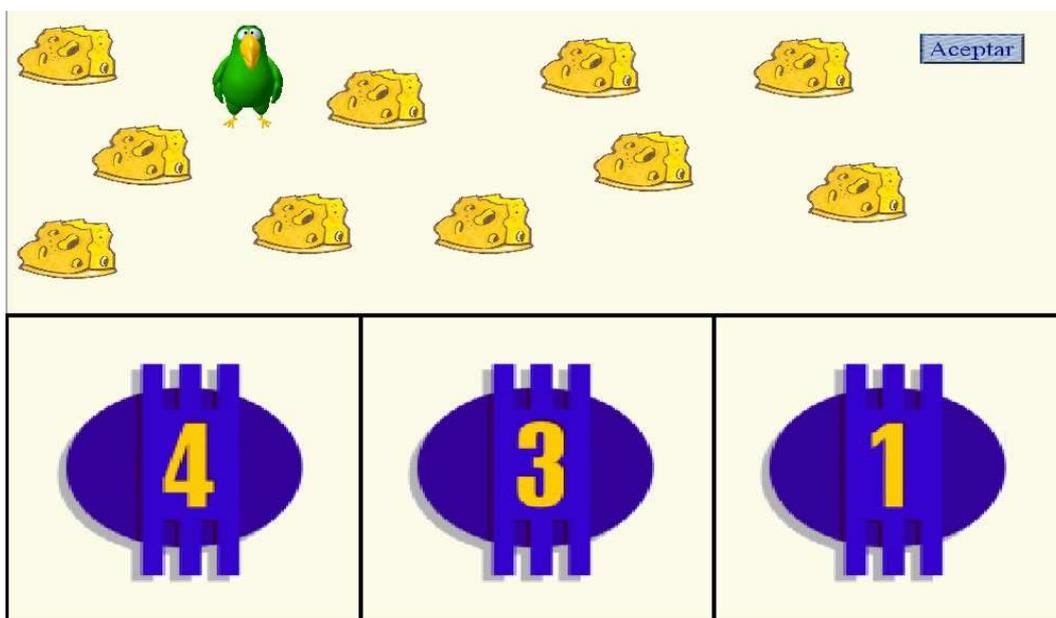
Ejemplo: Igual que la actividad de tipo bajo, pero en este caso, el número de objetos en las colecciones puede llegar hasta seis.



2.3.2.7 Actividad Tipo 8

Objetivo: Al ver el cardinal (símbolo) de una determinada colección de objetos, saber construir o completar dicha colección (hasta el número seis).

Ejemplo: La actividad es igual que en la de nivel bajo correspondiente, pero con símbolos y colecciones de hasta seis elementos. Pueden sobrar elementos a la hora de construir las colecciones.

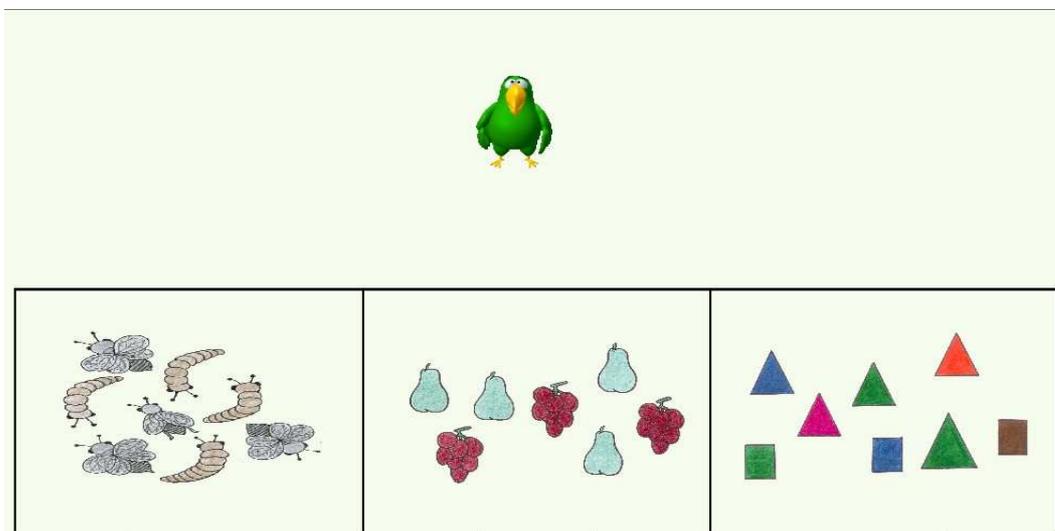


2.3.3 NIVEL ALTO

2.3.3.1 Actividad Tipo 2

Objetivo: Al oír el cardinal de una determinada colección de objetos, saber relacionar de entre un conjunto de colecciones de diferentes tamaños, el (los) correspondiente (s) con dicho cardinal (hasta el número nueve).

Ejemplo: Igual en la actividades de nivel bajo y medio, pero hasta nueve elementos. Si el agente dice siete, habría que pinchar en la fruta, si dice ocho, en los gusanos y mariposas, y si dice nueve, en las figuras geométricas.



2.3.3.2 Actividad Tipo 3

Objetivo: Al oír el cardinal de una determinada colección de objetos, saber construir o completar dicha colección (hasta el número nueve).

Ejemplo: La actividad es igual que la de nivel bajo y medio, pero hasta el número nueve. Así, en este ejercicio, el agente le pedirá al alumno que coloque siete peces en la pecera.



2.3.3.3 Actividad Tipo 4

Objetivo: Al mostrar una colección de un número determinado de objetos, saber seleccionar, de entre varios números, el símbolo correspondiente (hasta el número nueve).

Ejemplo: Es igual que en los ejercicios de nivel bajo y medio, pero hasta el número nueve. El agente, en este caso, pregunta: ¿Cuántos tortugas hay?, y como se puede observar, se marca como correcta la respuesta al pinchar en el símbolo del número 7.



2.3.3.4 Actividad Tipo 5

Objetivo: Dado el símbolo, saber seleccionar de entre una colección de conjuntos de objetos aquél o aquellos que tienen precisamente esos elementos (hasta el número nueve).

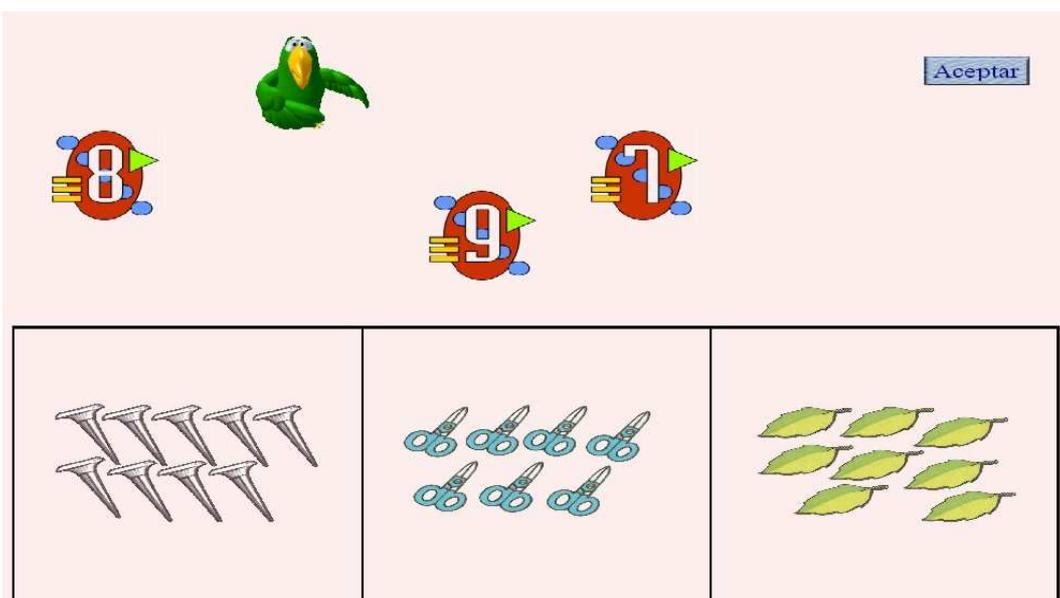
Ejemplo: Igual que en las actividades de tipo medio y alto, pero con números hasta el nueve.



2.3.3.5 Actividad Tipo 6

Objetivo: Relacionar cada una de las colecciones de un conjunto de colecciones de diferentes tamaños con los dígitos que indican su cardinal (igual número de dígitos que de colecciones y hasta el número nueve).

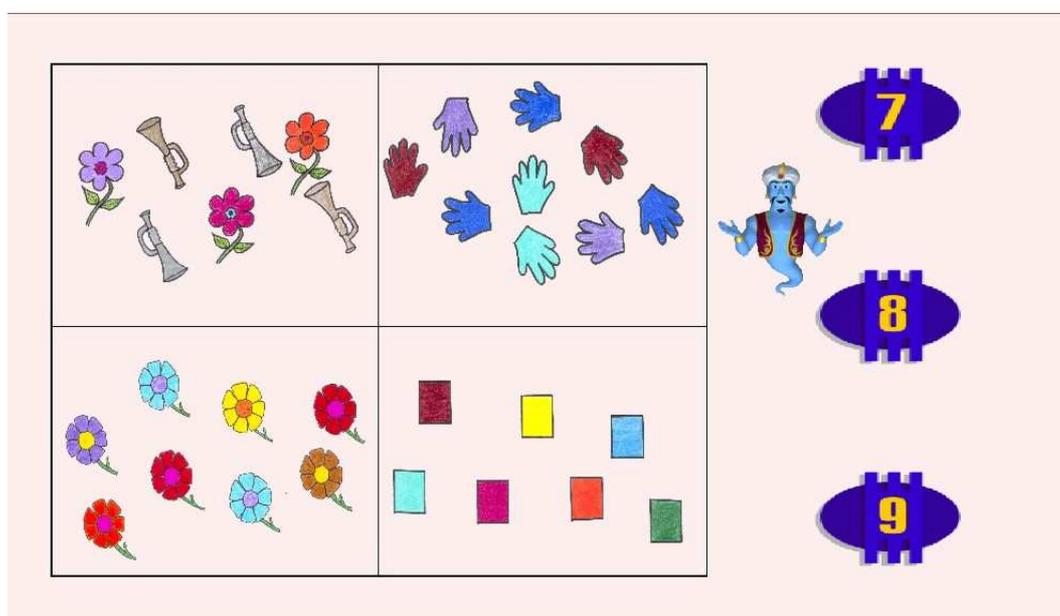
Ejemplo: Igual que la actividad de nivel medio, pero con números hasta el nueve.



2.3.3.6 Actividad Tipo 7

Objetivo: Relacionar cada una de las colecciones de un conjunto de colecciones de diferentes tamaños con los dígitos que indican su cardinal (menor número de dígitos que de colecciones y hasta el número nueve).

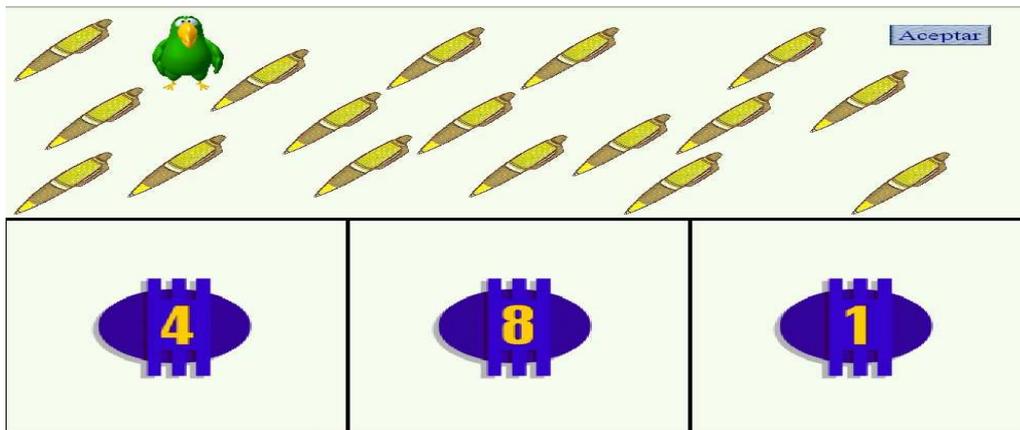
Ejemplo: La resolución es igual que en las actividades de tipo bajo y medio, pero el número de objetos en las colecciones puede llegar hasta nueve.



2.3.3.7 Actividad Tipo 8

Objetivo: Al ver el cardinal (símbolo) de una determinada colección de objetos, saber construir o completar dicha colección (hasta el número seis).

Ejemplo: La actividad es igual que las de nivel bajo y medio de este tipo, pero los símbolos y las colecciones pueden tener hasta nueve elementos. Pueden sobrar elementos a la hora de construir las colecciones.



2.3.3.8 Actividad Tipo 10

Objetivo: Relacionar cada una de las colecciones de un conjunto de colecciones de diferentes tamaños con los dígitos que indican su cardinal (menor número de dígitos que de colecciones) y la negación de los mismos (hasta el número nueve).

Ejemplo: Como se observa en la figura, en la izquierda de la pantalla hay conjuntos de colecciones con un número variable de elementos,, y a la derecha, un símbolo y su negación. En este ejercicio hay que relacionar la colección de campanas con el número 5 y los peces y los niños con la negación del número 5. Cuando se acierta se coloca el número encima de la figura para indicar que esa figura ya está señalada correctamente.



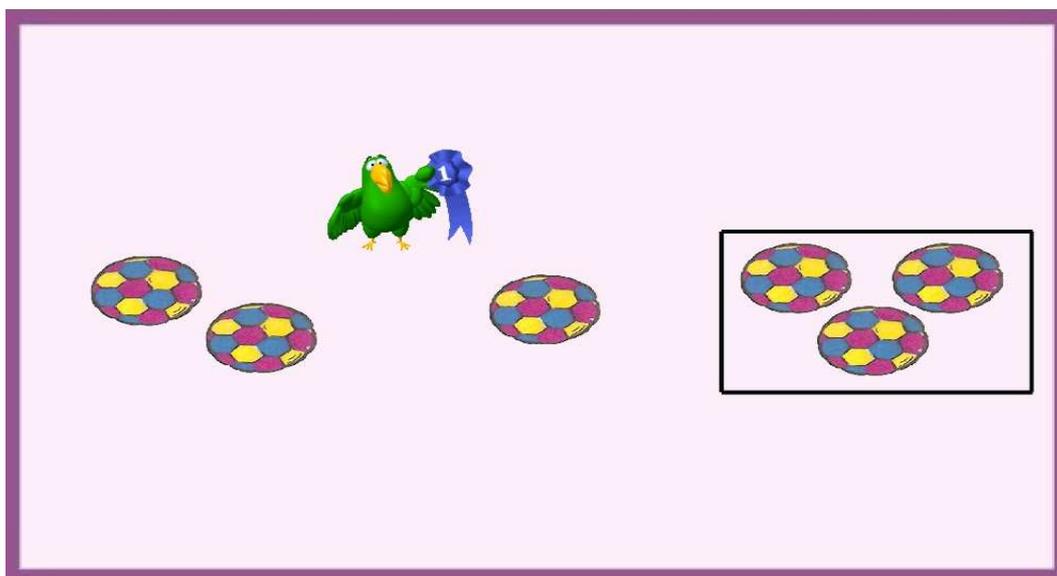
2.4 Orden

2.4.1 NIVEL BAJO

2.4.1.1 Actividad Tipo 1

Objetivo: Comparar colecciones, indicando dónde hay más o menos objetos (hasta el número tres).

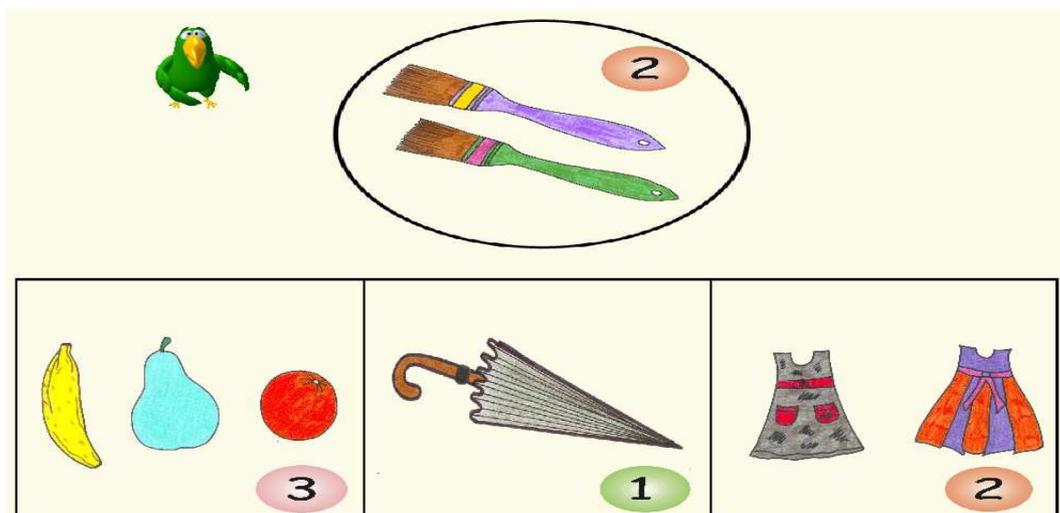
Ejemplo: Como se puede ver en la siguiente imagen, se presentan varias colecciones con diferente número de objetos y el alumno tiene que pinchar aquel que sea solución a la pregunta que formula el agente. En este caso, señala la colección en la que hay más pelotas.



2.4.1.2 Actividad Tipo 3

Objetivo: Etiquetar conjuntos con varios objetos y clasificarlos según el número de objetos (mayor que, menor que, igual que) (hasta el número tres).

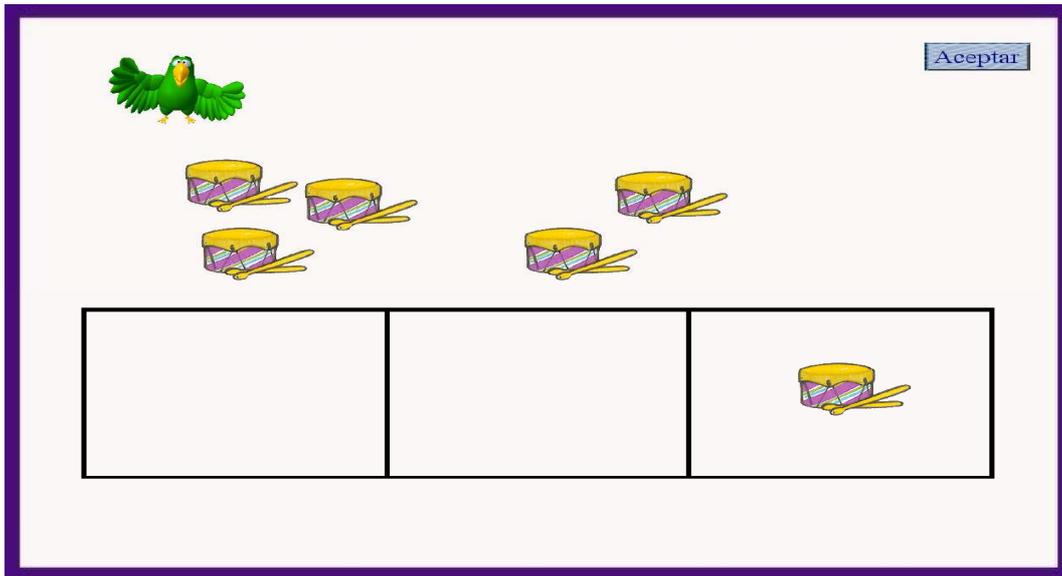
Ejemplo: En la parte superior de la pantalla, en un círculo, se muestra una colección de objetos con su cardinal y, en la parte inferior, otras tres colecciones, que contienen igual, menor y mayor número de objetos que la que está dentro del círculo. Hay que pinchar la colección de la parte inferior de la pantalla que sea la solución a la pregunta planteada por el agente. Por ejemplo, si se pide menor número de objetos que la colección dada, habría que pinchar en el paraguas, si dice igual número de objetos, en los trajes y, por último si dice mayor número de objetos, en las frutas.



2.4.1.3 Actividad Tipo 4

Objetivo: Cambiar la disposición física de colecciones de diferente número de objetos y ordenarlas (hasta el número tres).

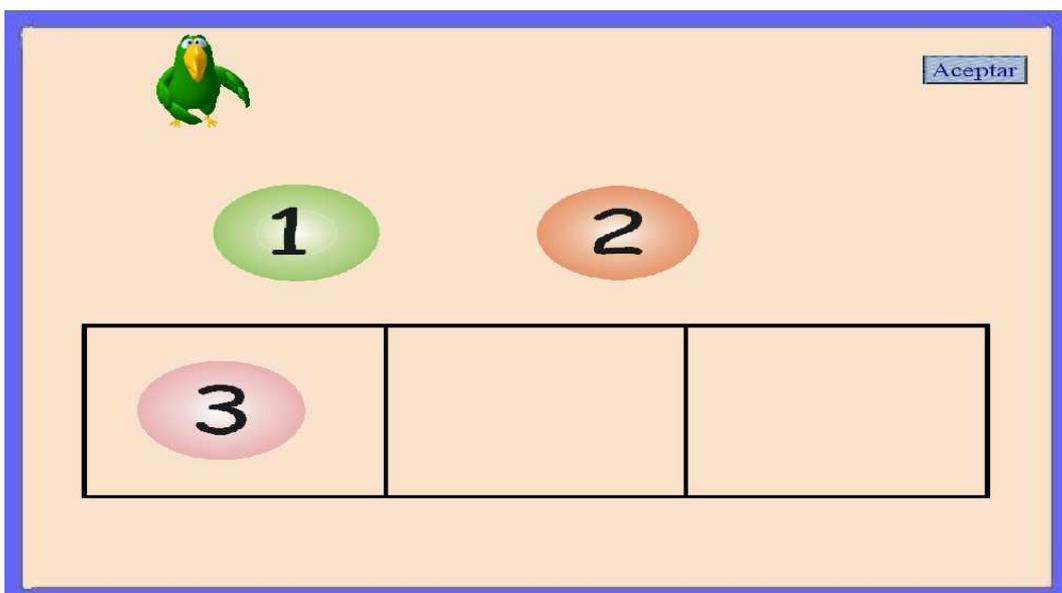
Ejemplo: Se dan tres colecciones de objetos que contienen uno, dos y tres objetos cada una (en este caso tambores) y hay que ordenarlas según se indique, de mayor a menor, o de menor a mayor (en el ejemplo de mayor a menor).



2.4.1.4 Actividad Tipo 5

Objetivo: Ordenar una pequeña secuencia de números (hasta el número tres)

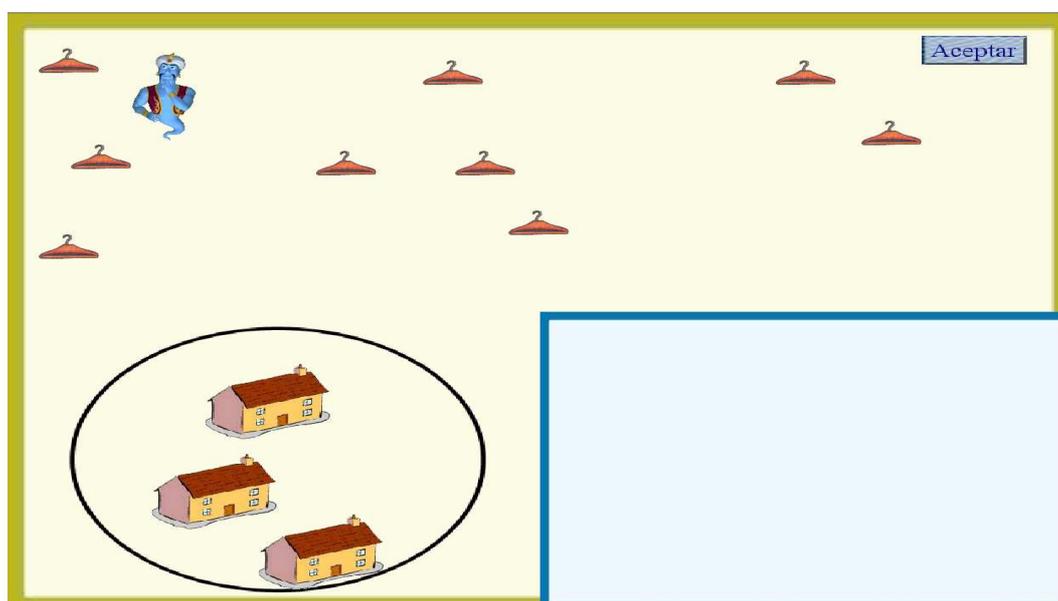
Ejemplo: En estas actividades hay que ordenar los números de menor a mayor (o de mayor a menor), colocándolos en el lugar que les corresponda dentro del recuadro negro.



2.4.1.5 Actividad Tipo 6

Objetivo: Construir un conjunto que contenga más, igual o menos elementos que uno dado (hasta el número tres).

Ejemplo: Como se puede ver en la figura, en la parte superior de la pantalla hay varios elementos en posiciones variables (en este caso percheros) y la parte inferior de la pantalla está dividida en dos zonas, un círculo y un rectángulo. El círculo contiene el conjunto de referencia y dentro del rectángulo es donde hay que construir la colección que contenga más, igual o menos elementos que el de referencia, usando los elementos que están en la parte superior de la pantalla. En este ejemplo, hay 3 casas en la zona de referencia por lo que si se pide contruir un conjunto que tenga más elementos que él, habría que poner cuatro o más percheros sobre el rectángulo azul, si pide igual, tres percheros, y si pide menor, uno o dos.

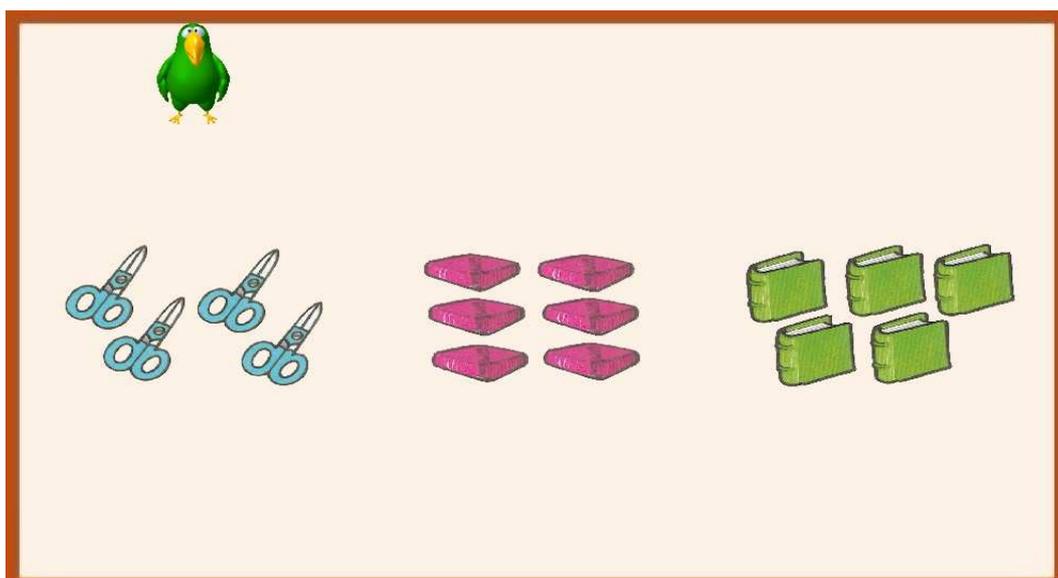


2.4.2 NIVEL MEDIO

2.4.2.1 Actividad Tipo 1

Objetivo: Comparar colecciones, indicando en la que hay más o menos objetos (hasta el número seis).

Ejemplo: El ejercicio es del mismo tipo que el de nivel bajo pero las colecciones pueden contener hasta seis objetos y ser distintas. Si en el nivel bajo se preguntaba, por ejemplo, por la colección en la que había más pelotas (puesto que todas las colecciones se componían de pelotas, en mayor o menor cantidad) aquí se habla de elementos en general, para aumentar la dificultad del ejercicio.

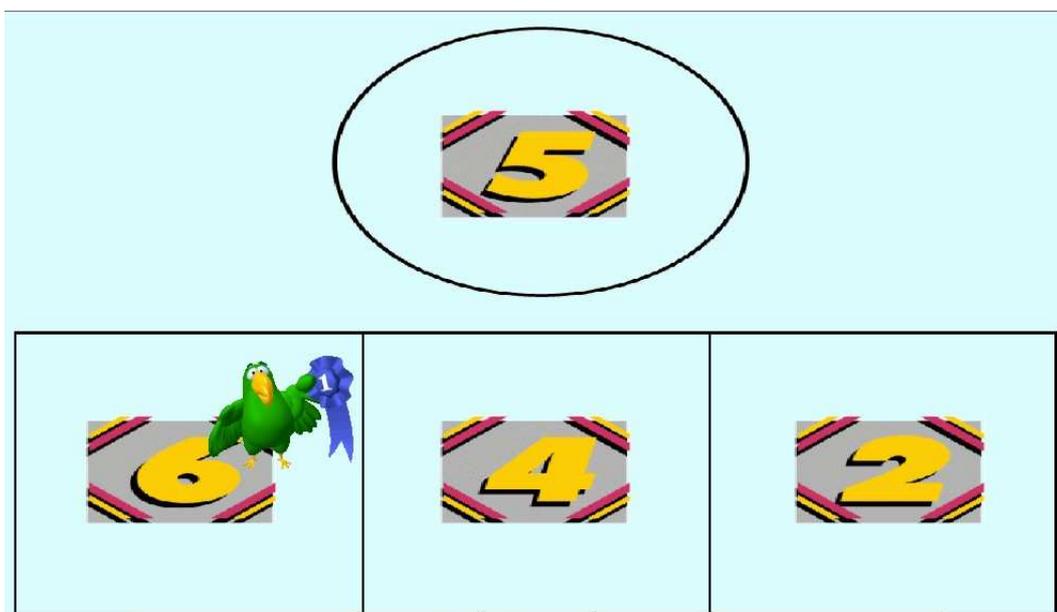


2.4.2.2 Actividad Tipo 2

Objetivo: Dado un número, señalar el anterior y/o posterior (hasta el número seis incluyendo en cero).

Ejemplo: En la parte superior de la pantalla, en un círculo se muestra el cardinal de un número y en la parte inferior, dentro de un rectángulo, hay tres números mayor, igual y menor que el que está dentro del círculo. Hay que pinchar

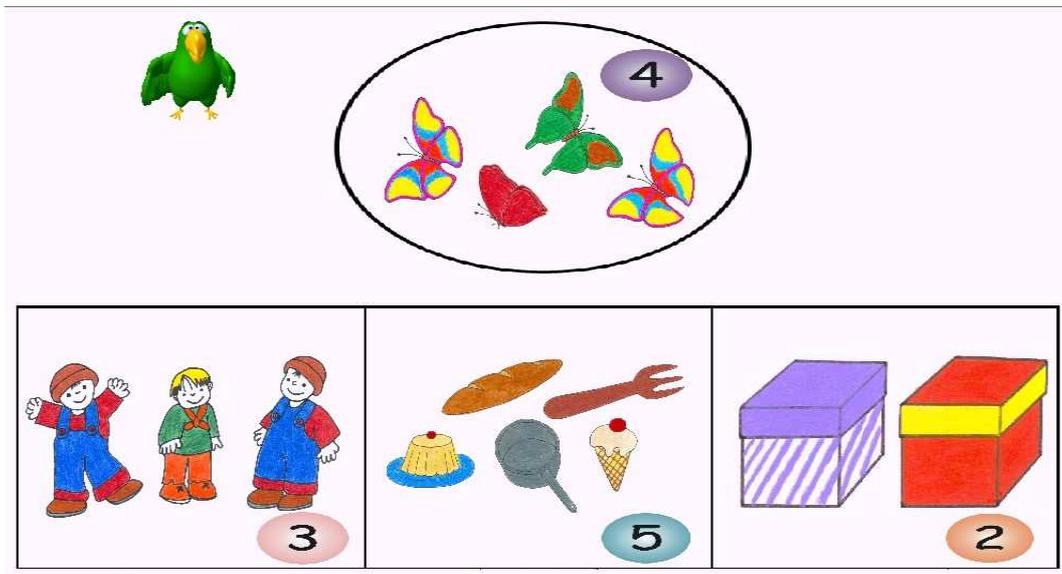
uno de estos números que sea la solución a la pregunta planteada por el agente. Por ejemplo, en este caso se pregunta por un número mayor que el indicado, por lo que la solución, como se ve en la figura, es el número seis.



2.4.2.3 Actividad Tipo 3

Objetivo: Etiquetar conjuntos con varios objetos y clasificarlos según el número de objetos (mayor que, menor que, igual que) (hasta el número seis incluyendo el cero).

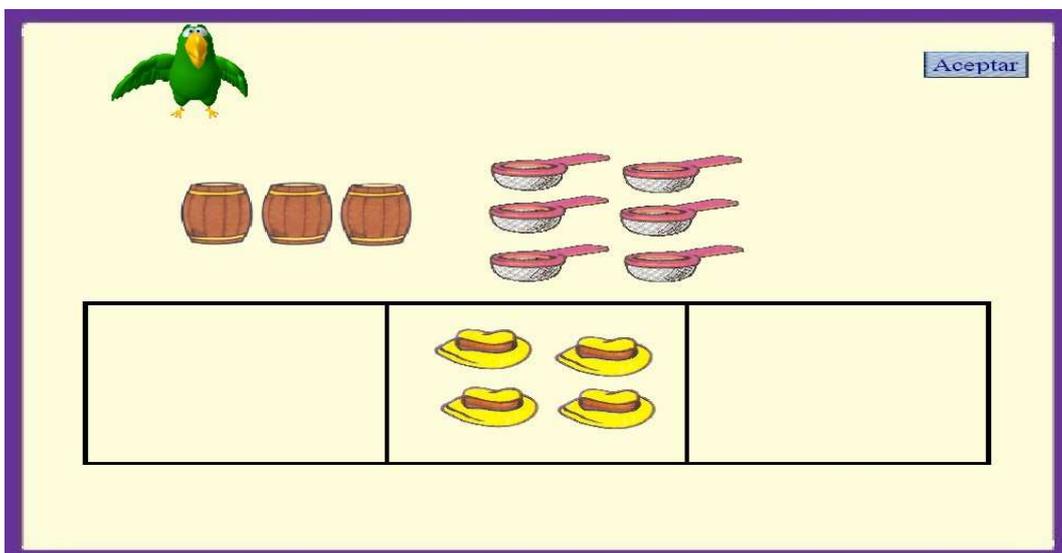
Ejemplo: La actividad es igual que la de nivel bajo, pero la cantidad de elementos en las colecciones (y, por tanto, los cardinales) pueden llegar hasta el seis (incluyendo el cero).



2.4.2.4 Actividad Tipo 4

Objetivo: Cambiar la disposición física de colecciones de diferente número de objetos y ordenarlas (hasta el número seis).

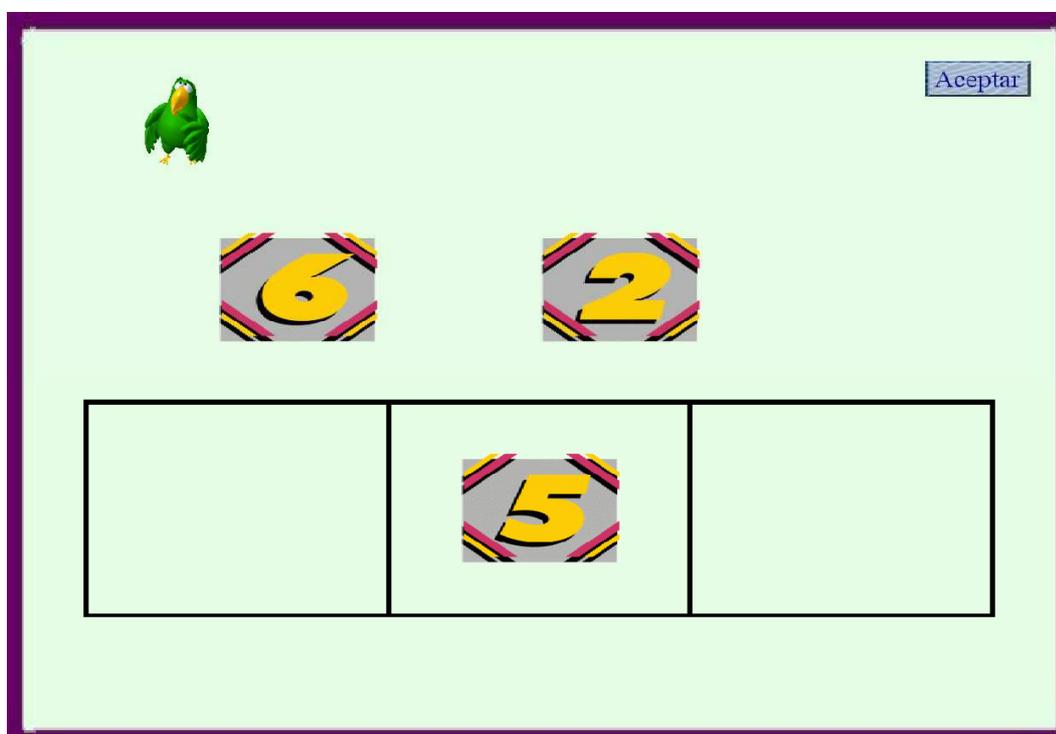
Ejemplo: Igual que en el caso de la actividad de nivel bajo, se dan tres colecciones de objetos (pueden contener hasta seis elementos cada una y ser estos distintos para aumentar la dificultad). Hay que ordenar las colecciones según se indique, de mayor a menor, o de menor a mayor.



2.4.2.5 Actividad Tipo 5

Objetivo: Ordenar una pequeña secuencia de números (hasta el número seis, incluyendo el cero).

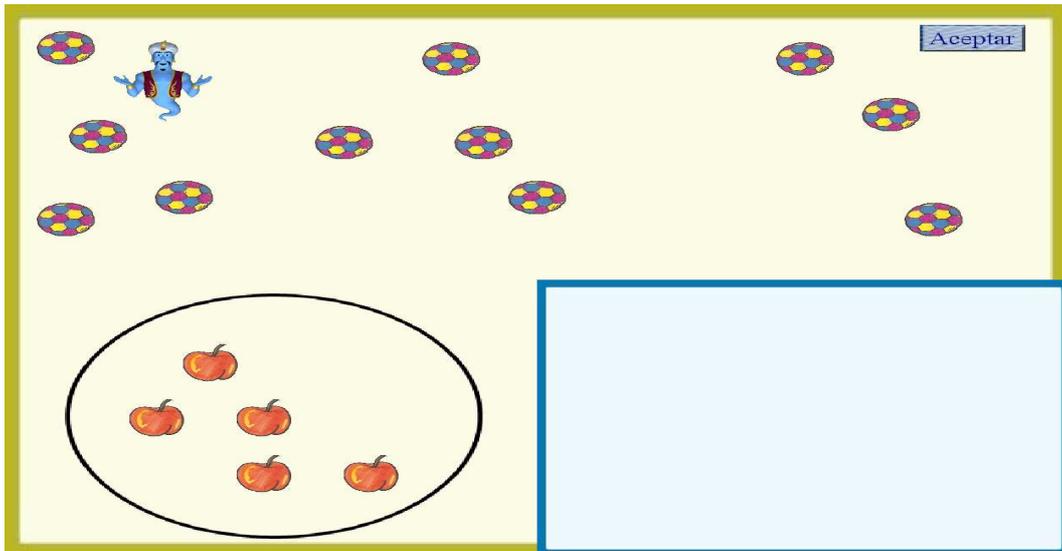
Ejemplo: En estas actividades hay que ordenar los números de menor a mayor (o de mayor a menor), colocándolos en el lugar que les corresponda dentro del recuadro negro. En este nivel no hace falta que los números sean correlativos.



2.4.2.6 Actividad Tipo 6

Objetivo: Construir un conjunto que contenga más, igual o menos elementos que uno dado (hasta el número seis).

Ejemplo: Esta actividad es igual que la de nivel bajo, pero la colección de referencia puede llegar a tener hasta seis elementos.

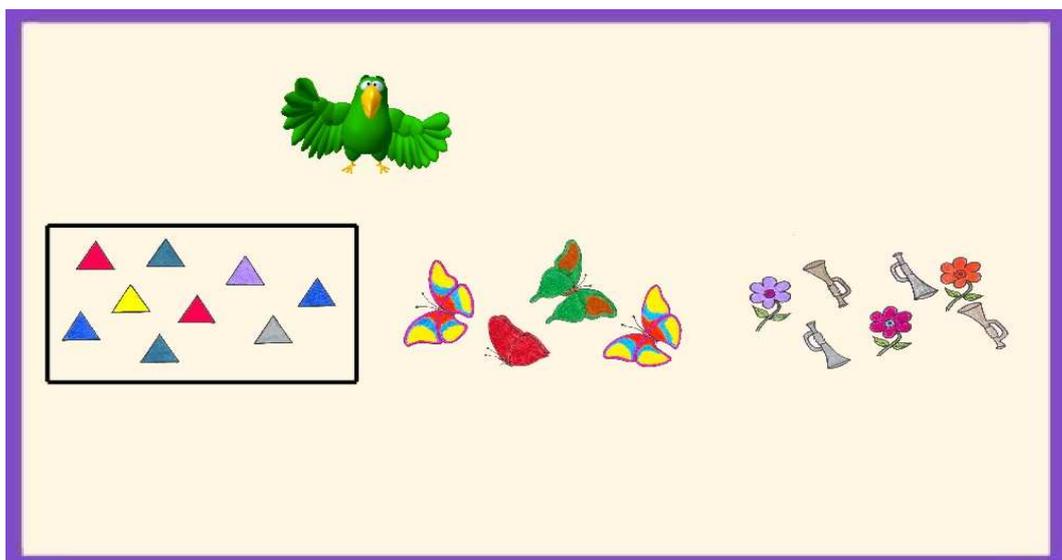


2.4.3 NIVEL ALTO

2.4.3.1 Actividad Tipo 1

Objetivo: Comparar colecciones indicando en la que hay más o menos objetos (hasta el número nueve).

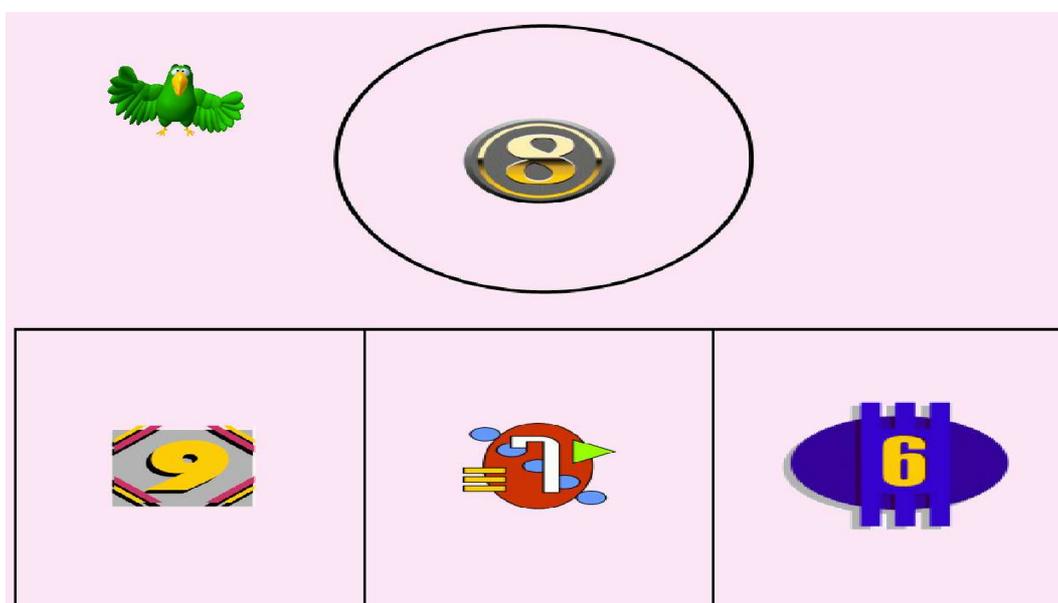
Ejemplo: El ejercicio es del mismo tipo que el de nivel medio, pero las colecciones pueden contener hasta nueve objetos.



2.4.3.2 Actividad Tipo 2

Objetivo: Dado un número señalar el anterior y/o posterior (hasta el número nueve, incluyendo en cero).

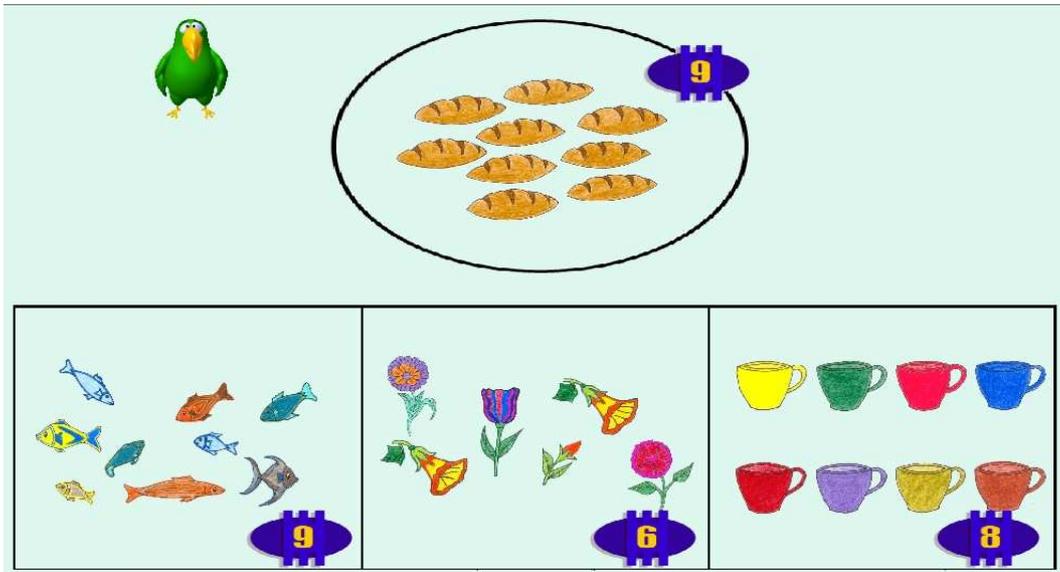
Ejemplo: Como se puede observar en la imagen la actividad es igual que la de nivel medio, pero los cardinales pueden llegar hasta el valor nueve.



2.4.3.3 Actividad Tipo 3

Objetivo: Etiquetar conjuntos con varios objetos y clasificarlos según el número de objetos (mayor que, menor que, igual que). Hasta el número nueve, incluyendo el cero.

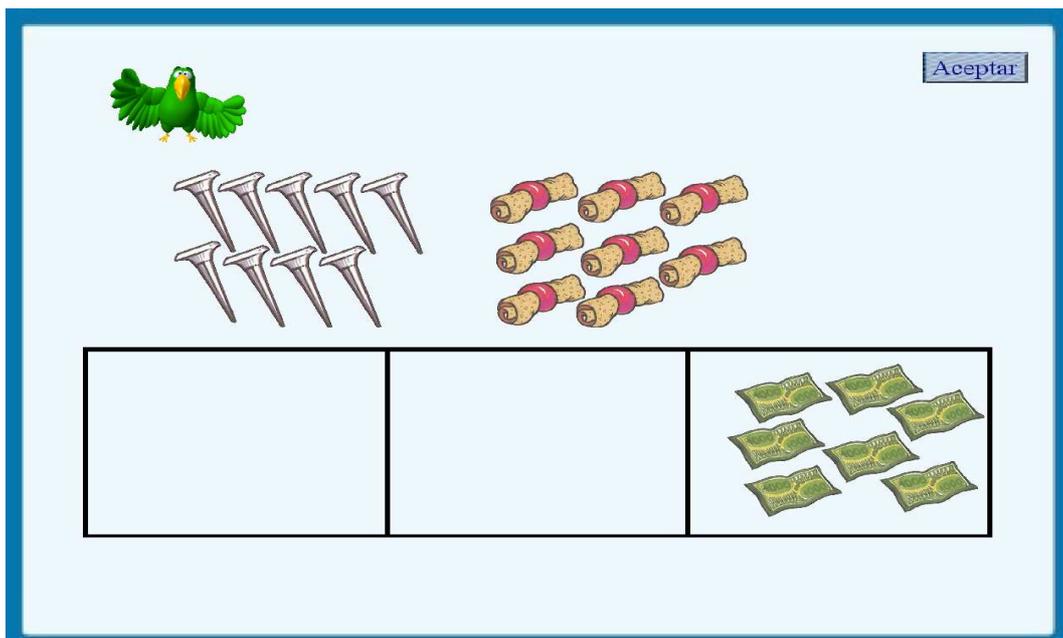
Ejemplo: La actividad es igual que las de nivel bajo y medio, pero la cantidad de elementos en las colecciones (y por tanto, los cardinales) pueden llegar hasta el nueve (incluyendo el cero).



2.4.3.4 Actividad Tipo 4

Objetivo: Cambiar la disposición física de colecciones de diferente número de objetos y ordenarlas (hasta el número nueve).

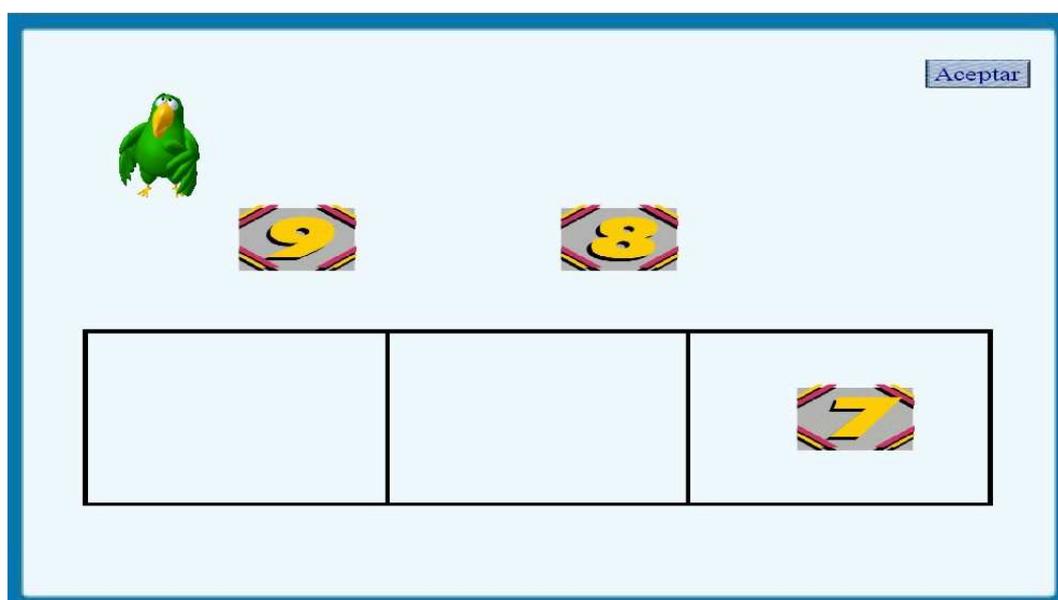
Ejemplo: Igual que la actividad de nivel medio, pero las colecciones pueden contener hasta nueve objetos.



2.4.3.5 Actividad Tipo 5

Objetivo: Ordenar una pequeña secuencia de números (hasta el número nueve, incluyendo en cero).

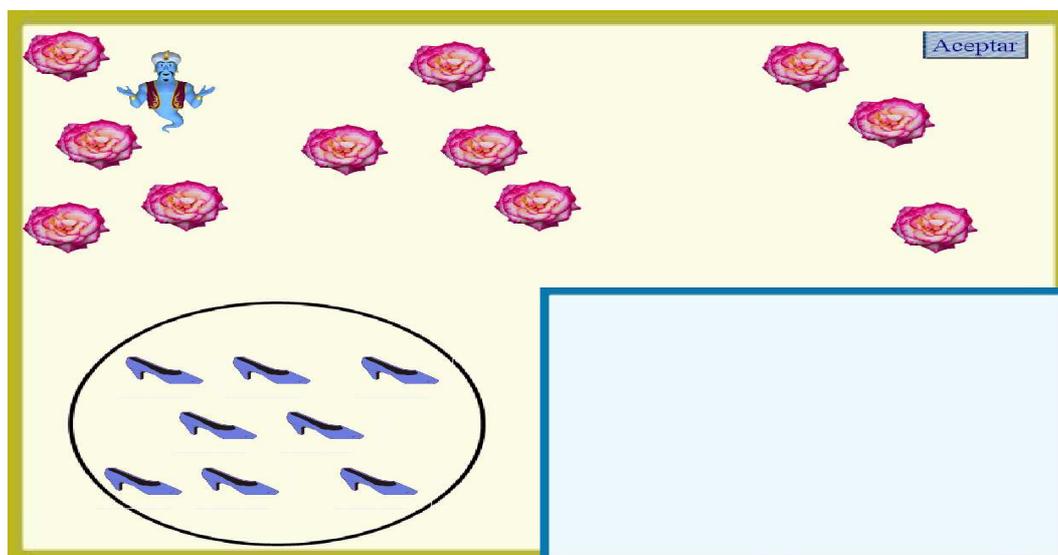
Ejemplo: Al igual que en el nivel medio, hay que ordenar los números de menor a mayor o de mayor a menor, colocándolos en el lugar que les corresponda dentro del recuadro negro. Los números pueden llegar hasta el nueve y no tienen por que ser correlativos.



2.4.3.6 Actividad Tipo 6

Objetivo: Construir un conjunto que contenga más, igual o menos elementos que uno dado (hasta el número nueve).

Ejemplo: Esta actividad es igual que las de niveles bajo y medio, pero la colección de referencia puede llegar a tener hasta nueve elementos.



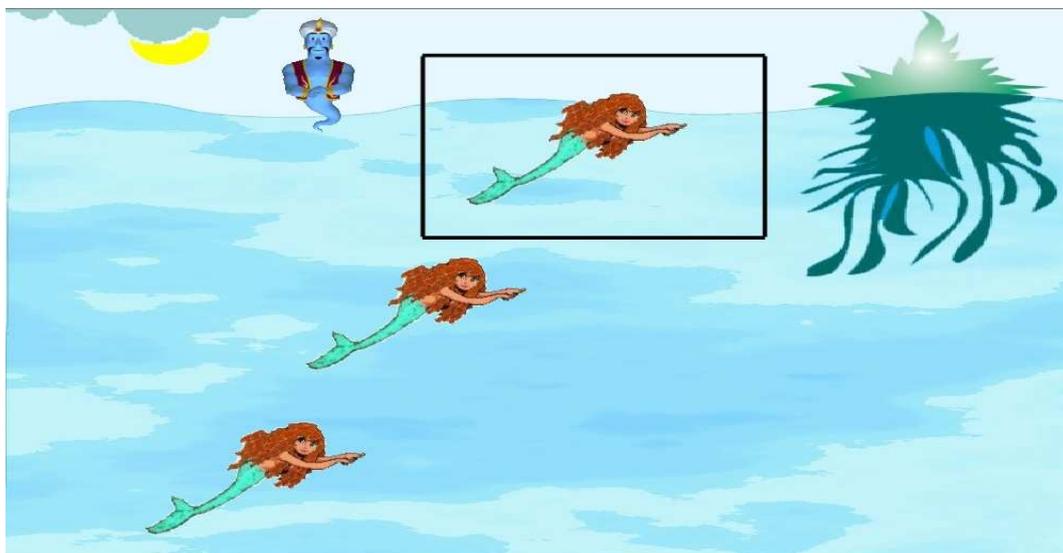
2.5 Ordinalidad

2.5.1 NIVEL BAJO

2.5.1.1 Actividad Tipo 1

Objetivo: Dada una secuencia de objetos indicar el primero y el último (hasta tres elementos).

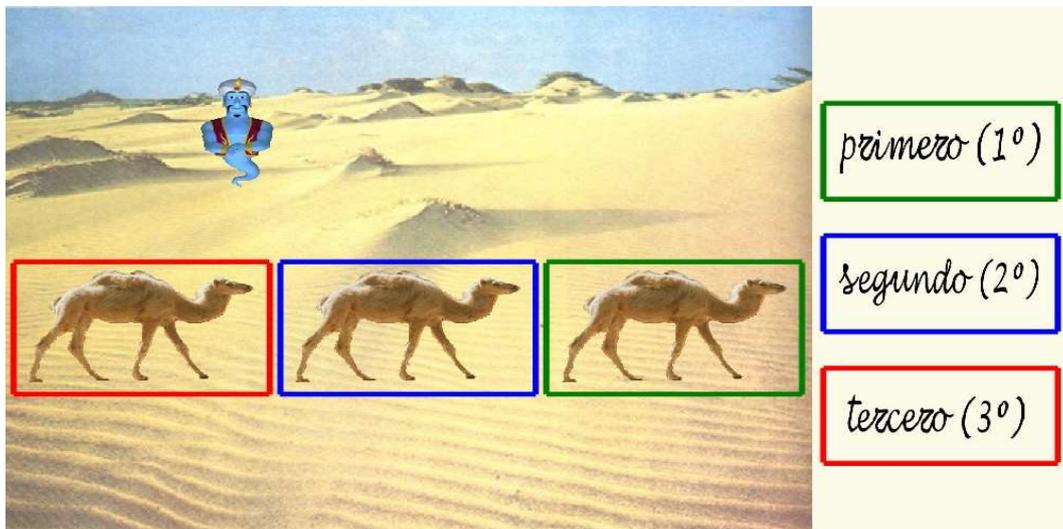
Ejemplo: Como se observa en la imagen inferior hay que señalar el primer o último elemento de las figuras que aparecen en la pantalla (las figuras son iguales para facilitar la resolución del ejercicio). En este caso el agente diría: "Señala la sirena que llega primero a la superficie". Como se puede ver en la imagen, ésta se marca con un recuadro.



2.5.1.2 Actividad Tipo 2

Objetivo: Dada una secuencia de objetos indicar el orden en que aparecen (hasta tres objetos).

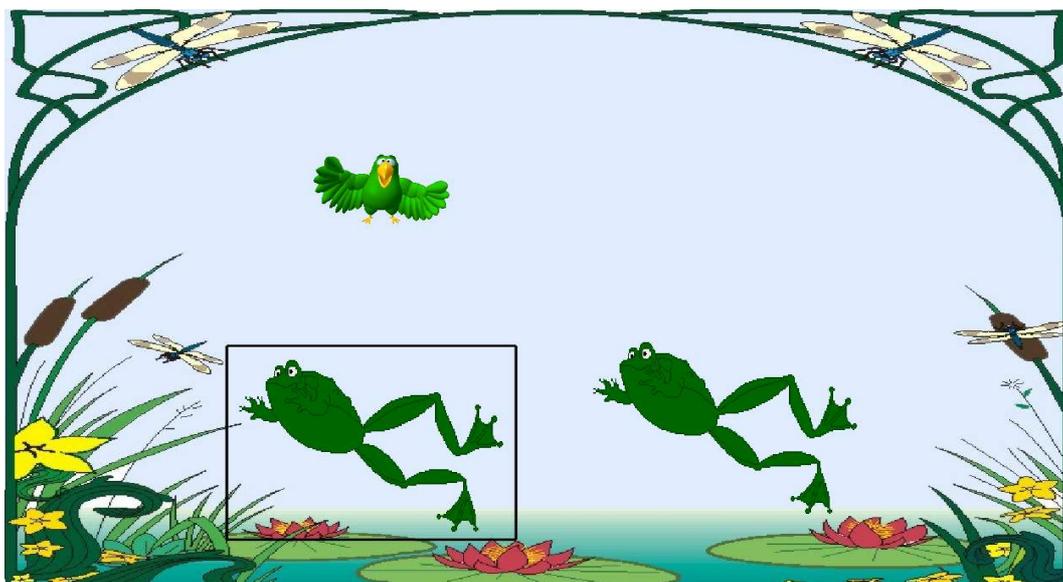
Ejemplo: Como se puede observar en la figura inferior, hay dos zonas en la pantalla. La de la izquierda contiene una imagen contextualizada en la que aparecen varios elementos siguiendo un orden (en este caso, tres camellos en fila) y la segunda, a la derecha, que contiene las posiciones que ocupan los elementos. Por tanto, en este ejemplo hay que unir los camellos de la fila con la posición que ocupan. Es decir, el primer camello de la fila con el símbolo de primero (1º), el segundo camello, con el 2º, y así sucesivamente. Para ello, hay que pinchar en los dos elementos que forman la pareja. Si el alumno realiza correctamente el emparejamiento se marcan las parejas con un recuadro del mismo color.



2.5.1.3 Actividad Tipo 3

Objetivo: Señalar el objeto del que se indique su posición (hasta tres objetos).

Ejemplo: En esta actividad hay que señalar la figura que se encuentre en la posición que indique el agente. Como en este caso se pregunta por la primera rana, al pincharla, ésta aparece marcada con un recuadro, como se observa en la siguiente imagen.



2.5.1.4 Actividad Tipo 4

Objetivo: Dada una colección de objetos, ordenarlos según las indicaciones que se den (hasta tres objetos).

Ejemplo: Como se puede observar en la imagen inferior, las indicaciones aparecen escritas en la pantalla. En este ejemplo habría que pinchar la palmera y colocarla donde aparece el símbolo 1º, el peluche en el 2º y el globo en el 3º. Una vez que el alumno piense que los ha colocado todos correctamente, se pulsa Aceptar para validar, y como en casos anteriores, si es correcto el agente felicita, y se pasa a la siguiente actividad, y si no, se vuelven a colocar las figuras en su posición inicial y se le da al alumno otra oportunidad.



La palmera en el primero
El peluche en el segundo
El globo en el tercero

1º	2º	3º
----	----	----

2.5.2 NIVEL MEDIO

2.5.2.1 Actividad Tipo 1

Objetivo: Dada una secuencia de objetos indicar el primero y el último (hasta seis objetos).

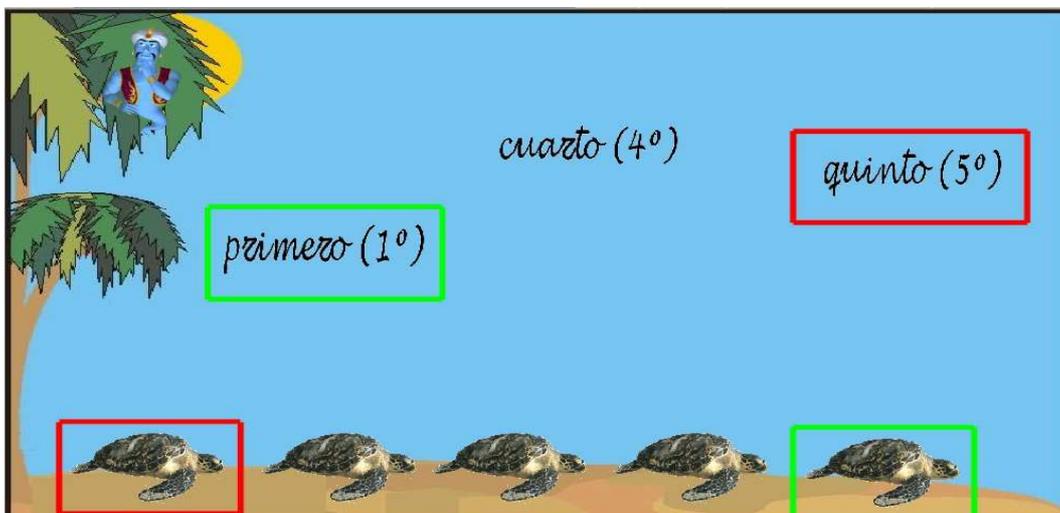
Ejemplo: La actividad se resuelve igual que la de nivel bajo, pero pueden aparecer hasta seis objetos en la pantalla.



2.5.2.2 Actividad Tipo 2

Objetivo: Dada una secuencia de objetos, indicar el orden en que aparecen (hasta el número seis objetos).

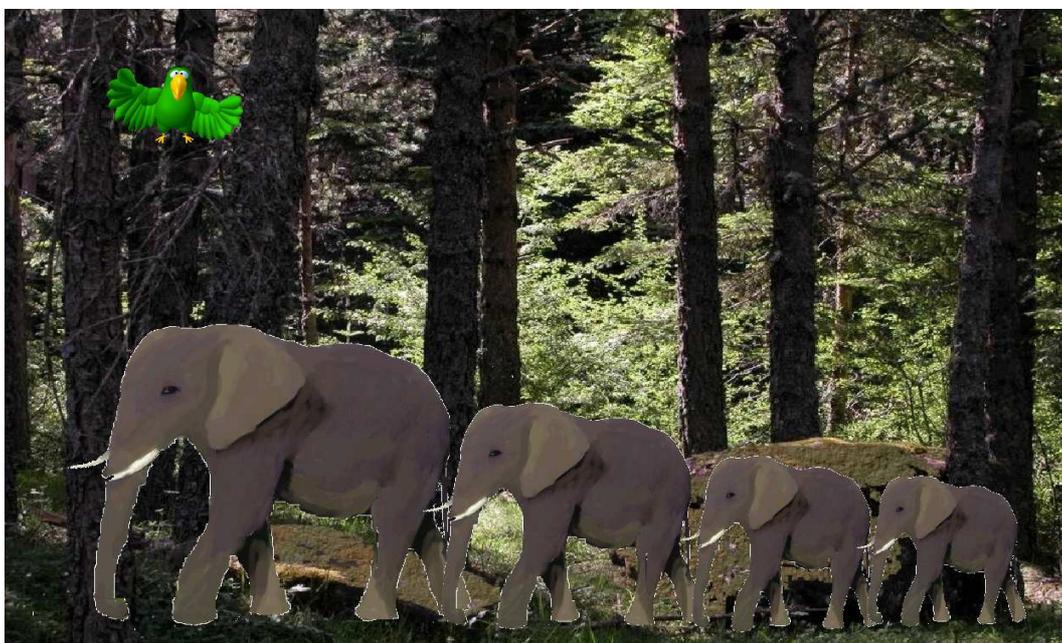
Ejemplo: La resolución de la actividad es como la de nivel bajo, pero ya no hay dos zonas diferenciadas de la pantalla, una con los objetos y otra con las posiciones, sino que todas pueden aparecer sobre el mismo fondo para aumentar la dificultad y, además, las posiciones no tienen que ser correlativas. El número de objetos puede llegar hasta seis.



2.5.2.3 Actividad Tipo 3

Objetivo: Señalar el objeto del que se indique su posición (hasta seis objetos).

Ejemplo: Al igual que en la actividad de nivel bajo, hay que señalar la figura que se encuentre en la posición que indique el agente, pero pueden haber hasta seis objetos en la pantalla.



2.5.2.4 Actividad Tipo 4

Objetivo: Dada una colección de objetos, ordenarlos según las indicaciones que se den (hasta seis objetos).

Ejemplo: Como se puede observar en la imagen, la actividad se realiza igual que la de nivel bajo, pero pueden existir hasta seis objetos para colocar en orden en el rectángulo.



Aceptar

*La azdilla en el primero
 El pájaro en el segundo
 La camiseta en el tercero
 El cubo en el cuarto
 La mariposa en el quinto*



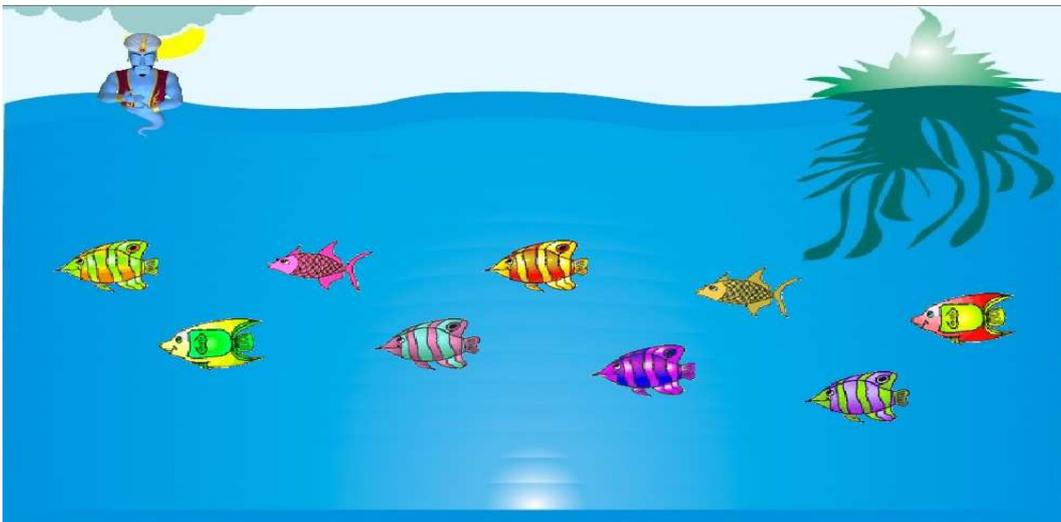
1º	2º	3º	4º	5º
----	----	----	----	----

2.5.3 NIVEL ALTO

2.5.3.1 Actividad Tipo 1

Objetivo: Dada una secuencia de objetos indicar el primero y el último (hasta nueve objetos).

Ejemplo: La actividad se realiza igual que las de nivel bajo y medio, pero pueden aparecer hasta un máximo de nueve objetos en la pantalla. No es necesario además que los objetos sean iguales.



2.5.3.2 Actividad Tipo 2

Objetivo: Dada una secuencia de objetos indicar el orden en que aparecen (hasta el número nueve objetos).

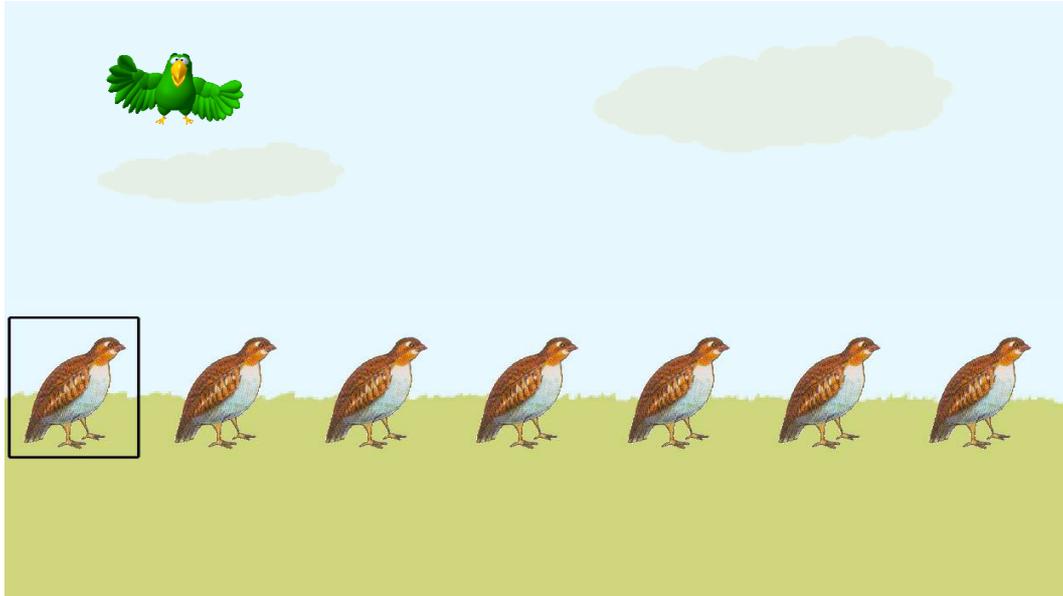
Ejemplo: Al igual que la actividad de nivel bajo y medio, hay que emparejar los objetos con la posición que ocupan. El número de objetos puede llegar hasta nueve y no tienen que ser iguales entre sí.



2.5.3.3 Actividad Tipo 3

Objetivo: Señalar el objeto del que se indique su posición (hasta nueve objetos).

Ejemplo: Al igual que en las actividad de nivel bajo y medio hay que señalar la figura que se encuentre en la posición que indique el agente, pero pueden haber hasta un máximo de nueve objetos.



2.5.3.4 Actividad Tipo 4

Objetivo: Dada una colección de objetos, ordenarlos según las indicaciones que se den (hasta nueve objetos).

Ejemplo: Como se puede observar en la imagen, la actividad se realiza igual que las de nivel bajo y medio, pero pueden existir hasta nueve objetos para su colocación.



*La pandereta en el primero
 El pato en el segundo
 La pelota en el tercero
 El pez en el cuarto
 La pluma en el quinto
 La regadera en el sexto
 El saco en el séptimo
 La sopa en el octavo
 El zapato en el noveno*

Aceptar



1º	2º	3º	4º	5º	6º	7º	8º	9º
----	----	----	----	----	----	----	----	----

2.6 Problemas

Todas las actividades de Problemas tienen como objetivo común resolver ejercicios de suma y resta con objetos, de combinación, cambio y comparación con dos números de un dígito. En primer lugar definiremos lo que son problemas combinación, cambio y comparación y luego en cada apartado la definición de los tipos de actividades y algunos ejemplos de ellos hasta ver todos los tipos.

Cambio: En los problemas de cambio se dispone de una cantidad inicial que cambia al aumentarse en otra determinada cantidad o disminuirse. En el primer caso es son problemas de cambio aumentando en los que a consecuencia del cambio la cantidad inicial se transforma por tanto en otra mayor, y por el contrario en los de cambio disminuyendo se produce una disminución en la cantidad inicial.

Combinación: Son problemas que muestran dos o más cantidades de elementos diferentes cuya suma o resta conduce al resultado final. La reunión de las dos por tanto es la solución del problema. Son estáticas ya que no cambian en el transcurso del tiempo como los de cambio.

Comparación: En estos problemas se comparan de forma estática dos cantidades que se dan simultáneamente (no como en el cambio) y no se corresponden a conjuntos incluidos en un todo (como en los problemas de combinación). La comparación se puede resolver por adición o por substracción.

2.6.1 NIVEL BAJO

2.6.1.1 Actividad Tipo 1

Objetivo: Resolver problemas de suma y resta con objetos, de combinación, cambio o comparación con dos números de un dígito sin escribir algoritmo, que no sumen más de tres.

Ejemplo de cambio aumentando: Son actividades en las que se produce una modificación de los objetos que aparecen en la pantalla con la que se inicia la actividad. A medida que la explicación del agente va progresando se incorporan

nuevos elementos a la actividad. Por ejemplo, en la siguiente actividad explicada por el agente: "En la fila de hormigas que sube por el árbol hay 2 hormigas y luego llegó otra hormiga ¿Cuántas hormigas hay en la fila?", inicialmente se presenta la pantalla de la izquierda, y cuando el agente dice el texto, se produce la aparición de la tercera hormiga. Para resolver el ejercicio hay que pinchar en el número que sea la solución a la pregunta planteada por el agente (en este caso tres hormigas).



2.6.1.2 Actividad Tipo 2

Objetivo: Resolver problemas de suma y resta con combinación, cambio o comparación con 2 números de un dígito y escribir el algoritmo horizontal o vertical, que no sumen más de 3.

Ejemplo de resta con combinación: Consta de una única pantalla dividida en dos partes. En la parte superior está escrito el algoritmo que tiene como incógnita el resultado y en la parte inferior la imagen que muestra el problema. Para solucionarlo hay que elegir entre los números que están dentro del recuadro negro, el resultado de la operación y moverlo al recuadro rojo que sirve como incógnita. En este ejemplo, la pregunta formulada por el agente es: "En el campo hay cuatro animales. Si uno es un perro, ¿cuántos gatos hay?". La solución es llevar el número 3 al recuadro rojo.

4 - 1 =

1 2 3

Aceptar

2.6.2 NIVEL MEDIO

2.6.2.1 Actividad Tipo 1

Objetivo: Resolver problemas de suma y resta con objetos, de combinación, cambio y comparación con 2 números de un dígito sin escribir algoritmo, que no sumen más de 6.

Ejemplo de suma con combinación: En la pantalla se muestra un problema que es explicado por el agente. En este caso: "En la pecera hay tres peces amarillos y uno rosa. ¿Cuántos peces hay en total?". Para solucionar el ejercicio hay que pinchar en el número correspondiente, que en este caso es el cuatro.



2.6.2.2 Actividad Tipo 2

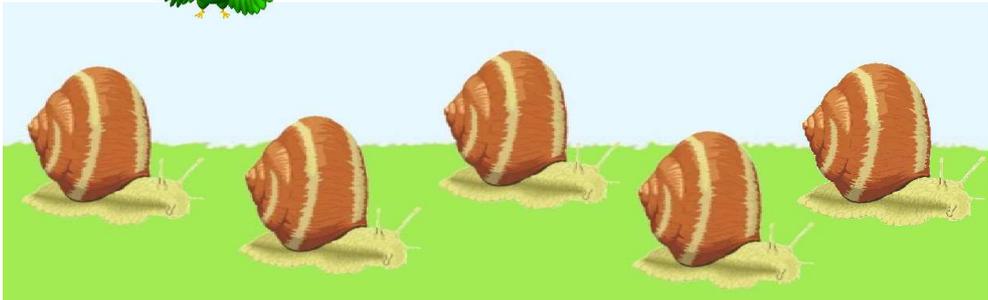
Objetivo: Resolver problemas de suma y resta con combinación, cambio o comparación con 2 números de un dígito y escribir el algoritmo horizontal o vertical, que no sumen más de 6.

Ejemplo de cambio disminuyendo: En la parte superior está escrito el algoritmo que tiene dos incógnitas (siendo siempre una de ellas el resultado) y en la parte inferior la imagen que muestra el problema. Al ser un problema de cambio disminuyendo desaparecen elementos de la actividad a medida que el agente va explicando la actividad. Por ejemplo, en la siguiente actividad la pregunta planteada por el agente es: "En el prado hay cinco caracoles y se va uno, ¿cuántos caracoles quedan?". La solución es llevar el número 1 al recuadro azul y el 4 al recuadro rojo.

Aceptar

$$5 - \square = \square$$

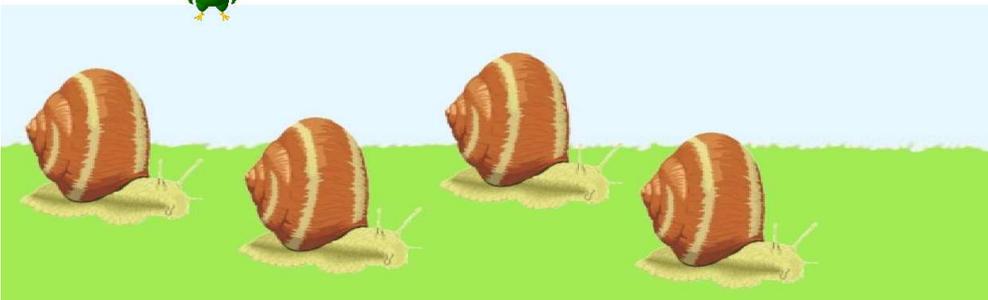
1	2	3
4	5	6



Aceptar

$$5 - \square = \square$$

1	2	3
4	5	6



2.6.3 NIVEL ALTO

2.6.3.1 Actividad Tipo 1

Objetivo: Resolver problemas de suma y resta con objetos, de combinación, cambio y comparación con 2 números de un dígito sin escribir algoritmo, que no sumen más de nueve.

Ejemplo de suma con comparación: Este es un problema de combinación en el que la explicación del agente es: *Juan tiene cuatro años. Si María tiene tres años más que Juan, ¿Coloca en la tarta tantas velitas como años tiene María?* Puesto que ya hay cuatro velitas en la tarta (que son los años que tiene Juan), hay que usar las velitas que están a la derecha de la pantalla y colocar tres encima de la tarta, para completar los años que tiene María, y pulsar en Aceptar para validar.



2.6.3.2 Actividad Tipo 2

Objetivo: Resolver problemas de suma y resta con combinación, cambio o comparación con 2 números de un dígito y escribir el algoritmo horizontal o vertical, que no sumen más de nueve.

Ejemplo de suma con combinación: Al igual que en las actividades de nivel bajo y medio, en la parte superior está escrito el algoritmo, pero en este caso hay tres incógnitas: un sumando, el símbolo de la operación (suma o resta) y el resultado. En este ejemplo la pregunta formulada por el agente es: "En el campo hay cuatro perros y cuatro pájaros. ¿Cuántos animales hay en total?". La solución es llevar el número 4 (número de perros) al recuadro azul, el símbolo de suma, al verde y el 8, al rojo que es el resultado.

2.7 Algoritmo

2.7.1 NIVEL MEDIO

2.7.1.1 Actividad Tipo 3

Objetivo: Hacer sumas y restas de 2 de números de un dígito (usando números hasta el seis, incluyendo el cero).

Ejemplo: Como se puede observar en la imagen en la parte izquierda de la pantalla hay un recuadro que contiene el algoritmo de la suma (o la resta) y un recuadro de color rojo donde debería estar el resultado. Al lado de cada operando hay un recuadro en el que el alumno puede colocar bolitas para ayudarse en el cálculo de la operación (esto es opcional). Para resolver el ejercicio hay que elegir un número de entre los que se encuentran en la parte inferior derecha de la pantalla, colocarlo en el recuadro rojo y pulsar en botón de Aceptar.

A cartoon character is on the left. The main area contains a math problem:

$$\begin{array}{r} 2 \\ + 3 \\ \hline \end{array}$$
 To the right of the numbers are empty boxes for the digits. Below the horizontal line is a red-outlined box for the answer.

To the right of the math problem is an oval containing 8 blue dots. Above it is a button labeled "Aceptar". Below the oval is a row of three boxes containing the numbers 4, 5, and 6.

2.7.2 NIVEL ALTO

2.7.2.1 Actividad Tipo 3

Objetivo: Hacer sumas y restas de 2 de números de un dígito (usando números hasta el nueve, incluyendo el cero).

Ejemplo: Este tipo de actividad es igual que la de nivel medio, pero se pueden usar los números hasta el nueve.

A cartoon character is on the left. The main area contains a math problem:

$$\begin{array}{r} 7 \\ - 2 \\ \hline \end{array}$$
 To the right of the numbers are empty boxes for the digits. Below the horizontal line is a red-outlined box for the answer.

To the right of the math problem is an oval containing 8 teal dots. Above it is a button labeled "Aceptar". Below the oval is a row of three boxes containing the numbers 4, 5, and 6.

APÉNDICE B. MANUAL DE USO

1 INTRODUCCIÓN

En este apéndice se pretende describir la instalación y el uso del Sistema Tutorial Inteligente (ITS) que se ha desarrollado para el refuerzo de los conceptos lógicos de número, suma y resta.

Está dirigido a padres y profesores que quieran utilizar el ordenador, núcleo de las nuevas tecnologías de la información y la comunicación, en la enseñanza.

En este documento se explica en primer lugar los objetivos para los que se ha diseñado y desarrollado el ITS. A continuación se comentan todos los pasos necesarios para la instalación del software y el manejo por parte de los usuarios en la aplicación (incluyendo la creación, modificación, borrado, etc. de alumnos). Posteriormente se explica el uso de la aplicación y por último se incluyen algunos ejemplos de los tipos de actividades que se pueden encontrar en la aplicación.

1.1 Objetivos

Se presenta en el manual, una definición general y una adaptada a este manual de un ITS, el cual puede definirse como un tutorial (un sistema que dirige la enseñanza) que incorpora técnicas de Inteligencia Artificial que permiten adaptar la materia a las características y expectativas de los estudiantes.

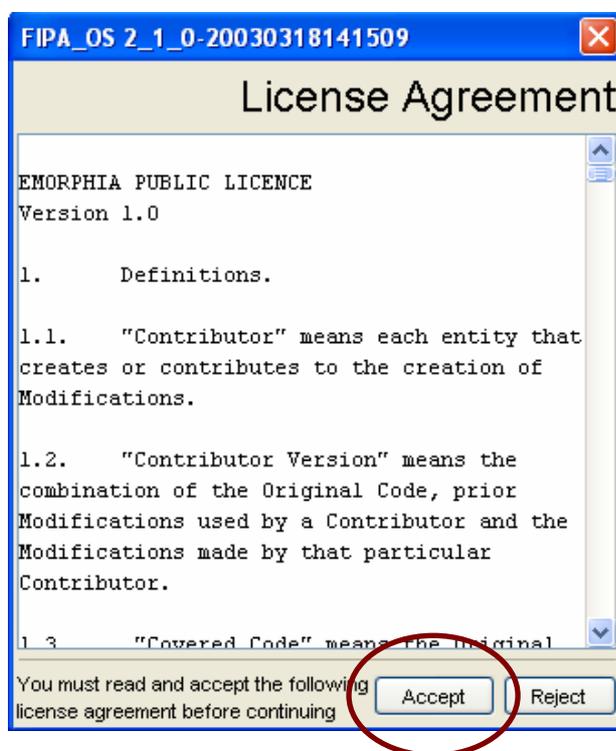
El tutorial, por lo tanto, sigue el proceso de enseñanza individualizada que consiste en determinar, a partir de unas características predeterminadas de los alumnos, cuáles son los objetivos de aprendizaje. A continuación se deben elaborar una serie de actividades a realizar por el alumno que le permita adquirir las habilidades marcadas en los objetivos. Este conjunto de actividades para un mismo objetivo no será estándar para todos los alumnos, sino que dependerá de las características de cada uno de ellos. Esto es, a partir de un alumno concreto se fijan los objetivos de aprendizaje y se considera una secuencia de acciones para lograr esos objetivos.

Además, en este software matemático tenemos en cuenta las características cognitivas de los alumnos, aprovechando los puntos fuertes de su perfil de aprendizaje. Para ello, se plantean actividades relacionadas con experiencias de su entorno cercanas a ellos y con un amplio número para no ser repetitivo.

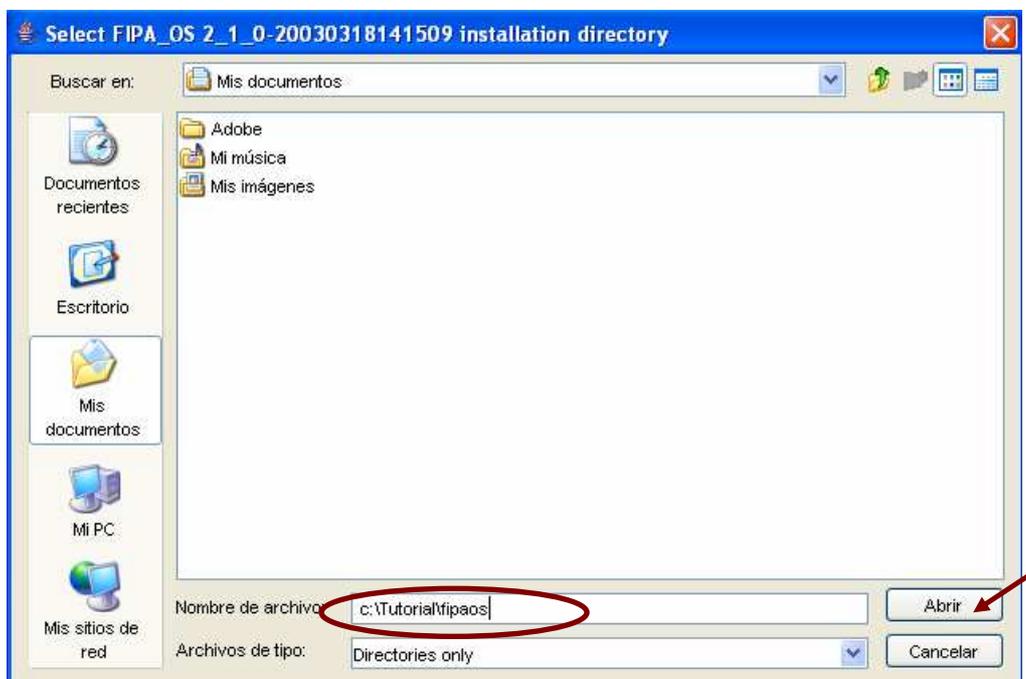
2 INSTALACIÓN

En primer lugar para instalar la aplicación se tienen que copiar los ficheros AddTutorial.exe, Instalacion.exe y FIPA_OS_Installer.jar en C:\.

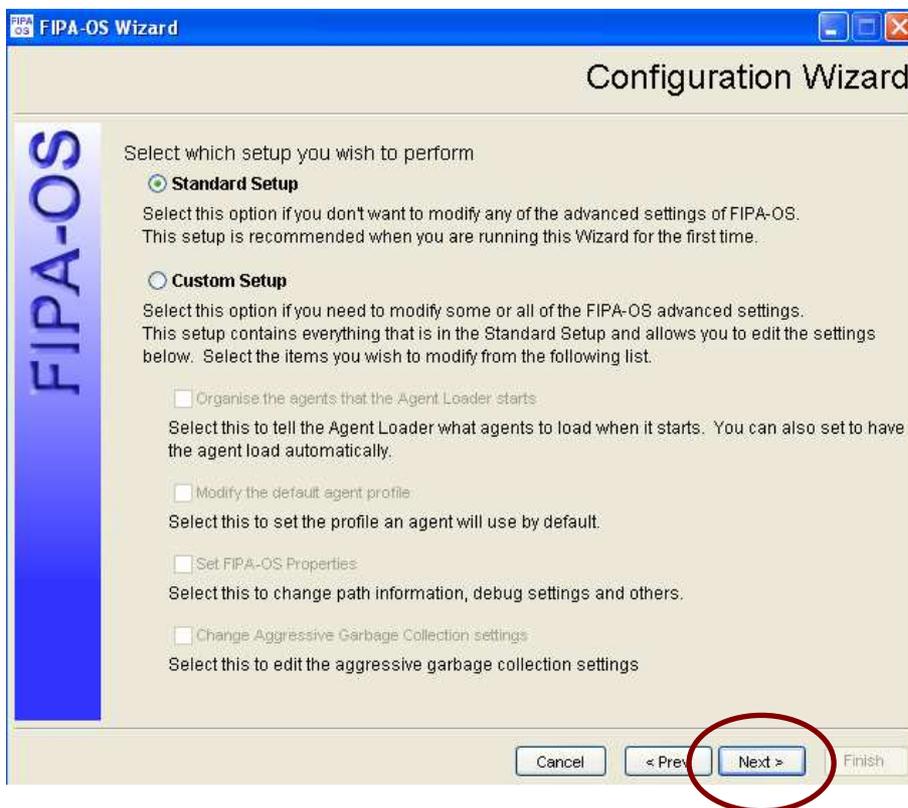
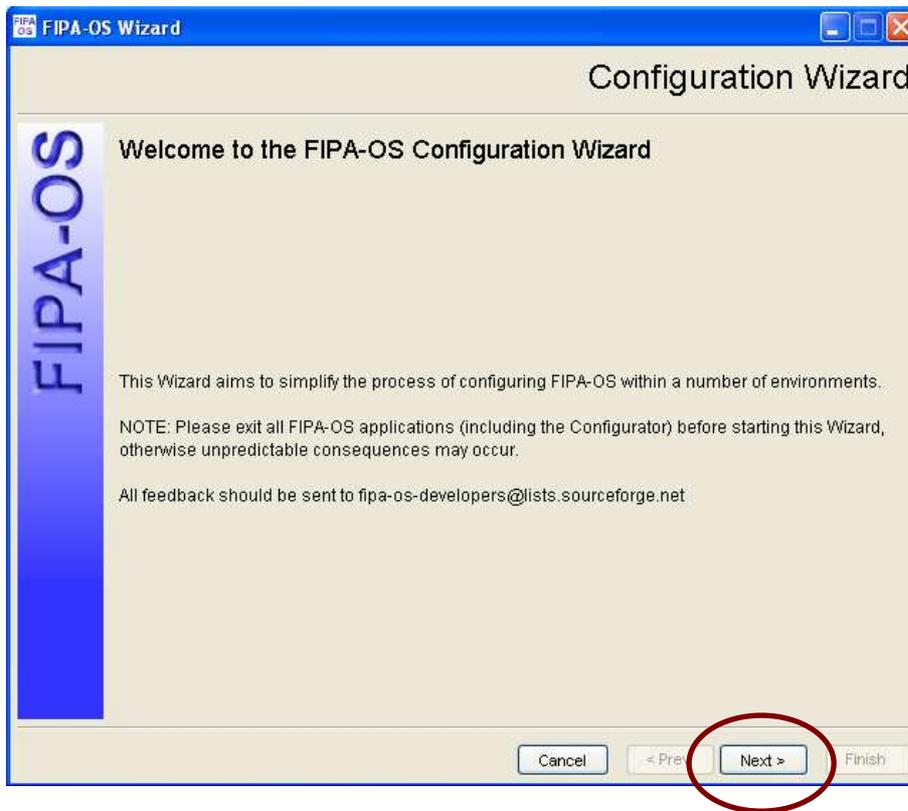
Hacer doble clic sobre FIPA_OS_Installer.jar para comenzar la instalación. En la pantalla aparecerá la aceptación licencia. Hay que pulsar el botón Accept para continuar la instalación.

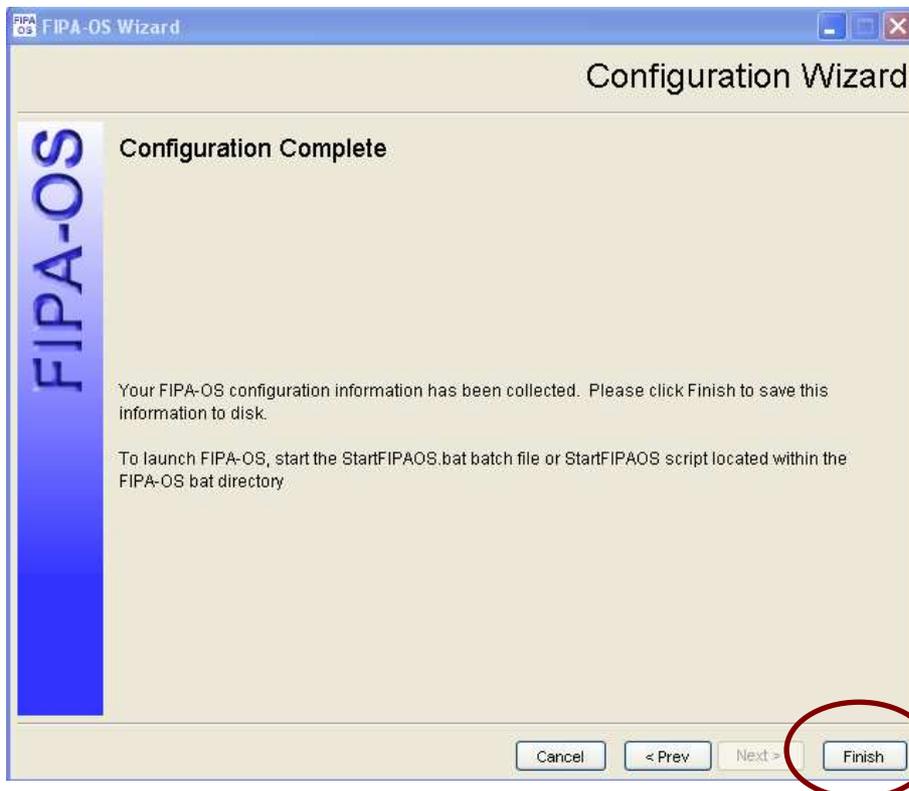
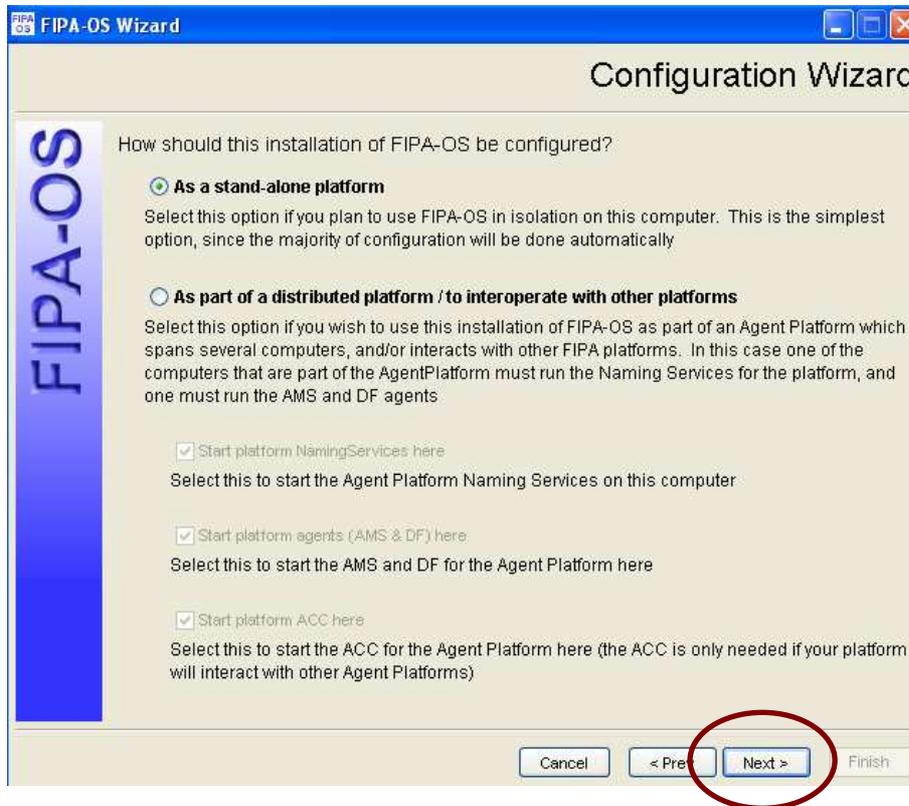


A continuación se verá una pantalla que pide la ruta en la que se va a instalar la aplicación. Poner C:\Tutorial\fipaos, como se ve en la imagen y pulsar en Abrir.

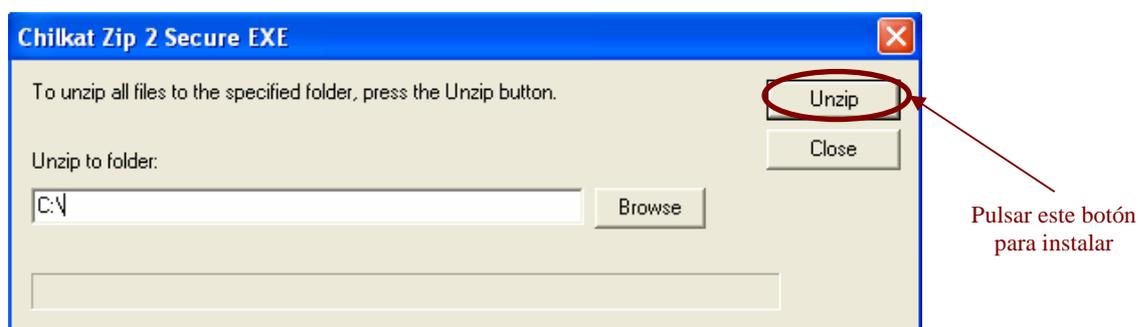


El programa comenzará a instalarse. Cuando finalice aparecerá un asistente de configuración como el que se ve en la figura. Las opciones ya aparecen marcadas por defecto en pantalla así que no es necesario modificar nada, si no simplemente, pulsar el botón **Next** hasta la última pantalla en la que deberemos pulsar **Finish**.





Una vez que se ha finalizado la instalación de la primera parte de la aplicación, se debe ir a C:\ y ejecutar AddTutorial.exe. En el caso de que aparezca cualquier mensaje pidiendo confirmación, contestar siempre afirmativamente (Sí, Sí a todo o Yes). En la pantalla que aparece, mantener como ruta por defecto para la instalación C:\Tutorial y pulsar en el botón **Unzip**, como se ve en la siguiente imagen.



2.1 *Requerimientos mínimos de instalación*

Sistema Operativo: Windows 98, Windows 2000, Windows XP o superior

RAM: 512 MB (1024 MB recomendado)

Espacio libre en disco: 1 Gb

Navegador: Internet Explorer 5.0 o superior

Base de datos: Microsoft Office Access

3 INSERCIÓN DE NUEVOS ALUMNOS

Para insertar nuevos alumnos se debe hacer doble clic sobre el icono de acceso directo, llamado Alumnos, que se encuentra en C:\Tutorial. Para facilitar el acceso, si se desea, se puede copiar dicho fichero al Escritorio.



En la pantalla que aparece se podrán ver los alumnos que ya están inscritos en la base de datos de la aplicación e insertar nuevos alumnos.

Inserción de Alumnos

id:

NombreApellidos:

Edad:

NivelEducativo:

Tipo:

Alumnos 1 de 3

3.1 Insertar nuevos alumnos

Pulsando en este símbolo  se podrá insertar un nuevo alumno. Los campos a rellenar son los siguientes (el campo id ya aparece relleno por defecto y no es necesario modificarlo):

- Nombre y Apellidos: Aquí se inserta el nombre y los apellidos del alumno, y son los datos que van a aparecer en el listado de alumnos de la pantalla de inicio.
- Edad: La edad del alumno, debe de ser un valor numérico.
- Nivel Educativo: Hace referencia al nivel educativo en el que se encuentra el alumno. Por ejemplo: Infantil, Primaria, etc.
- Tipo de alumno: Puede contener los valores 1, 2 y 3. Un alumno tipo 1 es un alumno con miedo al fracaso, un tipo 2 es un alumno hiperactivo y un tipo 3 es un alumno motivado al que no le afectan los errores.

Inserción de Alumnos

id: 4

NombreApellidos: María Rodríguez

Edad: 5

NivelEducativo: Infantil

Tipo: 1

Alumnos 4 de 4

Una vez que se ha rellenado la ficha del alumno para guardar la información en la base de datos es necesario pinchar en el botón . Por el contrario, si no se quieren guardar los datos habrá que pinchar en el botón de deshacer .

3.2 Moverse por las fichas de alumnos

Se podrá acceder a las fichas de los alumnos usando los iconos que aparecen en la barra. Con   podrá moverse entre los diferentes registros y con   ir al primer o al último registro, respectivamente.

Cuando un botón está de color gris significa que está deshabilitado y no es posible su uso. Si por ejemplo, el botón de desplazamiento hacia la derecha aparece como en la siguiente imagen , esto nos indica que estamos viendo el último registro de la base de datos y, por tanto, no es posible desplazarnos más registros hacia la derecha.

3.3 Modificar alumnos

Para modificar un alumno hay que ir a la ficha que se desea modificar (ver epígrafe 3.2 Moverse por las fichas de los alumnos) y cambiar los datos que se deseen. Una vez que se han modificado los datos para guardarlo en la base de datos, pinchar en el botón .

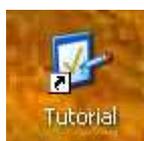
3.4 Eliminar alumnos

Para eliminar a un alumno se debe pulsar el botón  una vez que se esté mostrando su ficha en pantalla. Se pedirá confirmación antes de proceder a la eliminación del registro.

4 USO DE LA APLICACIÓN

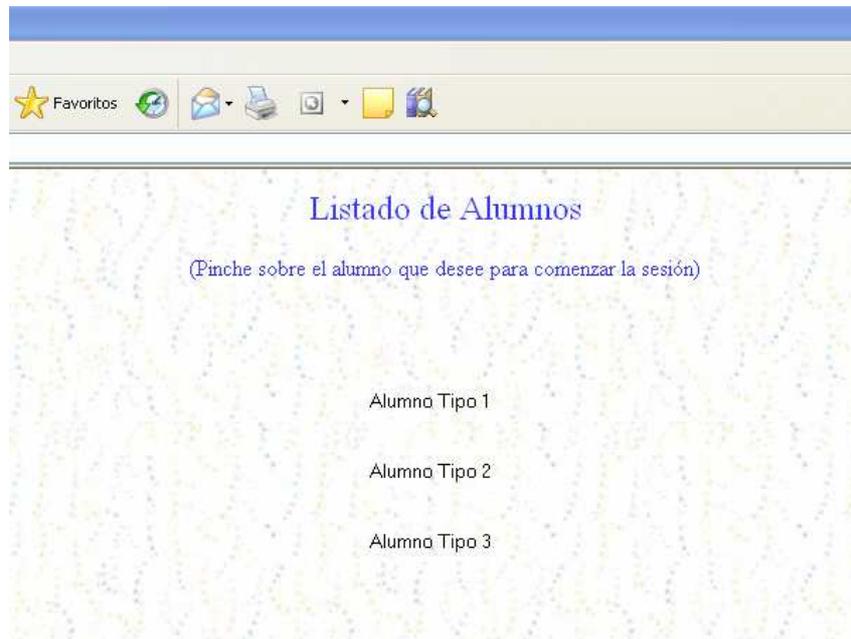
4.1 Inicio de la aplicación

Para comenzar a usar el tutorial, hacer doble clic sobre el icono de acceso directo llamado Tutorial que se encuentra en el C:\Tutorial. Para facilitar el acceso ha dicho icono, si se desea, se puede copiar al Escritorio (igual que en el caso del icono de acceso para el tratamiento de alumnos).

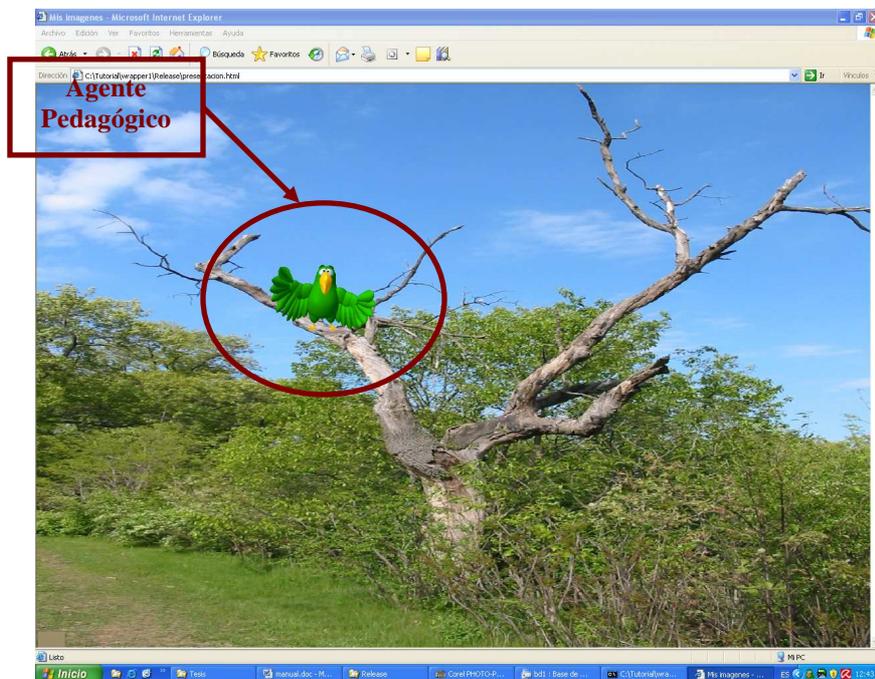


En la pantalla de inicio denominada 'Listado de Alumnos' aparecerán los tres tipos de alumnos en los que se pueden clasificar los usuarios de la aplicación:

- Alumno Tipo 1: Alumno con miedo al fracaso.
- Alumno Tipo 2: Alumno hiperactivo.
- Alumno Tipo 3: Alumno motivado, al que no le afectan los errores.



Al pinchar encima del alumno seleccionado da comienzo la aplicación. Si es la primera vez que se entra, aparecerá una pantalla de presentación y, posteriormente, dará comienzo la primera actividad. Como hay una gran variedad de actividades diseñadas en el Tutorial, estas irán variando en cada sesión.



4.2 Agentes Pedagógicos

En nuestro ITS, un agente pedagógico se encargará de ir guiando a los alumnos por las diferentes actividades que tienen que realizar. Los agentes pedagógicos interactúan y cooperan con el alumno, de una manera natural, para que una actividad sea ejecutada. Presentan el problema, guían la ejecución de la actividad y proporcionan estímulos positivos y/o negativos, (feedbacks) de acuerdo a los resultados obtenidos por el alumno en la resolución de la actividad. Sabemos que gracias a la utilización de estos agentes, la motivación del alumno se ve incrementada.

4.3 Dominio de la Aplicación

Cada ejecución del tutorial se compone de una secuencia de tres objetivos (por ejemplo clasificación, cardinalidad, correspondencia, etc.) cada uno de los cuales consta de dos actividades que se le presentarán al alumno. Es decir en total al alumno se le presentarán seis actividades en cada sesión.

Las respuestas del alumno a las actividades presentadas vienen dadas por las siguientes interacciones:

- 1.- pinchar en un objeto
- 2.- pinchar y mover

La interacción de pinchar y mover es una variación de la interacción de pinchar y arrastrar, ya que la acción de arrastrar les resulta difícil a los niños principiantes. Así pinchar y mover consiste en pinchar en un objeto, mover el objeto hasta el lugar deseado de la pantalla (sin tener pulsado el ratón, como en el caso de arrastrar) y volver a pinchar para colocarlo en la ubicación deseada de la pantalla. En las actividades de pinchar y mover se puede observar que hay un botón de **Aceptar** en la parte superior derecha de la pantalla. Una vez que se han movido los objetos y se han colocado en el lugar deseado hay que presionar dicho botón para que la actividad sea evaluada.



En las actividades de primer tipo, el paso a la siguiente actividad se realiza de forma automática, en cambio, en el segundo caso, es necesario pinchar en el botón Aceptar que aparece en la pantalla para que se pueda realizar la validación de la actividad y pasar a la siguiente.

Cuando se realizan seis actividades la sesión termina y es necesario ejecutar de nuevo la aplicación para continuar.

A medida que se vayan realizando las actividades el tutorial se encargará de ir ajustando el nivel de dificultad de las actividades a mostrar al alumno. El tutorial está estructurado en fases de forma que se puede avanzar o retroceder en las mismas en función de los resultados de la ejecución de las actividades por el alumno. Cada una de estas fases constará de objetivos a cubrir que son realizados en paralelo (las actividades a realizar por el alumno se pueden encuadrar en varios niveles de dificultad). Cuando el alumno realiza las actividades correspondientes a cada objetivo con un porcentaje adecuado de aciertos, puede pasar a la fase siguiente, caracterizada por otra serie de objetivos. Una vez superados éstos, se pasaría a la siguiente, y así sucesivamente.

Los objetivos de la fase 1 constan de dos niveles de dificultad: Poco y Alto, mientras que en los de la fase 2 podemos encontrar tres niveles de dificultad: Poco, Medio y Alto.

Las fases, y los objetivos dentro de cada fase, de los que se compone el Sistema Tutorial Inteligente son los siguientes:

FASE 1 – Lógica			
Clasificación	Relaciones de orden	Correspondencia término a término	Cuantificadores



FASE 2 – Concepto de número y suma con números de un dígito					
Contar y R. del número	Cardinalidad	Orden	Ordinalidad	Problemas	Algoritmo



FASE 3 – Concepto de número y suma sin llevar con números de dos dígitos				
Concepto de decena	Contar y R. del número	Orden	Problemas	Algoritmo

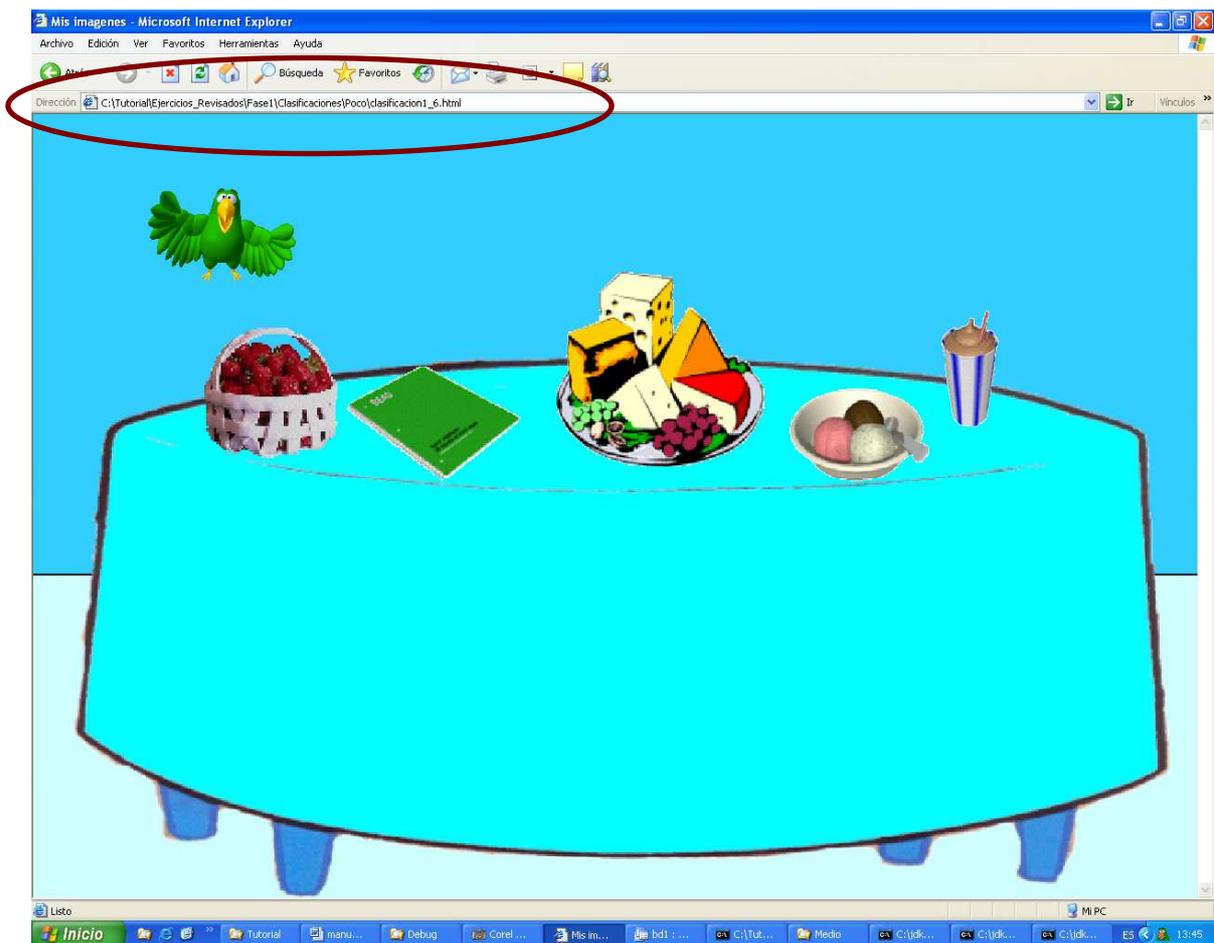


FASE 4 – Suma llevando con números de dos dígitos	
Problemas	Algoritmo

El texto que aparece en la visualización de la dirección en el explorador permite saber en todo momento en que fase se encuentra el alumno, de qué objetivo son las actividades que está realizando y en qué nivel de dificultad se encuentra dentro de ese objetivo (los niveles existentes son Poco, Medio y Alto). Por ejemplo, en la siguiente imagen vemos en la parte superior de la pantalla en la casilla dirección el texto:

C:\Tutorial\Ejercicios_Revisados\Fase1\Clasificaciones\Poco\clasificacion1_6.html

Esto indica que el alumno está en la Fase 1 del tutorial, realizando actividades del objetivo de Clasificaciones en el nivel de dificultad Poco y que esta actividad concreta que se esta realizando es la sexta del Tipo 1.



4.4 Resultados

Para ver los resultados de los alumnos se debe hacer doble clic sobre el icono de acceso directo llamado Resultados que se encuentra en C:\Tutorial (igual que en los casos anteriores si se desea copiar dicho acceso al Escritorio).

En esta página se muestran los datos personales de cada alumno y que ha obtenido en la ejecución del tutorial. El campo Fase indica en que fase se encuentra trabajando actualmente el alumno (como se ha mencionado anteriormente hay dos fases posibles Fase 1 y Fase 2). Para cada fase se muestran todos los objetivos que se trabajan en ella indicando para cada uno en que nivel de dificultad se encuentra el alumno (para la fase 1 los niveles son 1 = Poco y 2 = Alto y para la fase 2 son 1 = Poco, 2 = Medio, 3 = Alto) y que porcentaje de aciertos tiene en las actividades realizadas de ese nivel.

Resultados de los alumnos

Id:	<input type="text"/>
NombreApellidos:	<input type="text"/>
Edad:	<input type="text"/>
NivelEducativo:	<input type="text"/>
Tipo:	<input type="text"/>
Fase:	<input type="text"/>

Fase 1				
Clasificación	nivel:	<input type="text"/>	Acertos:	<input type="text"/>
Relaciones de Orden	nivel:	<input type="text"/>	Acertos:	<input type="text"/>
Correspondencia	nivel:	<input type="text"/>	Acertos:	<input type="text"/>
Cuantificadores	nivel:	<input type="text"/>	Acertos:	<input type="text"/>

Fase 2				
Contar y Rec del Num	nivel:	<input type="text"/>	Acertos:	<input type="text"/>
Cardinalidad	nivel:	<input type="text"/>	Acertos:	<input type="text"/>
Orden	nivel:	<input type="text"/>	Acertos:	<input type="text"/>
Ordinalidad	nivel:	<input type="text"/>	Acertos:	<input type="text"/>
Problemas	nivel:	<input type="text"/>	Acertos:	<input type="text"/>
Algoritmo	nivel:	<input type="text"/>	Acertos:	<input type="text"/>

⏪ ⏩

Como en el caso de la inserción de alumnos se podrá acceder a los resultados de los diferentes alumnos usando los iconos que aparecen en la barra. Con ⏪ ⏩ podrá moverse entre los diferentes registros y con ⏪ ⏩ ir al primer o al último registro respectivamente.

5 EJEMPLOS

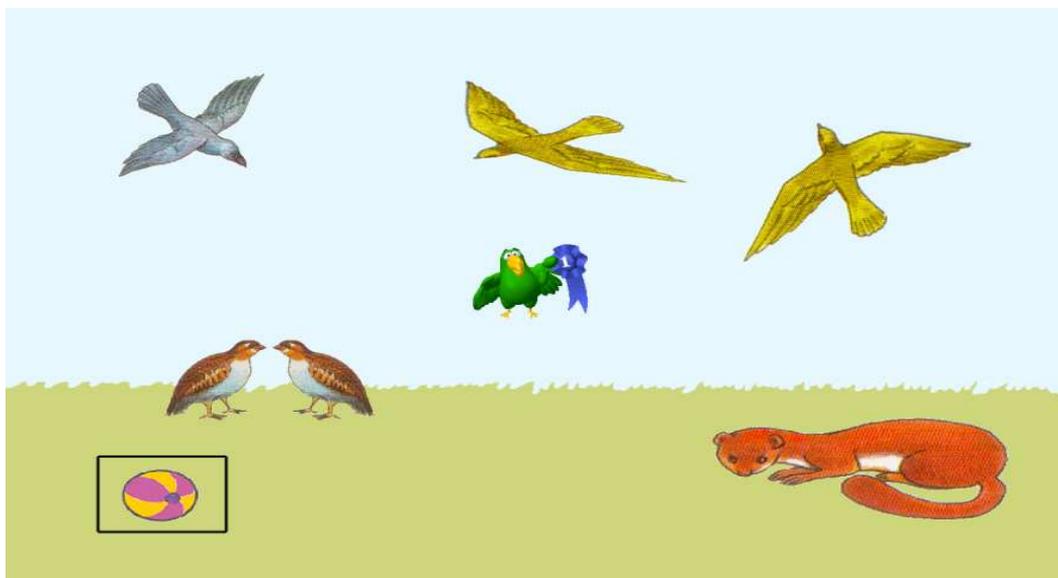
Vamos a ver algunos ejemplos de actividades y como se resuelven:

5.1 Tipo de Actividad Pinchar

5.1.1 CLASIFICACIÓN

Objetivo: Reconocer las características de un conjunto y separar elementos que no pertenezcan a él.

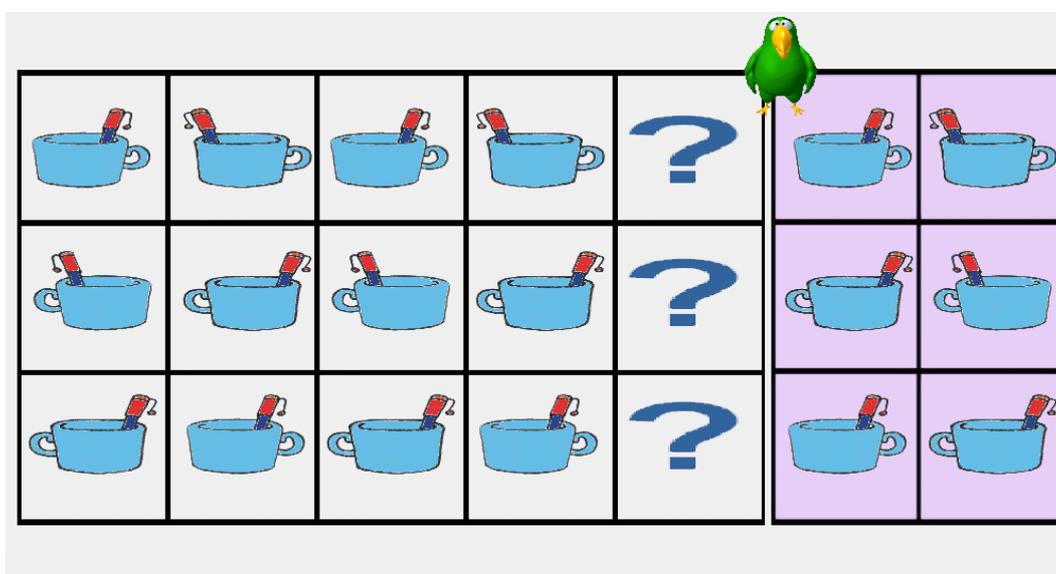
Ejemplo: Para realizar la actividad hay que pinchar sobre el elemento que no pertenezca al conjunto, el cual queda seleccionado mediante un recuadro de color negro. En el ejemplo vemos un grupo de animales y una pelota por lo que el elemento que no pertenece al conjunto es la pelota. Al pinchar sobre ella se marca con un recuadro y el agente pedagógico nos felicita indicándonos que hemos realizado la actividad correctamente.



5.1.2 RELACIONES DE ORDEN

Objetivo: Hacer seriaciones con alternancia de dos elementos y una incógnita en la que el niño tiene que señalar el objeto que seguiría en la serie.

Ejemplo: Para resolver la actividad, se tiene que pinchar primero en el interrogante, y luego, en uno de los dibujos situados en la parte derecha de la pantalla (sobre fondo malva). Si el resultado es correcto, el interrogante es sustituido por la figura en cuestión y el agente felicita al niño. En el caso contrario, no se produce ninguna modificación en las imágenes de la pantalla, pero el agente realiza un "feedback" correctivo y se le da al niño una segunda oportunidad.



5.1.3 CORRESPONDENCIA TÉRMINO A TÉRMINO

Objetivo: Relacionar dos conjuntos de objetos con una relación de igualdad.

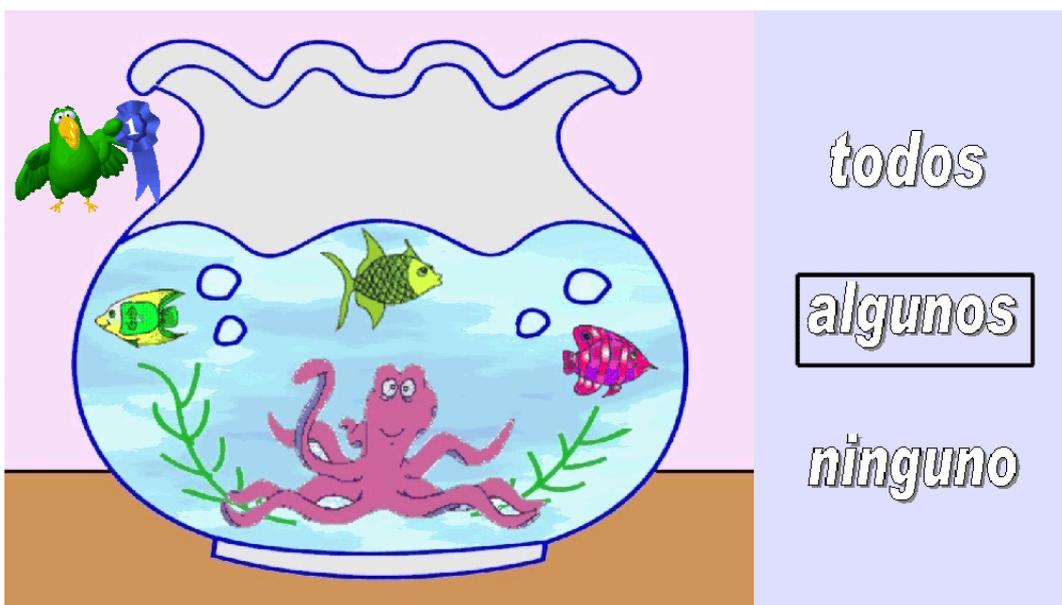
Ejemplo: Las actividades de este objetivo consisten relacionar dos conjuntos de objetos con una relación de igualdad como se puede ver en la figura siguiente. Si el alumno realiza correctamente el emparejamiento, se recuadran las parejas con un color y se da un feedback positivo. En caso contrario, no se marcan, se da feedback correctivo y otra oportunidad.



5.1.4 CUANTIFICADORES

Objetivo: Dado un conjunto de elementos ver si algunos de estos los elementos poseen una determinada característica (no todos).

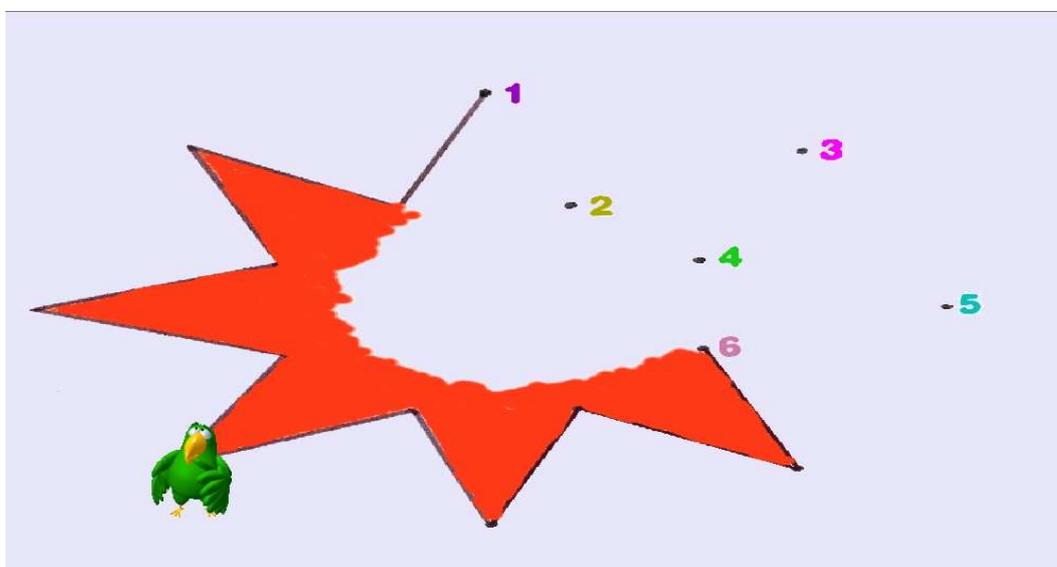
Ejemplo: En la figura se pregunta cuántos de los animales que se encuentran en la pecera tienen aletas. Si se acierta, la respuesta correcta se marca con un recuadro negro y, después de la felicitación del agente, se pasa automáticamente a la siguiente actividad.



5.1.5 CONTAR

Objetivo: Unir puntos de una serie.

Ejemplo: En este ejercicio hay que ir pinchando en orden los números para completar el dibujo. La resolución correcta es pinchar, en primer lugar sobre el número 1 (que es el que indica el comienzo de la parte del dibujo que falta), luego, en el 2, y así sucesivamente hasta el 6 para cerrar el dibujo. Las líneas irán apareciendo a medida que se pinchen en los números. Cuando esté completo, el agente felicita, y se pasa a la siguiente actividad.



5.1.6 CARDINALIDAD

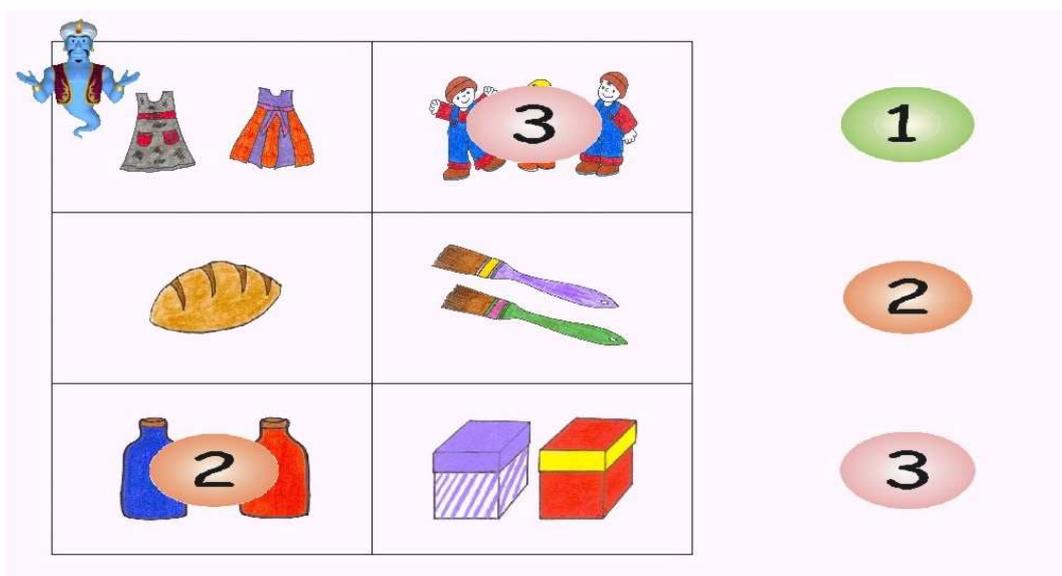
Objetivo: Al oír el cardinal de una determinada colección de objetos, saber relacionar de entre un conjunto de colecciones de diferentes tamaños, el (los) correspondiente (s) con dicho cardinal.

Ejemplo: Como se ve en la imagen, en este ejemplo, después de oír el número que diga el agente habría que pinchar en el (los) dibujo (s) que se encuentre en la zona inferior, el que tenga tantos objetos como el número que haya indicado el agente. Si el agente dice el número 1, en los pantalones, si dice el 2, en los tenedores y el 3, en los niños.



Objetivo: Relacionar cada una de las colecciones de un conjunto de colecciones de diferentes tamaños con los dígitos que indican su cardinal (menor número de dígitos que de colecciones).

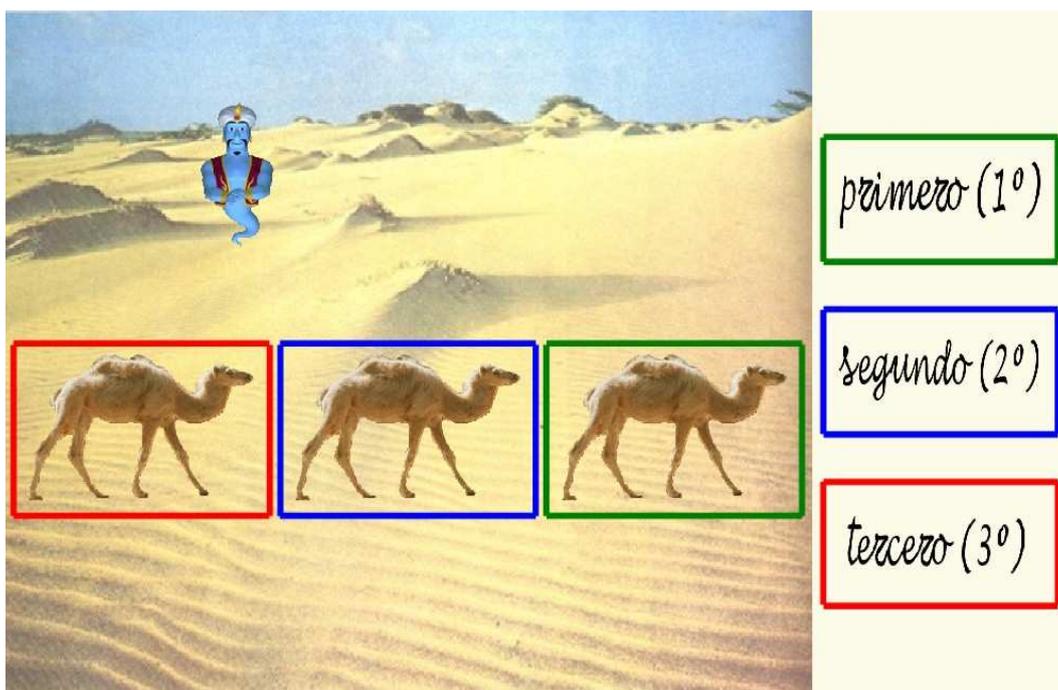
Ejemplo: En este tipo de actividades hay que pinchar, primero en una figura de la izquierda, y luego, en el número de la derecha que indica cuántos elementos hay en la figura que se ha pinchado. Como se puede ver en la imagen, cuando se acierta se coloca el número encima de la figura para indicar que esa figura ya está resuelta. Cuando se resuelven todas, se pasa a la siguiente actividad.



5.1.7 ORDINALIDAD

Objetivo: Dada una secuencia de objetos, indicar el orden en que aparecen.

Ejemplo: En la actividad que se ve en la siguiente imagen hay que unir los camellos de la serie con la posición que ocupan. Es decir, el primer camello de la fila, con el símbolo de primero (1º), el segundo camello, con el 2º y así sucesivamente. Para ello, hay que pinchar en los dos elementos que forman la pareja. Si el alumno realiza correctamente el emparejamiento, se marcan las parejas con un recuadro del mismo color y se da un feedback positivo. En caso contrario, no se marcan, el agente indica que no es correcto, y se le da al alumno otra oportunidad.

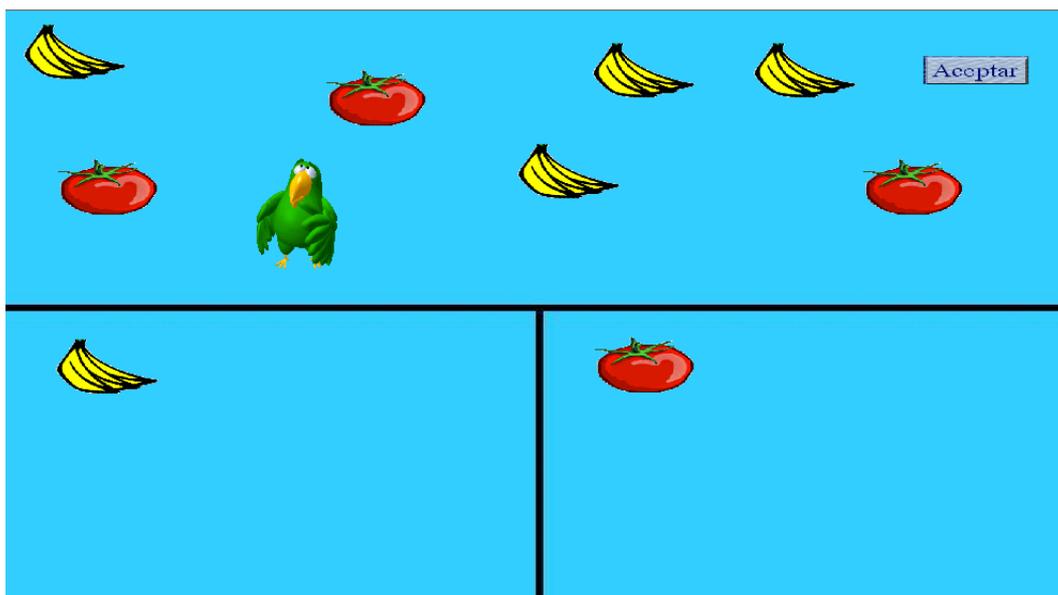


5.2 Tipo de Actividad Pinchar y Mover

5.2.1 CLASIFICACIÓN

Objetivo: Agrupar objetos idénticos entre sí.

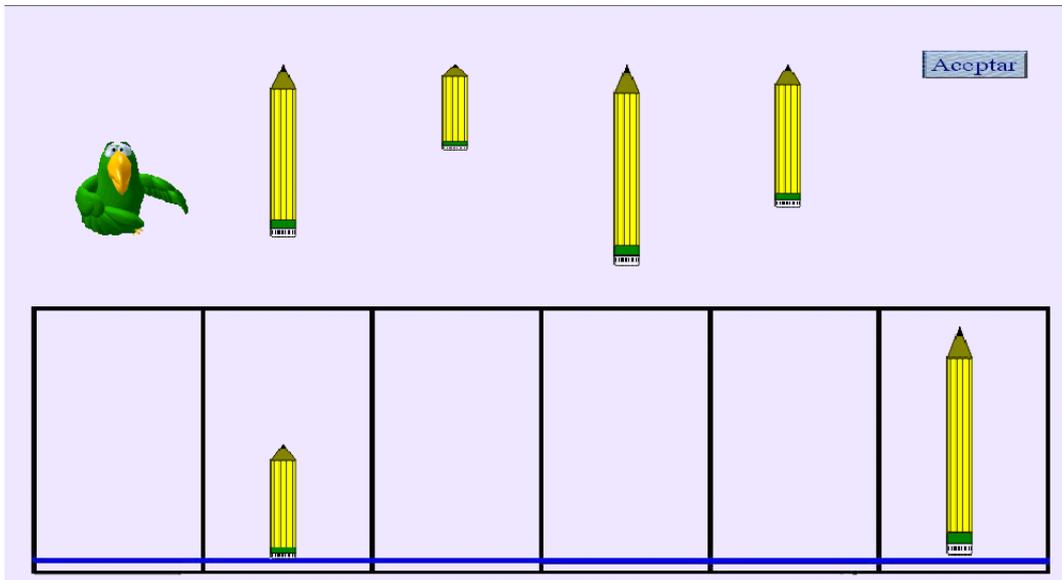
Ejemplo: Como se puede observar en la figura, esta actividad consiste en agrupar en cada uno de los recuadros de la parte inferior de la pantalla aquellas imágenes que sean iguales a las indicadas en estos recuadros. En el ejemplo los plátanos se colocarían en la parte inferior izquierda de la pantalla y los tomates en la inferior derecha. Una vez que se han colocado todos, hay que pulsar en Aceptar para que se realice la validación del ejercicio. Si es correcto, se pasa al siguiente ejercicio, y si no, se vuelven a colocar los objetos en su posición inicial y se da una nueva oportunidad para empezar el ejercicio.



5.2.2 RELACIONES DE ORDEN

Objetivo: Hacer seriaciones simples de mayor a menor, y viceversa.

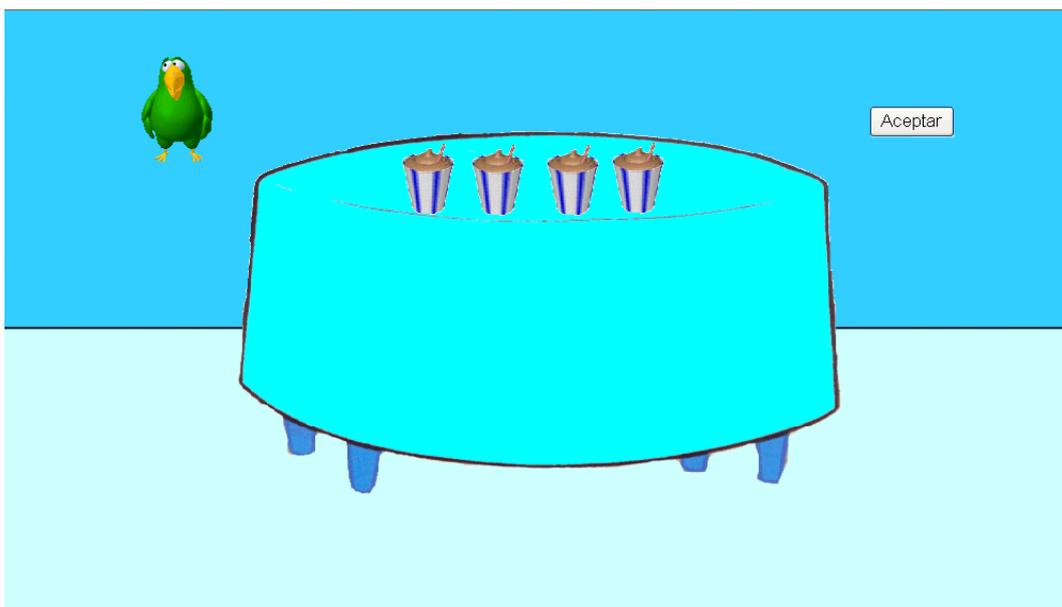
Ejemplo: En este ejemplo hay que colocar los lápices en su lugar correspondiente de menor a mayor. Igual que en el ejemplo anterior, hay que pulsar el botón Aceptar para validar el ejercicio. La marca azul que se puede observar en la figura se ha puesto como referencia base para que el niño pueda medir y comparar las alturas.



5.2.3 CUANTIFICADORES

Objetivo: Extraer elementos de un conjunto hasta que no quede ninguno.

Ejemplo: En la actividad que se ve en la figura hay que quitar todos los helados de encima de la mesa, hasta que no quede ninguno, y luego, pulsar Aceptar para validar el ejercicio.



5.2.4 ORDINALIDAD

Objetivo: Dada una colección de objetos, ordenarlos según las indicaciones que se den. Las indicaciones aparecen escritas en la pantalla.

Ejemplo: En este ejemplo de actividad de ordinalidad, habría que pinchar la palmera y colocarla donde aparece el símbolo 1º, el peluche, en el 2º y el globo, en el 3º. Una vez que el alumno piense que los ha colocado todos correctamente se pulsa Aceptar para validar y, como en casos anteriores, si es correcto el agente felicita y se pasa a la siguiente actividad, y si no, se vuelven a colocar las figuras en su posición inicial y se le da al alumno otra oportunidad.





Aceptar

*La palmeza en el primero
El peluche en el segundo
El globo en el tercero*

1º	2º	3º
----	----	----

APÉNDICE C. MANUAL DEL GENERADOR DE ACTIVIDADES

1. INTRODUCCIÓN

En este apéndice se describe la instalación y el uso del Generador de Actividades para la enseñanza y el refuerzo de los conceptos de número, suma y resta. Está dirigido a padres y profesores que quieran utilizar el ordenador como en la enseñanza núcleo de las nuevas tecnologías de la información y la comunicación.

Se explican, en primer lugar, los objetivos para los que se ha diseñado y desarrollado este Generador. A continuación, se comentan todos los pasos necesarios para la instalación del software y el uso de la aplicación, incluyendo algunos ejemplos de actividades.

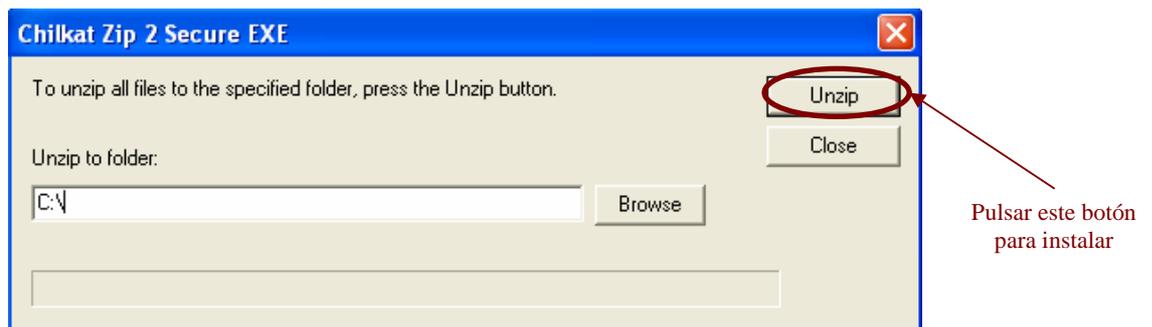
1.1 Objetivos

El objetivo del Generador es la creación de actividades para que los profesores puedan diseñar sus propios ejemplos para la enseñanza y el refuerzo de los conceptos de número, suma y resta. Se han incluido varios tipos de interacciones en las plantillas, para favorecer la versatilidad de los tipos de actividades que se puedan crear. Así, por ejemplo, hay plantillas para actividades de pinchar, de arrastrar, para ordenar elementos, etc.

2 INSTALACIÓN

Para instalar la aplicación se debe primero copiar los ficheros AddGenerador.exe, jdk140.exe y cocoon-2.1.5.1.exe en C:\

Hacer doble clic sobre AddGenerador.exe para comenzar la instalación. En el caso de que aparezca cualquier mensaje pidiendo confirmación, se tiene que contestar siempre afirmativamente (Sí, Sí a todo o Yes). En la pantalla que aparece a continuación, se debe mantener como ruta por defecto para la instalación C:\, y pulsar en el botón Unzip, como se ve en la siguiente imagen.



2.1 *Requerimientos mínimos de instalación*

Sistema Operativo: Windows 98, Windows 2000, Windows XP o superior

RAM: 512 MB (1024 MB recomendado)

Espacio libre en disco: 300 Mb

Navegador: Internet Explorer 5.0 o superior

3 USO DE LA APLICACIÓN

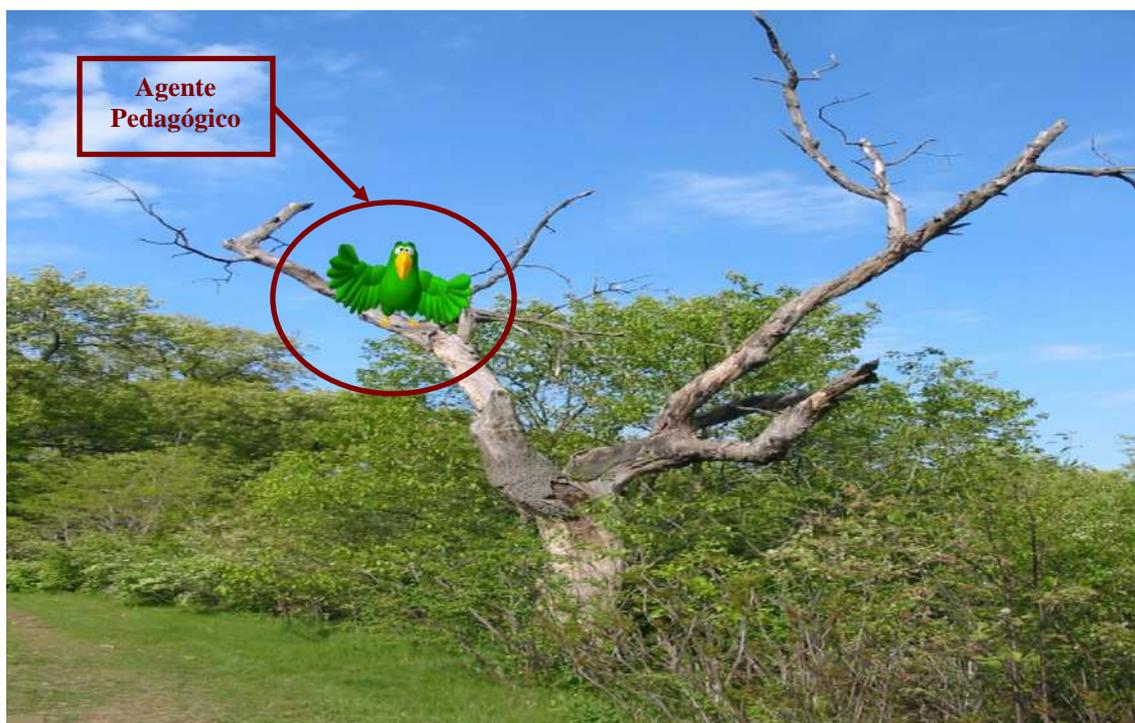
3.1 Inicio de la aplicación

Para comenzar a usar el Generador de Actividades hacer doble clic sobre el icono de acceso directo llamado Generador que se encuentra en el C:\cocoon-2.1.5.1. Si se desea se puede copiar dicho icono en el Escritorio, para facilitar el acceso.



3.2 Agentes Pedagógicos

En las actividades que se van a crear, un agente pedagógico se encargará de interactuar con los alumnos y explicarles la actividad que tienen que realizar. Los agentes pedagógicos interactúan y cooperan con el alumno de una manera natural para que una actividad sea ejecutada. De esta manera, se encargan de presentar el problema, guiar la ejecución de la actividad y presentar estímulos positivos y/o negativos (feedbacks) de acuerdo a los resultados obtenidos por el alumno en la resolución de la actividad. Se observa que gracias a la utilización de estos agentes, la motivación del alumno se ve incrementada.



3.3 *Generación de Actividades*

En la pantalla inicial: PÁGINA DE INICIO PARA LA GENERACIÓN DE PLANTILLAS, aparecerá una lista con las plantillas de las actividades que se pueden generar. Como se puede observar en la siguiente imagen, las plantillas se pueden generar usando archivos grabados en formato .wav, para usarlos como voz del agente pedagógico o usar la tecnología TTS (Text to Speech, es decir, la traducción del texto a voz sintetizada) para que el agente reproduzca los textos introducidos en el ordenador. Puesto que la plantilla que se usa en cada caso es diferente, la primera decisión que se deberá tomar en el Generador es si se graban las voces del agente pedagógico y se incorporan a la plantilla como un archivo .wav o si se usa TTS con lo que es necesario escribir el texto para que el agente lo reproduzca.

Todas las imágenes que se quieran incluir en las actividades se tienen que almacenar en la carpeta C:\Imágenes que se crea con la instalación del Generador.

PÁGINA DE INICIO PARA LA GENERACIÓN DE PLANTILLAS

Usando archivos .wav para el agente

[Pinchar](#)

[Arrastrar Auto](#)

[Arrastrar](#)

[Parejas](#)

[Ordenar Auto](#)

Usando Text to Speech para el agente

[Pinchar](#)

[Arrastrar Auto](#)

[Arrastrar](#)

[Parejas](#)

[Ordenar Auto](#)

Las posibles plantillas entre las que se puede elegir para crear actividades son:

1. Pinchar
2. Arrastrar Auto
3. Arrastrar
4. Parejas
5. Ordenar

A continuación se explicarán cada uno de estos tipos de plantillas.

3.3.1 PINCHAR

Son actividades en las que es necesario pinchar en uno, o más elementos, para que la actividad se considere correcta. La pantalla que se ve cuando se selecciona este tipo de actividad es la siguiente:

Plantillas de Pinchar

Imágenes:

Select	Id	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ancho	Alto	Ruta	Fijo	Fondo
<input type="button" value="Añadir Imagen"/>		<input type="button" value="Borrar imagen seleccionada"/>						

Agente:

Accion * ?

Select	Id	Posicion Eje X	Posicion Eje Y	Accion	Texto Explicacion
<input type="button" value="Añadir Explicacion Agente"/>		<input type="button" value="Borrar Explicacion seleccionada"/>			

Felicitation * ?

Correccion * ?

Siguiente Actividad:

Ruta * ?

Para insertar imágenes en nuestra actividad es necesario Pinchar en el botón Añadir Imagen, como se ve en la siguiente figura:

Imágenes:

Select	Id	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ancho	Alto	Ruta	Fijo	Fondo
<input type="button" value="Añadir Imagen"/>		<input type="button" value="Borrar imagen seleccionada"/>						

Esto nos permite abrir un formulario para incorporar una imagen a nuestra actividad. Se pueden incorporar tantas imágenes como se deseen, sólo teniendo en cuenta que éstas van a ir en capas superpuestas desde fondo de la pantalla, es decir, que la primera imagen que deberíamos crear sería la del fondo, puesto que si lo hacemos al revés y por ejemplo ponemos en último lugar el fondo, tapanía todas las demás imágenes que hubieramos puesto. Los campos que tenemos que rellenar para crear una imagen son los siguientes:

1. Id: Es el nombre o identificador de la imagen (por ejemplo: fondo, imagen1, imagen2, etc). El identificador es un valor único para cada imagen, por lo que no puede haber dos imágenes que tengan el mismo identificador.
2. Posición Inicial Eje X: Coordenada en el eje X de la pantalla donde se situa la imagen inicialmente. Son coordenadas relativas respecto a las dimensiones de la pantalla. El valor debe ser un número entre 0 (parte izquierda de la pantalla) y 100 (parte derecha de la pantalla).
3. Posición Inicial Eje Y: Igual que la anterior, pero en el eje Y.
4. Ancho: Anchura de la imagen en porcentaje relativo respecto al tamaño de la pantalla. El valor debe ser un número entre 0 y 100.
5. Alto: Altura de la imagen en porcentaje relativo respecto al tamaño de la pantalla. El valor debe ser un número entre 0 y 100.
6. Ruta: Nombre del fichero (incluyendo extensión) que contiene la imagen. Esta debe estar almacenada en la carpeta Imágenes.
7. Fijo: Indica si la imagen es solución o no. En las actividades de pinchar en un objeto, el campo fijo debe estar activado cuando la imagen no sea solución al problema. Es decir, que cuando la imagen sea solución de la actividad no se tendrá activado el valor fijo y habrá que activarlo en cualquier otro caso (cuando una imagen sea fondo siempre se debe activar también el valor fijo).
8. Fondo: Indica que la imagen es el fondo de la actividad. Puede haber uno o varios fondos. Por lo tanto, el campo fondo debe estar activado cuando la imagen sea el fondo de la actividad.

Imágenes:

Select	Id	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ancho	Alto	Ruta	Fijo	Fondo
<input type="checkbox"/>		<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>

Para ver un ejemplo hemos insertado tres imágenes fondo16, rana y reloj, siendo la primera el fondo de la actividad y las dos siguientes posibles soluciones al ejercicio. En este caso hemos decidido que la correcta sea la rana.

Imágenes:

Select	Id	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ancho	Alto	Ruta	Fijo	Fondo
<input type="checkbox"/>	fondo	0	0	100	100	fondo16.jpg	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	rana1	12	12	12	12	rana.gif	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	reloj	57	57	12	12	reloj.gif	<input checked="" type="checkbox"/>	<input type="checkbox"/>

A continuación, hay que rellenar el campo agente, que se corresponde con el agente pedagógico que queremos que explique e interactúe con el niño en la actividad. Los posibles agentes a elegir son:



Para elegir el agente se debe pinchar en el menú desplegable que está debajo del texto.

Una vez que hemos seleccionado alguno de estos agentes podemos definir las acciones que queremos que realicen en nuestra actividad. Igual que en el caso de las imágenes, podemos Insertar las explicaciones de los agentes pinchando en el botón Añadir Explicación Agente. Los campos que nos aparecen para rellenar son los siguientes:

1. Id: Es el identificador de la acción del agente (por ejemplo: accion1, accion2, exp1, saludo1, etc). El identificador es un valor único para cada acción por lo que no puede haber dos que tengan el mismo identificador.
2. Posición Eje X: Este campo indica la posición del agente en el eje X. El valor debe ser un numero entre 0 (parte izquierda de la pantalla) y 100 (parte derecha de la pantalla).
3. Posición Eje Y: Igual que campo anterior, pero en el Eje Y en lugar de en el X.
4. Acción: Indica la interacción a realizar por el agente. Las disponibles son:
 - a. Saludar: El agente saluda cuando comienza la actividad.
 - b. Explicar: Explicación del ejercicio.
 - c. Señalar hacia la derecha: El agente señala hacia la derecha de la pantalla.
 - d. Señalar hacia la izquierda: El agente señala hacia la izquierda de la pantalla.
 - e. Señalar hacia arriba: El agente señala hacia la parte superior de la pantalla.
 - f. Señalar hacia abajo: El agente señala hacia la parte inferior de la pantalla.
 - g. Texto Explicación: Es el mensaje que reproducirá el agente durante la acción elegida. Si en la pantalla inicial se seleccionó tecnología TTS, en este espacio, hay que escribir el texto que va a decir el agente, en cambio, si se está usando tecnología wav, hay que escribir la ruta del fichero que contiene el audio.

Además, tenemos que rellenar los feedbacks (felicitación y corrección) que dará el agente durante la resolución del ejercicio. Como su propio nombre indica, la felicitación es el texto que le dice el agente al alumno cuando resuelve correctamente el ejercicio, y la corrección es el texto que dice cuando realiza mal el ejercicio. Igual que en el texto de explicación, si en la pantalla inicial se seleccionó

tecnología TTS, aquí habrá que escribir el texto que va a decir el agente, en cambio si se usa tecnología wav, hay que escribir la ruta del fichero que contiene el audio. La ruta puede ser relativa (por ejemplo, saludo.wav) en cuyo caso el fichero de audio tiene que estar en el mismo directorio que la actividad resultante o absoluta (por ejemplo c:\\voces\\saludo.wav) en cuyo caso estará almacenado en la carpeta voces en la unidad c:.

En la siguiente imagen podemos ver un ejemplo de cómo se rellenan los campos para el agente:

Agente:

Accion Peedy * ?

Select	Id	Posicion Eje X	Posicion Eje Y	Accion	Texto Explicacion
<input type="checkbox"/>	accion1 * ?	30 * ?	20 * ?	Saludar * ?	Hola * ?
<input type="checkbox"/>	accion2 * ?	30 * ?	20 * ?	Explicar * ?	Señala la rana * ?

Anadir Explicacion Agente Borrar Explicacion seleccionada

Felicitation Muy bien * ?

Correccion Oh, oh * ?

Por último tenemos la parte que hemos llamado Siguiete Actividad. Como muestra el nombre se trata de indicar la siguiente actividad que se va a mostrar al alumno cuando se resuelva la que se está creando, en el caso de que se quiera hacer una secuenciación de actividades. Se debe escribir la ruta completa de la siguiente actividad (o relativa si están todas en el mismo directorio) teniendo que incluirse la extensión .html. Por defecto aparece el texto Ninguna en el campo, lo que indica que la actividad se ejecuta de forma independiente.

Siguiete Actividad:

Ruta Ninguna * ?

Enviar consulta

Una vez que hemos rellenido todos los campos, hay que pulsar en el botón Enviar consulta, para ver como ha quedado la actividad.

Plantillas de Pinchar

Imágenes:

Select	Id	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ancho	Alto	Ruta	Fijo	Fondo
<input type="checkbox"/>	fondo	0	0	100	100	fondo16.jpg	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	rana1	12	12	12	12	rana.gif	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	reloj	57	57	12	12	reloj.gif	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Agente:

Accion Peedy

Select	Id	Posicion Eje X	Posicion Eje Y	Accion	Texto Explicacion
<input type="checkbox"/>	leccion1	30	20	Saludar	Hola
<input type="checkbox"/>	leccion2	30	20	Explicar	Señala la rana

Felicitation Muy bien

Correccion Oh, oh

Siguiente Actividad:

Ruta Ninguna

El resultado para los valores que hemos definido es el siguiente:



Si una vez que se ha visto el resultado de la actividad se quiere hacer algún cambio, se puede volver a la pantalla anterior pulsando en el botón Atrás de la barra de herramientas del navegador.



Vamos, por ejemplo, a cambiar la posición Inicial de la rana poniéndola hacia la parte derecha de la pantalla, y un poco más hacia abajo de su posición anterior. Los valores que se han cambiado son los señalados en la siguiente imagen.

Plantillas de Pinchar

Imágenes:

Select	Id	Posición Inicial Eje X	Posición Inicial Eje Y	Ancho	Alto	Ruta	Fijo	Fondo
<input type="checkbox"/>	fondo	0	0	100	100	fondo16.jpg	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	rana1	80	27	12	12	rana.gif	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	reloj	57	57	12	12	reloj.gif	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Agente:

Accion Peedy * ?

Select	Id	Posicion Eje X	Posicion Eje Y	Accion	Texto Explicacion
<input type="checkbox"/>	accion1	30	20	Saludar	Hola
<input type="checkbox"/>	accion2	30	20	Explicar	Señala la rana

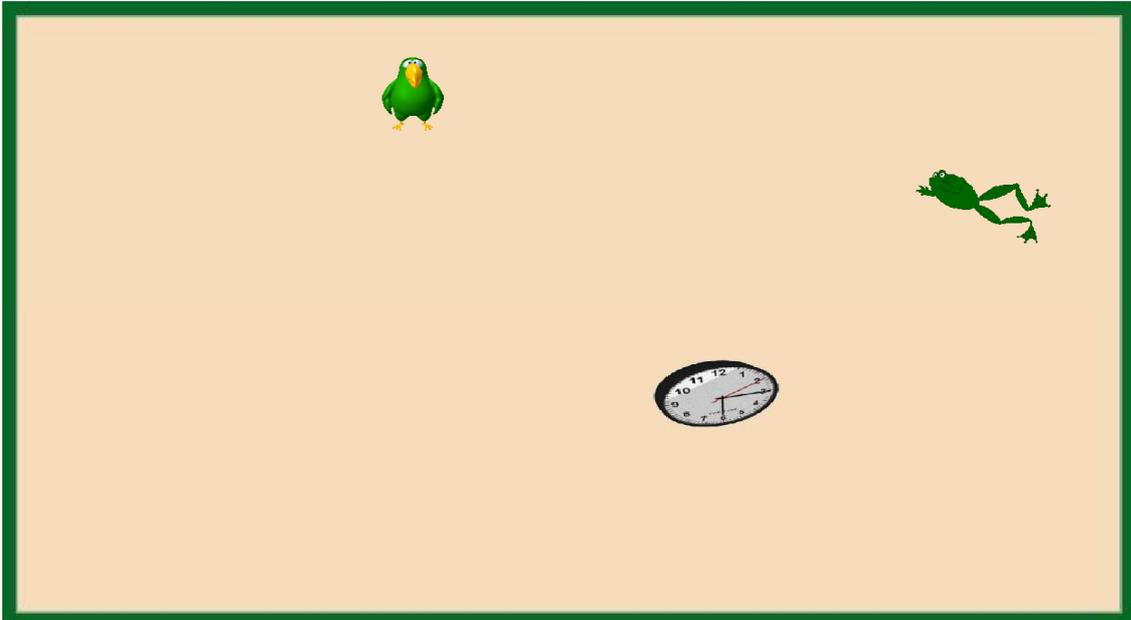
Felicitation Muy bien * ?

Correccion Oh, oh * ?

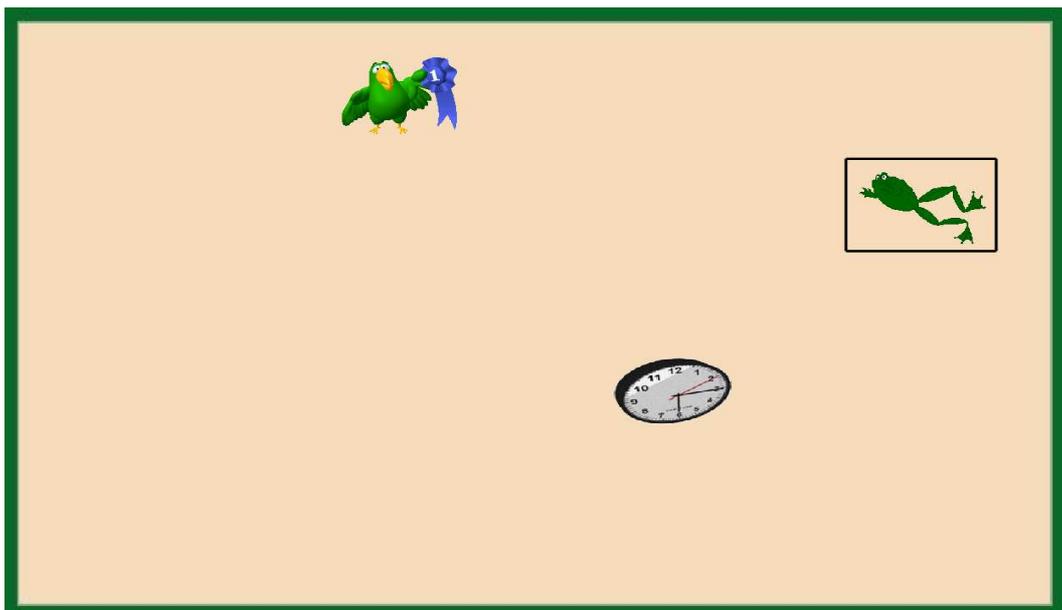
Siguiente Actividad:

Ruta Ninguna * ?

Con lo cual, si volvemos a pinchar en Enviar Consulta, la actividad queda como sigue:



En este ejemplo hemos modificado la posición inicial, sin embargo, se puede modificar cualquiera de los campos, tanto de las imágenes como del agente. Como hemos decidido que la rana sea la solución del ejercicio, si en la pantalla pinchamos sobre ella, vemos que aparece rodeada de un recuadro negro y el agente nos felicita. Si por el contrario nos hubiéramos equivocado y pinchado sobre el reloj, habría aparecido un feedback correctivo.



Si se quiere eliminar una imagen o una acción ya insertada habría que activar en el campo Select correspondiente a esa imagen o acción, y pinchar en el botón Borrar Imagen Seleccionada o Borrar Explicación Seleccionada. Por ejemplo, la siguiente figura muestra el borrado de la imagen del reloj.

Plantillas de Pinchar

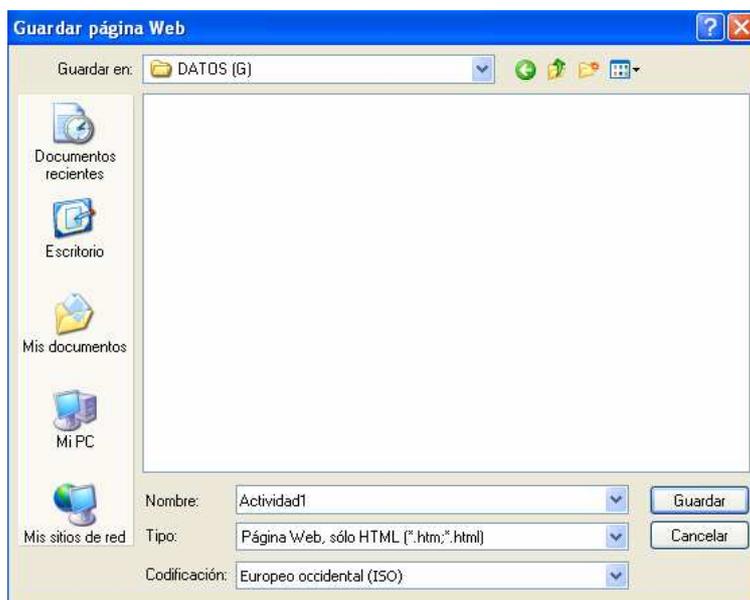
Imágenes:

Select	Id	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ancho	Alto	Ruta	Fijo	Fondo
<input type="checkbox"/>	fondo	0	0	100	100	fondo16.jpg	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	rana1	80	27	12	12	rana.gif	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	reloj	57	57	12	12	reloj.gif	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Anadir Imagen **Borrar imagen seleccionada**

Una vez que hemos creado la actividad, se guarda, haciendo uso del menú Archivo y eligiendo Guardar como. Cuando nos aparezca la pantalla para guardar la actividad, podemos buscar el directorio donde queremos almacenarla y en nombre poner el que queramos. La opción correcta para el tipo de fichero es la que aparece por defecto: Página Web, sólo HTML (*.htm, *.html).





3.3.2 ARRASTRAR AUTO (PINCHAR Y MOVER)

Aunque a esta actividad la llamamos Arrastrar en realidad se podría considerar pinchar y mover, ya que sólo arrastrar resulta más difícil para los niños. La modificación consiste en pinchar en un objeto, mover el objeto hasta el lugar deseado de la pantalla (sin tener pulsado el ratón, como en el caso de arrastrar) y volver a pinchar para colocarlo en la ubicación deseada de la pantalla.

El objetivo de estas actividades es que dada una serie de objetos colocados en posiciones variables de la pantalla, estos se tengan que colocar en el lugar que se indique.

En las actividades de pinchar y mover hay un botón de **Aceptar** en la parte superior derecha de la pantalla. Una vez que se han movido los objetos y se han colocado en el lugar deseado, hay que presionar dicho botón para que la actividad sea evaluada.

Aceptar

Para explicar la plantilla de este tipo de actividad vamos a definir sólo los campos nuevos o los que cambian con respecto a las plantillas de Pinchar ya que comparten una base común.

Plantillas de Arrastrar

Imágenes:

Select	Id	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ancho	Alto	Ruta	Arrea Correcta	Fijo	Fondo
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>
	* ?	* ?	* ?	* ?	* ?	* ?	* ?	?	* ?
<input type="button" value="Añadir Imagen"/> <input type="button" value="Borrar imagen seleccionada"/>									

Agente:

Accion * ?

Select	Id	Posicion Eje X	Posicion Eje Y	Accion	Texto Explicacion
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="Saludar"/>	<input type="text"/>
	* ?	* ?	* ?	* ?	* ?
<input type="button" value="Añadir Explicacion Agente"/> <input type="button" value="Borrar Explicacion seleccionada"/>					

Felicitation * ?

Correccion * ?

Siguiente Actividad:

Ruta * ?

El campo Área Correcta indica la imagen sobre la cual hay que colocar el objeto para que la actividad se considere correcta. Por ejemplo si tenemos varias ranas y relojes y queremos colocar las ranas en un recuadro, en la parte inferior de la pantalla, en el campo Área Correcta de cada rana, tiene que ir el Id con el que hayamos llamado al recuadro. Si no hay área correcta para una imagen, se escribe Ninguna en la casilla correspondiente.

El campo Fijo tiene que estar activado para los objetos que no se pueden mover por la pantalla. Por el contrario, para todos los que se puedan mover (sean, o no, solución al ejercicio) hay que dejar esta casilla sin marcar. Además para los que no sean solución es necesario activar también el campo Fondo.

Plantillas de Arrastrar

Imágenes:

Select	Id	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ancho	Alto	Ruta	Arrea Correcta	Fijo	Fondo
<input type="checkbox"/>	recuadro * ?	0 * ?	50 * ?	100 * ?	50 * ?	fondo16.jpg * ?	Ninguna * ?	<input checked="" type="checkbox"/> * ?	<input checked="" type="checkbox"/> * ?
<input type="checkbox"/>	rana1 * ?	12 * ?	12 * ?	12 * ?	12 * ?	rana.gif * ?	recuadro * ?	<input type="checkbox"/> * ?	<input type="checkbox"/> * ?
<input type="checkbox"/>	rana2 * ?	67 * ?	25 * ?	12 * ?	12 * ?	rana.gif * ?	recuadro * ?	<input type="checkbox"/> * ?	<input type="checkbox"/> * ?
<input type="checkbox"/>	reloj * ?	33 * ?	18 * ?	12 * ?	12 * ?	reloj.gif * ?	Ninguna * ?	<input type="checkbox"/> * ?	<input checked="" type="checkbox"/> * ?

Agente:

Accion: Genio * ?

Select	Id	Posicion Eje X	Posicion Eje Y	Accion	Texto Explicacion
<input type="checkbox"/>	accion * ?	45 * ?	45 * ?	Explicar * ?	Pon las ranas en el recu * ?

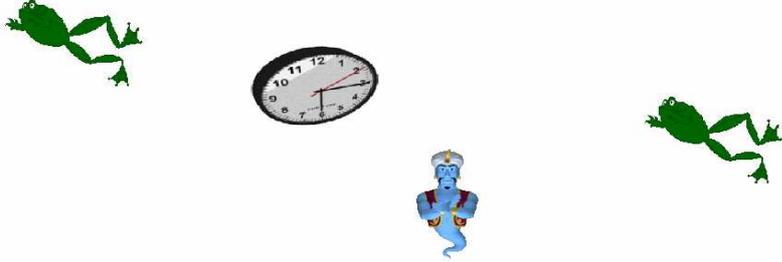
Felicitation: Muy bien * ?

Correccion: Oh Oh * ?

Siguiente Actividad:

Ruta: Ninguna * ?

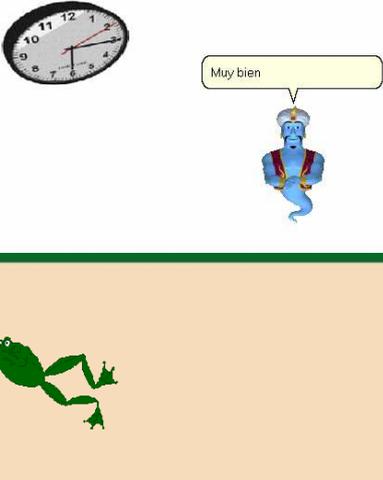
Aceptar



A screenshot of a software interface. The top section is white and contains four objects: a green frog on the left, a round clock in the center, a blue character with a crown in the middle, and another green frog on the right. In the top right corner, there is a blue button labeled "Aceptar". Below this section is a large, empty rectangular area with a light beige background and a dark green border.

Aceptar

Muy bien



A screenshot of a software interface. The top section is white and contains three objects: a round clock in the top left, a blue character with a crown in the top right, and a yellow speech bubble next to the character containing the text "Muy bien". In the top right corner, there is a blue button labeled "Aceptar". Below this section is a large rectangular area with a light beige background and a dark green border. In the bottom left corner of this area, there are two green frogs.

3.3.3 ARRASTRAR

Esta plantilla es igual que la del apartado anterior, pero en lugar de seleccionar la imagen sobre la que hay que colocar las imágenes para que el ejercicio se considere correcto se definen las áreas a mano. Para ello, se incorporan en el apartado de imágenes los siguientes campos:

- Ax1: Este campo indica la posición inicial del área en el que la imagen se considera correcta en el eje X. El valor debe ser un número entre 0 (parte izquierda de la pantalla) y 100 (parte derecha de la pantalla)
- Ax2: En este campo hay que indicar el ancho del área correcta en porcentaje relativo, respecto al tamaño de la pantalla. El valor debe ser un número entre 0 y 100.
- Ay1: Igual que Ax1, pero en el eje Y.
- Ay2: Igual que Ax2, pero hace referencia al alto del área.

Plantillas de Arrastrar

Imágenes:

Select	Id	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ax1	Ax2	Ay1	Ay2	Ancho	Alto	Ruta	Fijo	Fondo
<input type="button" value="Añadir Imagen"/>		<input type="button" value="Borrar imagen seleccionada"/>										

Agente:

Accion * ?

Select	Id	Posicion Eje X	Posicion Eje Y	Accion	Texto Explicacion
<input type="button" value="Añadir Explicacion Agente"/>		<input type="button" value="Borrar Explicacion seleccionada"/>			

Felicitation * ?

Correccion * ?

Siguiente Actividad:

Ruta * ?

3.3.4 PAREJAS (UNIR COLECCIONES)

Esta plantilla es como las de tipo Pinchar, pero en ella se pueden emparejar objetos. Por ello la plantilla es igual que la de Pinchar, pero se añade un campo llamado Pareja en el que hay que escribir el Id de la imagen con la que queremos emparejar la actual. Cuando se relacionan dos imágenes de forma correcta aparecerá un recuadro alrededor de ellas. El campo Color permite elegir el color de dichos recuadros.

Plantillas de Unir Colecciones

Imágenes:

Select	Id	Color	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ancho	Alto	Ruta	Pareja	Fijo	Fondo
<input type="checkbox"/>	* ?	amarillo	* ?	* ?	* ?	* ?	* ?	* ?	<input type="checkbox"/>	<input type="checkbox"/>
Anadir Imagen		Borrar imagen seleccionada								

Agente:

Accion * ?

Select	Id	Posicion Eje X	Posicion Eje Y	Accion	Texto Explicacion
Anadir Explicacion Agente		Borrar Explicacion seleccionada			

Felicitation * ?

Correccion * ?

Siguiente Actividad:

Ruta * ?

Por ejemplo en la siguiente pantalla vamos a colocar dos ranas y dos relojes y tendremos que unir los objetos que sean iguales.

Imágenes:

Select	Id	Color	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ancho	Alto	Ruta	Pareja	Fijo	Fon
<input type="checkbox"/>	rana1	verde	12	12	12	12	rana.gif	rana2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	rana2	verde	67	84	12	12	rana.gif	rana1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	reloj1	azul	25	54	12	12	reloj.gif	reloj2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	reloj2	azul	68	35	12	12	reloj.gif	reloj1	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Anadir Imagen Borrar imagen seleccionada

Agente:

Accion **Merlin** * ?

Select	Id	Posicion Eje X	Posicion Eje Y	Accion	Texto Explicacion
<input type="checkbox"/>	accion1	50	50	Explicar	Une los objetos iguales

Anadir Explicacion Agente Borrar Explicacion seleccionada

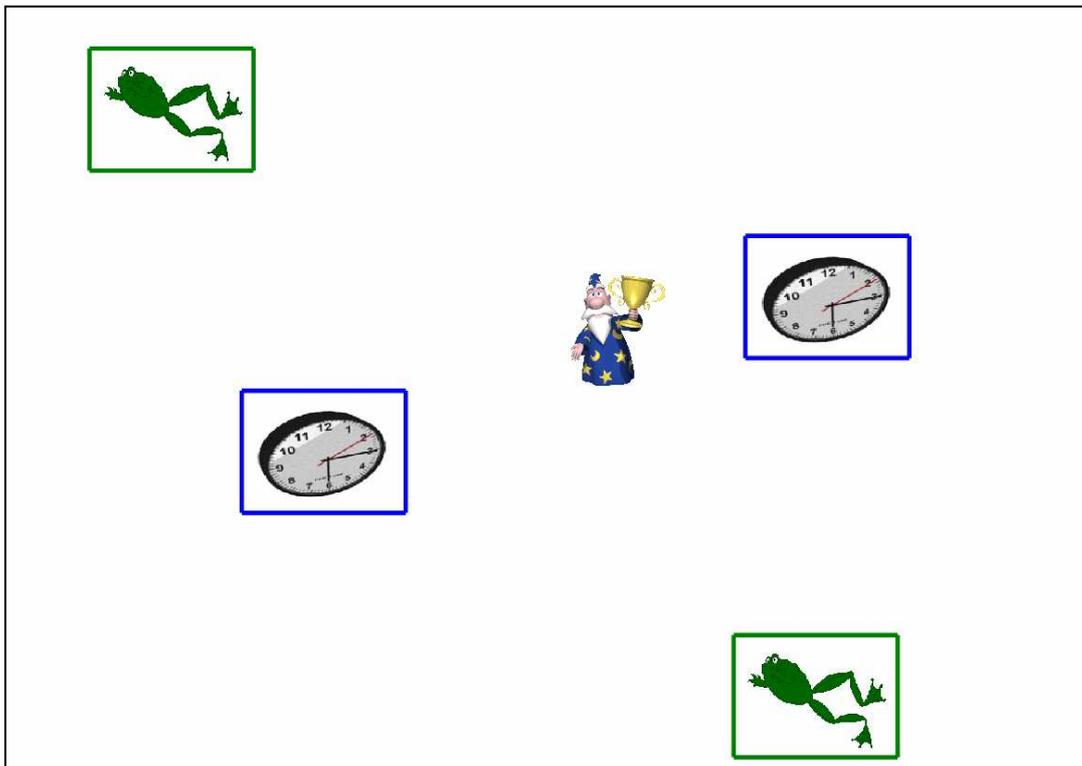
Felicitation **Muy bien** * ?

Correccion **Oh Oh** * ?

Siguiente Actividad:

Ruta **Ninguna** * ?

Enviar consulta



3.3.5 ORDENAR AUTO

Con esta plantilla se pueden ordenar colecciones de imágenes. El objetivo de esta actividad es dado una serie de elementos que se colocan en la parte superior de la pantalla, colocarlos en el orden que les corresponda en el rectángulo de la parte inferior (por ejemplo, de menor a mayor, de mayor a menor, etc.). Por defecto, aparece una primera imagen que consta de cinco divisiones que nos permitirán ordenar cinco objetos, teniendo ya calculado de antemano en qué área hay que situar esos objetos, para que el resultado sea el esperado (para tamaños superiores e inferiores se pueden usar las plantillas Arrastrar Auto o Arrastrar).

El campo Orden indica qué posición en el rectángulo tiene que ocupar el objeto. El fondo tiene que tener el valor 0. Igual que en el caso de la plantilla Arrastrar Auto el campo Fijo tiene que estar activado para los objetos que no se puedan mover por la pantalla. Para todos los que se puedan mover (sean, o no, solución al ejercicio) hay dejar esta casilla sin marcar. Además, para los que no sean solución, hay que activar también el campo Fondo.

Plantillas de Orden

Imágenes:

Select	Id	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ancho	Alto	Ruta	Orden	Fijo	Fondo
<input type="checkbox"/>	division3 *	2 *	50 *	97 *	50 *	c:\division3.gif *	0 *	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Anadir Imagen Borrar imagen seleccionada

Vamos a ver un ejemplo en el que queremos ordenar unos barriles de menor a mayor. Para ello primero ponemos dos barriles fijos (en este caso hemos puesto el primero y el cuarto) en el recuadro de la solución. Así el alumno puede deducir que se quieren ordenar de menor a mayor.

Plantillas de Orden

Imágenes:

Select	Id	Posicion Inicial Eje X	Posicion Inicial Eje Y	Ancho	Alto	Ruta	Orden	Fijo	Fondo
<input type="checkbox"/>	division3	2	50	97	50	division3.gif	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	barri1	10	77	15	8	barri.gif	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	barri2	45	15	20	10	barri.gif	2	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	barri3	62	15	25	12	barri.gif	3	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	barri4	62	63	30	14	barri.gif	4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	barri5	22	15	35	16	barri.gif	5	<input type="checkbox"/>	<input type="checkbox"/>

Agente:

Accion * ?

Select	Id	Posicion Eje X	Posicion Eje Y	Accion	Texto Explicacion
<input type="checkbox"/>	accion1	12	12	Saludar	Ordena los barriles

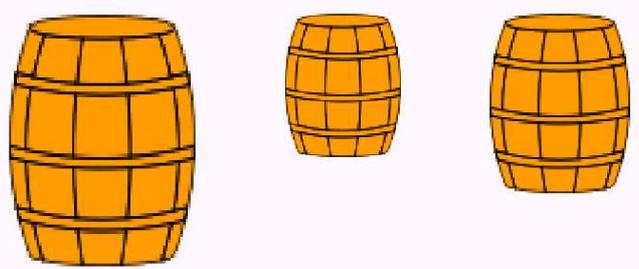
Felicitacion * ?

Correccion * ?

Siguiente Actividad:

Ruta * ?








APÉNDICE D. XML SCHEMA

En este Apéndice se presenta el Código del XML Schema que se ha diseñado para las plantillas del Tutorial.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema
elementFormDefault="qualified" attributeFormDefault="unqualified">
<xs:element name="IMAGEN">
  <xs:annotation>
    <xs:documentation>Comment describing your root
element</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="imagen" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Descripcion" type="xs:string"/>
            <xs:element name="Color" type="xs:string"/>
            <xs:element name="Posicion">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Inicial">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="Ix" type="xs:string"/>
                        <xs:element name="Iy" type="xs:string"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="Final">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="Ax1" type="xs:int"/>
                        <xs:element name="Ax2" type="xs:int"/>
                        <xs:element name="Ay1" type="xs:int"/>
                        <xs:element name="Ay2" type="xs:int"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="Tamano">
              <xs:complexType>
```

```

        <xs:sequence>
            <xs:element name="Alto" type="xs:string"/>
            <xs:element name="Ancho" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Ruta" type="xs:string"/>
<xs:element name="Fijo" type="xs:boolean"/>
<xs:element name="Fondo" type="xs:boolean"/>
<xs:element name="Pareja" type="xs:string"/>
<xs:element name="Orden" type="xs:int"/>
<xs:element name="Cantidad"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="agente">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Nombre" type="xs:string"/>
            <xs:element name="Explicacion" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Id" type="xs:int"/>
                        <xs:element name="MoveTo">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="PosicionX" type="xs:string"/>
                                    <xs:element name="PosicionY" type="xs:string"/>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="Play" type="xs:string"/>
                        <xs:element name="Speak" type="xs:string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="Felicitacion" type="xs:string"/>
            <xs:element name="Correccion" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="web">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Siguiete" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

APÉNDICE E. EJEMPLO DE XSLT

En este Apéndice se presenta el Código de un XSLT para una actividad de Tipo Pinchar.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance">
<xsl:template match="/">
<html>
  <head>
    <title>Mis imagenes</title>
    <!-- Objeto identificador del agente -->
    <OBJECT classid="clsid:D45FD31B-5C6E-11D1-9EC1-00C04FD7081F"
height="0" id="AgentCtl" width="0"/>

<script language="JavaScript">
<xsl:text>
<!--Definición de variables -->
var new_x = 0;
var new_y = 0;
var inicio = 0;
var final = 0;
var tiempo = 0;
espera = true;
imgArray = new Array(0);
datosArray = new Array(0);
resultadoArray = new Array(0);
datosAgente = new Array(0);
feedbacksAgente = new Array(0);
var nombreAgente = " ";
var webSiguiente = " ";
var Agent;
var cont = 0;
var total = 0;
var numImagenes = 0;
</xsl:text>

<!-- Se almacenan los datos de cada imagen que proviene del xml en un array de
imágenes -->
<xsl:for-each select="//imagen">
imgArray['<xsl:value-of select="Descripcion"/>'] = false;
datosArray['<xsl:value-of select="Descripcion"/>'] = new Array(0);
datosArray['<xsl:value-of select="Descripcion"/>']['Id'] = "<xsl:value-of
select="Descripcion"/>";
```

```

datosArray['<xsl:value-of select="Descripcion"/>']['Ruta'] = "<xsl:value-of
select="Ruta"/>";
datosArray['<xsl:value-of select="Descripcion"/>']['Fijo'] = <xsl:value-of
select="Fijo"/>;
datosArray['<xsl:value-of select="Descripcion"/>']['Ix'] = <xsl:value-of
select="Posicion//Inicial//Ix"/>;
datosArray['<xsl:value-of select="Descripcion"/>']['Iy'] = <xsl:value-of
select="Posicion//Inicial//Iy"/>;
datosArray['<xsl:value-of select="Descripcion"/>']['Alto'] = <xsl:value-of
select="Tamano//Alto"/>;
datosArray['<xsl:value-of select="Descripcion"/>']['Ancho'] = <xsl:value-of
select="Tamano//Ancho"/>;
resultadoArray['<xsl:value-of select="Descripcion"/>'] = new Array(0);
</xsl:for-each>

<!-- Se almacenan los datos del agente en un array -->
nombreAgente = "<xsl:value-of select="//agente//Nombre"/>";

<xsl:for-each select="//agente//Explicacion">
datosAgente['<xsl:value-of select="Id"/>'] = new Array(0);
datosAgente['<xsl:value-of select="Id"/>']['PosicionX'] = <xsl:value-of
select="MoveTo//PosicionX"/>;
datosAgente['<xsl:value-of select="Id"/>']['PosicionY'] = <xsl:value-of
select="MoveTo//PosicionY"/>;
datosAgente['<xsl:value-of select="Id"/>']['Play'] = "<xsl:value-of
select="Play"/>";
datosAgente['<xsl:value-of select="Id"/>']['Speak'] = "<xsl:value-of
select="Speak"/>";
</xsl:for-each>

feedbacksAgente['Felicitacion'] = "<xsl:value-of select="//agente/Felicitacion"/>";
feedbacksAgente['Correccion'] = "<xsl:value-of select="//agente/Correccion"/>";

<!-- Se almacena la siguiente actividad con la que se encadena ésta en la variable
webSiguiente -->
webSiguiente = "<xsl:value-of select="//web//Siguiente"/>";

<xsl:text>
<!-- Función para calcular la Fecha actual -->
function time() {
    inicio = new Date();
}

<!-- Función para calcular el tiempo que se ha tardado en realizar la actividad -->
function time_out() {
    final = new Date();
    tiempo = (final.getTime()-inicio.getTime())/1000;
    tiempo = Math.round(tiempo);
}

function set_drag(name)

```

```
{
  imgArray[name] = !imgArray[name];
}
```

<!-- Posición donde se está pinchando con el ratón -->

```
function handle_move()
{
  for (var i in imgArray) {
    if ( eval('imgArray.' + i) == true ) {
      if (datosArray[i]['Fijo'] == false) {
        window.new_x = window.event.x;
        window.new_y = window.event.y;
        setTimeout("drag_it('" + i + "');",1);
        break;
      }
    }
  }
}
```

<!-- Posición donde se está pinchando con el ratón -->

```
function drag_it(name) {
  with ( window.document.all[name].style ) {
    pixelLeft = window.new_x - pixelWidth/2;
    pixelTop = window.new_y - pixelHeight/2;
    resultadoArray[name]['pixelLeft'] = pixelLeft;
    resultadoArray[name]['pixelTop'] = pixelTop;
  }
}
```

```
function Mostrar() {
  var correcto = false;
  var izquierda, derecha, arriba, abajo;
```

```
  time_out();
  total++;
  espera = false;
  continuar = false;
```

<!-- alert('Tiempo empleado : ' + tiempo); -->

```
  for (var i in imgArray) {
    <!-- Si la imagen es una posible solución del ejercicio -->
    if (datosArray[i]['Fijo'] == false) {
      correcto = false;
      <!-- Se calculan las dimensiones -->
      resultadoArray[i]['pixelLeft'] = window.event.x;
      resultadoArray[i]['pixelTop'] = window.event.y;
      winWid = (NS4) ? innerWidth : document.body.clientWidth;
      winHgt = (NS4) ? innerHeight : document.body.clientHeight;
      izquierda = winWid * datosArray[i]['Ix'] / 100;
      arriba = winHgt * datosArray[i]['Iy'] / 100;
      derecha = izquierda + winWid * datosArray[i]['Ancho'] / 100;
```

```

        abajo = arriba + winHgt * datosArray[i]['Alto'] / 100;
        <!-- Se comprueba si se está pinchando en la imagen correcta
-->
        if (resultadoArray[i]['pixelLeft'] >= izquierda) {
            if (derecha >= resultadoArray[i]['pixelLeft']) {
                if (resultadoArray[i]['pixelTop'] >= arriba) {
                    if (abajo >= resultadoArray[i]['pixelTop'])
                    {
                        correcto = true;
                        cont++;
                    }
                }
            }
        }
    }
    <!-- Si se ha solucionado el ejercicio se felicita al niño -->
    if (correcto) {
        felicitar();
    }
    <!-- Si no se le corrige y se le da otra oportunidad -->
    else {
        corregir();
    }
}
</xsl:text>
</script>

<script language="JavaScript">
<xsl:text>

<!-- Función que actualiza la BD -->
function escribirBD() {
    <!-- Conexión con la BD -->
    var adoConn = new ActiveXObject("ADODB.Connection");
    adoConn.Open("PROVIDER=Microsoft.Jet.OLEDB.4.0; DATA
SOURCE=C:\\Tutorial\\wrapper1\\bd1.mdb");
    <!-- Se inserta el resultado obtenido por el alumno en la BD -->
    var sql = "INSERT INTO historial2 (correcto, total) values (" +
numImagenes + ", " + total + ")";
    adoConn.Execute(sql);
    <!-- Le indica al Gestor que ha finalizado la actividad -->
    var sql = "UPDATE control SET listo = 1";
    adoConn.Execute(sql);

}

</xsl:text>
</script>

<script LANGUAGE="JavaScript">
<xsl:text>

```

```
<!-- Identificación de la página web en Netscape y Explorer -->
NS = (document.layers);
IE = (document.all);

<!-- Se activan los escalados para poder redimensionar la pantalla-->
scaleWidth = true;
scaleHeight = true;
scaleLeft = true;
scaleTop = true;

<!-- Recalcular la pantalla si se está usando Netscape -->
if (NS) window.onload = setResize;

function setResize(){
    setTimeout("window.onresize=reDo;",500);
}

function reDo(){
    window.location.reload()
}

<!-- Recalcular la pantalla si se está usando Explorer -->
if (IE) window.onresize = reDoIE;

<!-- Función para recalcular los valores de las imágenes si cambia el tamaño de la
pantalla -->
function reDoIE(){

    for (var name in imgArray) {

        <!-- Obtención de las nuevas dimensiones de la pantalla -->
        winWid = (NS4) ? innerWidth : document.body.clientWidth;
        winHgt = (NS4) ? innerHeight : document.body.clientHeight;

        <!-- Actualizando coordenadas de las imágenes -->
        width = winWid * dataArray[name]['Ancho'] / 100;
        height = winHgt * dataArray[name]['Alto'] / 100;

        left = winWid * dataArray[name]['Ix'] / 100;
        vtop = winHgt * dataArray[name]['Iy'] / 100;

        eval(name + '.style.width = ' + width);
        eval(name + '.style.height= ' + height);
        eval(name + '.style.left = ' + left);
        eval(name + '.style.top= ' + vtop);
    }

}

function makeIm() {
    var cont = 0;
```

```

<!-- Obtención de las nuevas dimensiones de la pantalla -->
winWid = (NS4) ? innerWidth : document.body.clientWidth;
winHgt = (NS4) ? innerHeight : document.body.clientHeight;

<!-- Calculando las coordenadas de cada imagen en la pantalla-->
for (var name in imgArray) {

    left = winWid * datosArray[name]['Ix'] / 100;
    vtop = winHgt * datosArray[name]['Iy'] / 100;
    width = winWid * datosArray[name]['Ancho'] / 100;
    height = winHgt * datosArray[name]['Alto'] / 100;

    <!-- Dibujando las imágenes -->
    imStr = "&lt;IMG NAME= \" + datosArray[name]['Id'] + "\" BORDER=0
SRC=" + datosArray[name]['Ruta'] + " galleryimg = no onclick = \"Mostrar();\"
style = \"position:absolute;";
    if (scaleWidth) imStr += " WIDTH:" + width;
    if (scaleHeight) imStr += "; HEIGHT:" + height;
    if (scaleLeft) imStr += "; LEFT:" + left;
    if (scaleTop) imStr += "; TOP:" + vtop;

    imStr += ";\"&gt;";

    document.write(imStr);

    if (datosArray[name]['Fijo'] == false) {
        numImágenes++;
    }
}
}

</xsl:text>
</script>

<SCRIPT language="JavaScript">
<xsl:text>

<!-- Script para el agente -->

var espera = true;
var MyRequest;

function OnLoad() {
AgentCtl.Connected = true;

<!-- Cargando el agente -->
LoadRequest = AgentCtl.Characters.Load ("char1", nombreAgente);
<!-- Si no se encuentra mostrando error -->
<!--If (LoadRequest.Status == 1) {
    MsgBox ("You don't have a Agent stored in the \chars directory."+chr(13));
}

```

```
-->
<!-- Instanciando un objeto agente-->
Agent = AgentCtl.Characters.Character ("char1");

<!-- Indicando lenguaje español (para el TextToSpeech (TTS)) -->
Agent.LanguageID = 0x0C0A;

<!-- Inicializando voz del TextToSpeech -->
<!--InitializeCommands(); -->

<!-- Operando con el agente -->
DoIntro();
}

<!-- Función para mover el agente por la pantalla y dar las explicaciones de la
actividad -->
function DoIntro() {

<!-- Calcular dimensiones de la pantalla -->
winWid = (NS4) ? innerWidth : document.body.clientWidth;
winHgt = (NS4) ? innerHeight : document.body.clientHeight;

<!-- Hacer visible el agente -->
Agent.Show();

<!-- Para cada interacción del agente moverlo, realizar gesto y decir texto -->
for (var i in datosAgente) {

pleft = winWid * datosAgente[i]['PosicionX'] / 100;
ptop = winHgt * datosAgente[i]['PosicionY'] / 100;

<!-- Mover agente -->
Agent.MoveTo(pleft, ptop);
<!-- Realizar acción (gesto) -->
Agent.Play(datosAgente[i]['Play']);
<!-- Decir texto sin TTS (con TTS: Agent.Speak(datosAgente[i]['Speak']); -->
Agent.Speak (" ", datosAgente[i]['Speak']);

}

<!-- Interacción de pensar y capturar tiempo -->
<!--Agent.Play ("Think");
-->time();

}

<!-- Felicitar con el agente -->
function felicitar() {
    Agent.Play ("Congratulate");
    MyRequest = Agent.Speak(" ", feedbacksAgente['Felicitacion']);
}
}
```

```
<!-- Corregir con el agente -->
function corregir() {
    Agent.Play ("Decline");
    Agent.Speak (" ", feedbacksAgente['Correccion']);
<!-- Agent.Play ("Think");
-->}

</xsl:text>
</SCRIPT>

<SCRIPT language="JavaScript" FOR="AgentCtl"
EVENT="RequestComplete(Request)">
<xsl:text>

    <!-- Si el agente felicita y se ha resuelto el ejercicio -->
    if (Request == MyRequest) {
        if (cont == numImágenes) {
            <!-- Almacenar el resultado en la BD -->
            escribirBD();
        }
        <!-- Si hay enlazada otra actividad mostrarla-->
        if (webSiguiente != "Ninguna") {
            window.location = webSiguiente;
        }
    }

</xsl:text>
</SCRIPT>

</head>
<body onload="OnLoad()" Marginheight="0" Marginwidth="0">
    <script LANGUAGE="JavaScript">
        <xsl:text>
            <!-- Cargar y mostrar las imágenes -->
            makeIm();
        </xsl:text>
    </script>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

APÉNDICE F. PERFORMATIVAS KQML

En este apéndice se presenta el conjunto de performativas del lenguaje KQML.

- Achieve: El emisor pide al receptor que ejecute una acción.
- Advertise: El emisor quiere que el receptor sepa que se va a procesar un mensaje como el que aparece en el contenido.
- Ask-all: El emisor quiere conocer todas las instancias del contenido del mensaje en la base de conocimiento del receptor.
- Ask-if: El emisor quiere saber si el contenido del mensaje es cierto (está en la base de conocimientos del receptor).
- Ask-one: El emisor quiere conocer una de las instancias del contenido del mensaje en la base de conocimiento del receptor.
- Broadcast: El emisor pide al receptor que reenvíe el mensaje a todos los agentes.
- Broker-one: El emisor pide al receptor que encuentre una respuesta a una performativa.
- Broker-all: El emisor pide al receptor que encuentre todas las respuestas a una performativa.
- Delete-all: El emisor pide al receptor que elimine todas las instancias de su base de conocimientos.
- Delete-one: El emisor pide al receptor que elimine una instancia de su base de conocimientos.
- Deny: Negación del contenido.
- Discard: El emisor no quiere que el receptor envíe todas las respuestas restantes a un mensaje recibido previamente.
- Eos: Final de una respuesta múltiple.
- Error: El emisor considera que el mensaje enviado por el receptor no cumple la sintaxis KQML.
- Forward: El emisor pide al receptor que reenvíe el mensaje al agente que aparece en el :to del mensaje.
- Insert: El emisor solicita al receptor que añada el contenido del mensaje a su base de conocimientos.

- Next: El emisor pide al receptor la siguiente respuesta a un mensaje recibido previamente.
- Ready: El emisor indica que está listo para responder a un mensaje.
- Recommend-all: El emisor solicita todos los agentes que puedan responder a una preformativa.
- Recommend-one: El emisor solicita un agente que pueda responder a una preformativa.
- Recruit-all: El emisor solicita al receptor que busque a todos los agentes adecuados para responder a una preformativa.
- Recruit-one: El emisor solicita al receptor que busque un agente adecuado para responder a una preformativa.
- Register: El emisor informa de su presencia y de su nombre simbólico.
- Rest: El emisor pide al receptor todas las respuestas restantes a un mensaje recibido previamente.
- Sorry: El emisor entiende el mensaje enviado por el receptor, pero es incapaz de hacer nada más con él.
- Standby: El emisor pregunta al receptor si está preparado para responder a un mensaje.
- Stream-all: Como ask-all pero con respuesta múltiple.
- Subscribe: El emisor quiere que el receptor le envíe un mensaje cada vez que se actualiza el resultado de la acción indicada en la performativa.
- Tell: Indica que el contenido está en la base de conocimientos del emisor (es cierto para el emisor).
- Transport-address: El emisor asocia su nombre simbólico con una nueva dirección.
- Unachieve: El emisor pide al receptor la anulación de un achieve previo.
- Unadvertise: Anulación de un advertise previo.
- Undelete: El emisor pide al receptor que anule un delete previo.
- Uninsert: Se pide la anulación de un Insert previo.
- Unregister: Anulación de una acto previo de register.
- Untell: Indica que el contenido no está en la base de conocimientos del emisor (es cierto para el emisor).

APÉNDICE G. PERFORMATIVAS FIPA-ACL

En este apéndice se presenta el conjunto de performativas del lenguaje FIPA-ACL.

- Accept-Proposal: Acto de aceptar una proposición remitida anteriormente mediante un Propose.
- Agree: Acto de estar de acuerdo en una acción.
- Cancel: Acto mediante el cual un agente le indica a otro que cancele la ejecución de la acción que le envía indicado anteriormente.
- Cfp: Acto de pedir propuesto para la ejecución de una acción.
- Confirm: Acto en el que el emisor confirma al receptor de la veracidad de una proposición.
- Disconfirm: Acto en el que el emisor informa al receptor de que una proposición es falsa.
- Failure: Acto de indicar a otro agente el fallo en la ejecución de una acción.
- Inform: Acto de informar a un agente que una proposición es verdadera.
- Inform If: Acto para que el agente encargado de realizar una acción informe al receptor sobre si una proposición es cierta o no.
- Inform Ref: Acto para que el agente emisor informe al receptor sobre un objeto que corresponde a un cierto descriptor.
- Not Understood: Acto en el que el emisor informa al agente receptor de que sabe que dicho agente ha llevado a cabo una acción pero no la entiende.
- Propagate: Acto de enviar un mensaje a un agente para que este lo propague a otros que figuran en un descriptor de dicho mensaje.
- Propose: Acto de enviar una propuesta para llevar a cabo una acción.
- Proxy: Acto de enviar un mensaje al receptor para que este seleccione unos agentes de destino y les envíe un mensaje.
- Query If: Acto de preguntar a un agente si una proposición es verdadera o no.
- Query Ref: Acto de preguntar a un agente por el objeto referenciado por una expresión.
- Refuse: Acto de rechazar una acción.
- Rejet Proposal: Acto de rechazar una propuesta de una negociación.

- Request: Acto de enviar un mensaje al receptor para que ejecute una acción.
- Request When: Igual que el anterior pero la acción se ejecutará cuando se cumplan unas condiciones.
- Request Whenever: Igual que el anterior pero la acción se repetirá cada vez que se cumplan las condiciones.
- Subscribe: Acto de pedir la notificación del valor de una referencia y de que si dicho objeto sufre alguna variación se notifique de nuevo.

APÉNDICE H. PROTOCOLOS DE INTERACCIÓN FIPA

En este anexo se describen brevemente los protocolos de interacción en el estándar FIPA, los cuales determinan las conversaciones que se pueden establecer entre los diversos agentes.

1 PROTOCOLO DE INTERACCIÓN FIPA PROPOSE

El iniciador (agente que comienza el protocolo) manda un mensaje *propose* al participante (agente receptor) indicando que ejecutará una acción si el participante acepta. El participante responde aceptando o rechazando la propuesta, a través de los actos comunicativos *accept-proposal* o *reject-proposal* respectivamente.

2 PROTOCOLO DE INTERACCIÓN FIPA REQUEST

Este protocolo permite a un agente pedir a otro agente que lleve a cabo una acción. El iniciador ejecuta la petición mediante el acto comunicativo *request*. El participante procesa el mensaje correspondiente y decide si aceptar o rechazar la petición. La respuesta tiene lugar mediante los actos comunicativos *agree* (opcional) y *refuse* respectivamente. Una vez que el participante ha aceptado el *request* inicial, puede mandar los siguientes actos de comunicación al iniciador:

- *failure* si el intento de llevar a cabo la acción ha fracasado.
- *inform-done* si ha conseguido llevar a cabo la acción y solamente quiere informar que se ha completado.
- *inform-result* si ha completado la tarea y además quiere comunicar los resultados al iniciador.

3 PROTOCOLO DE INTERACCIÓN FIPA REQUEST WHEN

Este protocolo permite a un agente pedir a otro agente que lleve a cabo una acción, cuando se cumplan unas ciertas condiciones previas. Es una variación del protocolo de interacción FIPA Request. El iniciador ejecuta la petición mediante el acto comunicativo *request when*. El participante procesa el mensaje correspondiente y decide si aceptar o rechazar la petición. La respuesta tiene lugar mediante los mismos actos comunicativos que en el apartado anterior: *failure*, *inform-done* o *inform-result*.

4 PROTOCOLO DE INTERACCIÓN FIPA QUERY

Este protocolo permite a un agente llevar a cabo algún tipo de acción sobre otro agente. El iniciador pide al participante que lleve a cabo alguna clase de acción *inform* usando para ello uno de estos actos de comunicación: *query-if* o *query-ref*. El acto *query-if* se emplea cuando el iniciador desea saber si una determinada proposición es cierta o no. Por otro lado, el *query-ref* se emplea cuando el iniciador desea consultar por algunos objetos. En ambos casos, el participante procesa el acto de comunicación y decide si la acepta (*agree* que puede ser opcional) o la rechaza (*refuse*) con lo que termina la interacción. Si el participante es incapaz de realizar la consulta, devuelve al iniciador un acto de comunicación *failure*. En caso de que la consulta se haya realizado satisfactoriamente, existen dos posibilidades:

- En el caso de un *query-if*, el participante devuelve un *inform-t/f*, informando sobre la veracidad (*truth*) o falsedad (*falsehood*) de la proposición consultada.
- En el caso de un *query-ref*, el participante devuelve un *inform-result*, informando en su campo *content* de la expresión de los objetos cuya consulta fue especificada.

5 PROTOCOLO DE INTERACCIÓN FIPA CONTRACT NET

En este protocolo de interacción, un agente (iniciador) toma el papel de gestor que desea que otros agentes (los participantes) lleven a cabo una tarea y además desea optimizar una cierta función que caracteriza la tarea en cuestión (el precio asociado a la tarea). Para una tarea dada, cualquier número de participantes podrían responder con una propuesta, rechazando el resto la negociación. La negociación continúa con aquellos agentes que han realizado una propuesta. El iniciador solicita m propuestas de otros tantos agentes mediante un acto de comunicación *cfp* (call for proposals), donde se especifica la tarea así como cualquier condición impuesta por el iniciador para llevar a cabo la negociación. Los participantes son vistos como potenciales contratistas y que son capaces de generar n respuestas. De éstas, j son propuestas para llevar a cabo la tarea a través de un acto de comunicación *propose*. La propuesta del participante incluye las precondiciones de éste para llevar a cabo la tarea, como por ejemplo el precio, el tiempo en que estará lista, etc. De modo alternativo, los $i=n-j$ participantes restantes deberían ejecutar un acto comunicativo *refuse*. Una vez que expire el plazo dado para obtener respuesta, el iniciador evalúa las ofertas recibidas y selecciona el agente o agentes a los que encargar la realización de la tarea. A los l agentes que son seleccionados se les envía un *accept-proposal*, mientras que a los $k=j-l$ se les envía un *reject-proposal*. Las propuestas son de obligado cumplimiento para el participante, por lo que una vez que el iniciador acepta la propuesta, el participante adquiere un compromiso para ejecutar la tarea. Una vez que el participante ha llevado a cabo la tarea, envía un mensaje al iniciador en la forma de un *inform-done* o en la versión más explicativa *inform-result*. Por el contrario, si no ha sido capaz de llevar a cabo la tarea, el participante envía un acto comunicativo *failure* al iniciador. Este protocolo necesita que el iniciador sepa cuándo ha recibido todas las respuestas. En caso de que un participante no pueda responder al llamamiento inicial, el iniciador potencialmente podría quedar bloqueado esperando indefinidamente por las respuestas. Para evitarlo el *cfp* inicial incluye un plazo de

tiempo para que los participantes puedan responder. Las propuestas recibidas después de este plazo son automáticamente rechazadas.

6 PROTOCOLO DE INTERACCIÓN FIPA ITERATED CONTRACT NET

Este protocolo es una extensión del protocolo de iteración FIPA Contract Net, que permite un ciclo de ofertas en varias rondas. Como en el caso del protocolo de interacción *FIPA Contract Net*, el iniciador lanza m *cfp* iniciales. De los n participantes que responden, k son mensajes *propose* provenientes de los participantes que quieren y son capaces de llevar a cabo la tarea en cuestión bajo las condiciones especi_cadas, mientras que las restantes j respuestas consisten en *refuse*. De estas k propuestas recibidas, el iniciador podría decidir que se encuentra en la iteración final del proceso y aceptar p de las propuestas recibidas, rechazando el resto. De modo alternativo, el iniciador podría decidir iterar el proceso, lanzando un *cfp* revisado a l de los participantes y rechazando mediante un *reject-proposal* a los restantes. El motivo consiste en que el iniciador busca obtener mejores ofertas modificando el *cfp* inicial. El proceso termina cuando el iniciador rechaza todas las ofertas y no manda un nuevo *cfp*, el iniciador acepta una o más ofertas o todos los participantes rechazan la negociación.

7 PROTOCOLO DE INTERACCIÓN FIPA ENGLISH AUCTION

Este protocolo trata de implementar un sistema de subasta a la inglesa. El iniciador trata de encontrar el precio de mercado de una mercancía proponiendo un precio inferior al que se supone de su valor de mercado, para después ir subiendo progresivamente ese precio. Cada vez que se anuncia el nuevo precio, el iniciador

espera a ver si alguno de los posibles compradores (participantes) está dispuesto a pagar ese nuevo precio. Tan pronto como algún comprador anuncia esta disposición, el iniciador lanza una nueva oferta con un precio incrementado. Este proceso de subasta termina cuando ningún comprador acepta el nuevo precio. Si el último precio aceptado por un comprador supera al precio esperado por el iniciador (este precio es desconocido para los participantes), la mercancía es vendida al participante correspondiente. En caso contrario, la mercancía no es vendida. Inicialmente, el iniciador lanza su oferta inicial a todos los participantes mediante un *cfp*. Obsérvese que cuando se recibe una respuesta de uno de los participantes, el resto de participantes no tienen por qué enterarse de esa respuesta. Lo que si debe ocurrir es que cada participante que realiza una oferta tiene que conocer si su oferta ha sido aceptada o no, mediante la recepción respectivamente de los actos de comunicación *accept-proposal* y *reject-proposal*. Para finalizar el protocolo, el subastador introduce un protocolo *FIPA Request* con el postor ganador con el objetivo de completar la transacción objeto de la subasta.

8 PROTOCOLO DE INTERACCIÓN FIPA DUTCH AUCTION

Este protocolo implementa un sistema de subasta holandesa. En contraste con el protocolo de *FIPA English Auction*, el iniciador trata de encontrar el precio de mercado de una mercancía proponiendo un precio muy superior al que se supone su valor de mercado, para después ir reduciendo progresivamente ese precio, hasta que uno de los participantes en la subasta acepta el precio. Esa reducción progresiva depende del iniciador, el cual suele tener un precio mínimo de venta, oculto para los participantes en la subasta.

9 PROTOCOLO DE INTERACCIÓN FIPA BROKERING

Este protocolo está diseñado para soportar las interacciones de intermediación (*brokerage*) en la mediación en sistemas multiagentes. En general, un *broker* es un agente que ofrece un conjunto de servicios de comunicación a otros agentes empleando para ello algún tipo de conocimiento acerca de las peticiones y capacidades de esos agentes. Un ejemplo típico de intermediación es aquél en que un agente pide a un *broker* encontrar a uno o más agentes que le puedan responder a un acto de comunicación *query*. El broker determina un conjunto de agentes adecuados para la consulta, establece un protocolo de interacción *Query* y transmite las respuestas recibidas al iniciador. Además, los *brokers* también permiten a un sistema ser robusto y adaptarse ante situaciones dinámicas, incluyendo los aspectos de seguridad en estos agentes. El protocolo comienza cuando el iniciador manda un mensaje referente a un acto de comunicación *proxy* al *broker*. El iniciador embebe dentro del *proxy* otro mensaje con el acto comunicativo deseado. El agente *broker* procesa la petición y decide si la acepta (*agree*) o la rechaza (*refuse*) en cuyo caso termina la interacción. En caso de aceptarla, el reclutador intenta localizar los agentes cuya descripción concuerde con la expresión referencial del *proxy* inicial. Si no es posible encontrar a ningún agente, el reclutador enviará un *failure-no-match* al iniciador, lo que supone el final de la interacción. En caso contrario, el reclutador podría modificar la lista de los agentes encontrados basándose en el parámetro de condición del *proxy*. Entonces comienza m interacciones con la lista resultante de n agentes, con cada interacción en sub-protocolo por separado. En este punto, el *broker* debería recordar algunos de los parámetros del *proxy* inicial, por ejemplo el *conversation-id*, *reply-with* y *sender*, con el fin de enviar al iniciador las r respuestas obtenidas en el proceso. La naturaleza del sub-protocolo y de las respuestas dependen de los protocolos de interacción especificados en el mensaje embebido en el *proxy* original. A medida que el sub-protocolo tiene lugar, el *broker* reenvía los mensajes recibidos de los participantes al iniciador.

10 PROTOCOLO DE INTERACCIÓN FIPA RECRUITING

Este protocolo está diseñado para soportar las interacciones de reclutamiento (*recruiting*) en la mediación en sistemas multiagentes. Un agente *reclutador* es una especie de *broker*, pero con la diferencia que los agentes seleccionados por el reclutador se dirigen directamente al agente iniciador o a algún receptor previamente designado. Del mismo modo que con el *broker*, el empleo de agentes reclutadores puede simplificar de modo significativo la interacción de los agentes en un MAS, así como dotar al sistema de robustez y adaptabilidad ante situaciones dinámicas. El protocolo comienza cuando el iniciador manda un mensaje referente a un acto de comunicación *proxy* al reclutador. Este mensaje contiene una expresión referencial indicando los agentes a los que el reclutador debiera enviar el mensaje embebido, el acto comunicativo a enviar y una serie de parámetros como por ejemplo el número máximo de agentes a los que reenviar el mensaje. El reclutador procesa la petición y decide si la acepta (*agree*) o la rechaza (*refuse*, en cuyo caso termina la interacción). En caso de aceptarla, el reclutador intenta localizar los agentes cuya descripción concuerde con la expresión referencial del *proxy* inicial. Si no es posible encontrar a ningún agente, el reclutador enviará un *failure-no-match* al iniciador, lo que supone el fin de la interacción. En caso contrario, el reclutador podría modi_car la lista de los agentes encontrados basándose en el parámetro de condición del *proxy*. Entonces comienzan m interacciones con la lista resultante de n agentes, con cada interacción en un sub-protocolo por separado. El inicio del protocolo debe realizarse con cuidado, usando los parámetros para manejar las respuestas de la petición. Así, si el mensaje que ha llegado al reclutador tiene un parámetro *designated-receiver*, necesita empezar cada sub-protocolo con un parámetro *reply-to* conteniendo al receptor designado, así como el identi_cador de conversación *conversation-id* de la conversación original. Es posible que sea necesario propagar otros parámetros. La naturaleza del sub-protocolo y de las respuestas dependen de los protocolos de interacción especificados en el mensaje embebido en el *proxy* original.

11 PROTOCOLO DE INTERACCIÓN FIPA SUBSCRIBE

Este protocolo permite a un agente pedir a otro agente llevar a cabo una acción de suscripción (*subscribe*). El iniciador comienza la interacción con acto de comunicación *subscribe* conteniendo la referencia de los objetos en los que está interesado. El participante procesa la petición y decide si la acepta (agree opcional según las circunstancias) o la rechaza (*refuse*), en cuyo caso termina la interacción. Tras aceptarla, el participante responde con un *inform-result* cuyo campo *content* corresponde a una expresión referencial de los objetos de la suscripción. El participante continúa enviando un *inform-result* cada vez que se produce algún cambio en dichos objetos. En caso de que en algún momento se produjese algún problema, se envía un *failure* al iniciador, finalizando el protocolo.

APÉNDICE I. SIMULACIONES PARA EL OBJETIVO 4.

Iter.	Actividad Actual	Agente Actividad 1	Agente Actividad 2	Agente Actividad 3	Agente Actividad 4
50	Act4 (GCO = 0.5)	Fact = 20.27	Fact = 20.27	Fact = 20.27	Fact = 5.16
51	Act1 (GCO = 0.5)	Fact = 9.98	Fact = 20.39	Fact = 20.39	Fact = 5.18
52	Act2 (GCO = 0.5)	Fact = 10.04	Fact = 10.04	Fact = 21.05	Fact = 5.19
53	Act3 (GCO = 0.5)	Fact = 10.10	Fact = 10.10	Fact = 10.10	Fact = 5.21
54	Act1 (GCO = 0.5)	Fact = 4.98	Fact = 10.16	Fact = 10.16	Fact = 5.22
55	Act2 (GCO = 0.5)	Fact = 5	Fact = 5	Fact = 10.21	Fact = 5.23
56	Act3 (GCO = 0.5)	Fact = 5.03	Fact = 5.03	Fact = 5.03	Fact = 5.25
57	Act4 (GCO = 0.5)	Fact = 5	Fact = 5	Fact = 5	Fact = 2.58
58	Act1 (GCO = 0.5)	Fact = 3.33	Fact = 5	Fact = 5	Fact = 2.59
59	Act2 (GCO = 0.5)	Fact = 3.34	Fact = 3.34	Fact = 5.11	Fact = 2.60
60	Act3 (GCO = 0.5)	Fact = 3.36	Fact = 3.36	Fact = 3.36	Fact = 2.60
61	Act1 (GCO = 0.5)	Fact = 2.44	Fact = 3.38	Fact = 3.38	Fact = 2.61
62	Act2 (GCO = 0.5)	Fact = 2.5	Fact = 2.5	Fact = 3.4	Fact = 2.61
63	Act3 (GCO = 0.5)	Fact = 2.51	Fact = 2.51	Fact = 2.51	Fact = 2.65
64	Act4 (GCO = 0.5)	Fact = 2.53	Fact = 2.53	Fact = 2.53	Fact = 1.29
65	Act1 (GCO = 0.5)	Fact = 2	Fact = 2.54	Fact = 2.54	Fact = 1.29
66	Act2 (GCO = 0.5)	Fact = 2.01	Fact = 2.01	Fact = 2.55	Fact = 1.30
67	Act3 (GCO = 0.5)	Fact = 2.01	Fact = 2.01	Fact = 2.01	Fact = 1.30
68	Act1 (GCO = 0.5)	Fact = 1.66	Fact = 2.02	Fact = 2.02	Fact = 1.30
69	Act2 (GCO = 0.5)	Fact = 1.67	Fact = 1.67	Fact = 2.03	Fact = 1.30
70	Act3 (GCO = 0.5)	Fact = 1.67	Fact = 1.67	Fact = 1.67	Fact = 1.31
71	Act1 (GCO = 0.5)	Fact = 1.42	Fact = 1.68	Fact = 1.68	Fact = 1.31
72	Act2 (GCO = 0.5)	Fact = 1.43	Fact = 1.43	Fact = 1.69	Fact = 1.31
73	Act3 (GCO = 0.5)	Fact = 1.43	Fact = 1.43	Fact = 1.43	Fact = 1.31
74	Act1 (GCO = 0.5)	Fact = 1.24	Fact = 1.44	Fact = 1.44	Fact = 1.32
75	Act2 (GCO = 0.5)	Fact = 1.24	Fact = 1.24	Fact = 1.44	Fact = 1.32
76	Act3 (GCO = 0.5)	Fact = 1.25	Fact = 1.25	Fact = 1.25	Fact = 1.32
77	Act4 (GCO = 0.5)	Fact = 1.25	Fact = 1.25	Fact = 1.25	Fact = 0.875
78	Act1 (GCO = 0.5)	Fact = 1.108	Fact = 1.26	Fact = 1.26	Fact = 0.876
79	Act2 (GCO = 0.5)	Fact = 1.11	Fact = 1.11	Fact = 1.26	Fact = 0.877
80	Act3 (GCO = 0.5)	Fact = 1.11	Fact = 1.11	Fact = 1.11	Fact = 0.879
81	Act1 (GCO = 0.5)	Fact = 0.99	Fact = 1.21	Fact = 1.21	Fact = 0.88
82	Act2 (GCO = 0.5)	Fact = 0.99	Fact = 0.99	Fact = 1.21	Fact = 0.88
83	Act3 (GCO = 0.5)	Fact = 1	Fact = 1	Fact = 1	Fact = 0.88
84	Act1 (GCO = 0.5)	Fact = 0.90	Fact = 1	Fact = 1	Fact = 0.88
85	Act2 (GCO = 0.5)	Fact = 0.90	Fact = 0.90	Fact = 1	Fact = 0.88
86	Act3 (GCO = 0.5)	Fact = 0.91	Fact = 0.91	Fact = 0.91	Fact = 0.88
87	Act1 (GCO = 0.5)	Fact = 0.82	Fact = 0.91	Fact = 0.91	Fact = 0.88
88	Act2 (GCO = 0.5)	Fact = 0.83	Fact = 0.83	Fact = 0.91	Fact = 0.88
89	Act3 (GCO = 0.5)	Fact = 0.83	Fact = 0.83	Fact = 0.83	Fact = 0.89
90	Act4 (GCO = 0.5)	Fact = 0.83	Fact = 0.83	Fact = 0.83	Fact = 0.75
91	Act1 (GCO = 0.5)	Fact = 0.76	Fact = 0.83	Fact = 0.83	Fact = 0.75
92	Act2 (GCO = 0.5)	Fact = 0.76	Fact = 0.76	Fact = 0.84	Fact = 0.75
93	Act3 (GCO = 0.5)	Fact = 0.77	Fact = 0.77	Fact = 0.77	Fact = 0.75
94	Act1 (GCO = 0.5)	Fact = 0.71	Fact = 0.77	Fact = 0.77	Fact = 0.76
95	Act2 (GCO = 0.5)	Fact = 0.71	Fact = 0.71	Fact = 0.77	Fact = 0.76
96	Act3 (GCO = 0.5)	Fact = 0.71	Fact = 0.71	Fact = 0.71	Fact = 0.76
97	Act4 (GCO = 0.5)	Fact = 0.71	Fact = 0.71	Fact = 0.71	Fact = 0.75
98	Act4 (GCO = 0.5)	Fact = 0.71	Fact = 0.71	Fact = 0.71	Fact = 0.75
99	Act4 (GCO = 0.5)	Fact = 0.72	Fact = 0.72	Fact = 0.72	Fact = 0.74

Tabla 33. Resultados del MAS Actividad para el Obj4 beneficiando y perjudicando con QB de 31, 31, 31 y 7

Iter.	Actividad Actual	Agente Actividad 1	Agente Actividad 2	Agente Actividad 3	Agente Actividad 4
50	Act4 (GCO = 0.5)	Fact = 20.27	Fact = 20.27	Fact = 20.27	Fact = 7.38
51	Act1 (GCO = 0.5)	Fact = 9.98	Fact = 20.39	Fact = 20.39	Fact = 7.40
52	Act2 (GCO = 0.5)	Fact = 10	Fact = 10	Fact = 20.50	Fact = 7.42
53	Act3 (GCO = 0.5)	Fact = 10	Fact = 10	Fact = 10	Fact = 7.44
54	Act1 (GCO = 0.5)	Fact = 4.98	Fact = 10.16	Fact = 10.16	Fact = 7.46
55	Act2 (GCO = 0.5)	Fact = 5	Fact = 5	Fact = 10.21	Fact = 7.48
56	Act3 (GCO = 0.5)	Fact = 5.03	Fact = 5.03	Fact = 5.03	Fact = 7.5
57	Act4 (GCO = 0.5)	Fact = 5.06	Fact = 5.06	Fact = 5.06	Fact = 3.69
58	Act1 (GCO = 0.5)	Fact = 3.33	Fact = 5.09	Fact = 5.09	Fact = 3.70
59	Act2 (GCO = 0.5)	Fact = 3.34	Fact = 3.34	Fact = 5.11	Fact = 3.71
60	Act3 (GCO = 0.5)	Fact = 3.36	Fact = 3.36	Fact = 3.36	Fact = 3.725
61	Act4 (GCO = 0.5)	Fact = 3.38	Fact = 3.38	Fact = 3.38	Fact = 1.83
62	Act1 (GCO = 0.5)	Fact = 2.5	Fact = 3.4	Fact = 3.4	Fact = 1.84
63	Act2 (GCO = 0.5)	Fact = 2.51	Fact = 2.51	Fact = 3.41	Fact = 1.84
64	Act3 (GCO = 0.5)	Fact = 2.53	Fact = 2.53	Fact = 2.53	Fact = 1.85
65	Act1 (GCO = 0.5)	Fact = 2	Fact = 2.54	Fact = 2.54	Fact = 1.85
66	Act2 (GCO = 0.5)	Fact = 2.01	Fact = 2.01	Fact = 2.55	Fact = 1.85
67	Act3 (GCO = 0.5)	Fact = 2.01	Fact = 2.01	Fact = 2.01	Fact = 1.86
68	Act1 (GCO = 0.5)	Fact = 1.66	Fact = 2.02	Fact = 2.02	Fact = 1.86
69	Act2 (GCO = 0.5)	Fact = 1.67	Fact = 1.67	Fact = 2.03	Fact = 1.87
70	Act3 (GCO = 0.5)	Fact = 1.67	Fact = 1.67	Fact = 1.67	Fact = 1.875
71	Act4 (GCO = 0.5)	Fact = 1.68	Fact = 1.68	Fact = 1.68	Fact = 1.23
72	Act1 (GCO = 0.5)	Fact = 1.43	Fact = 1.69	Fact = 1.69	Fact = 1.23
73	Act2 (GCO = 0.5)	Fact = 1.43	Fact = 1.43	Fact = 1.70	Fact = 1.24
74	Act3 (GCO = 0.5)	Fact = 1.44	Fact = 1.44	Fact = 1.44	Fact = 1.24
75	Act1 (GCO = 0.5)	Fact = 1.24	Fact = 1.44	Fact = 1.44	Fact = 1.24
76	Act2 (GCO = 0.5)	Fact = 1.25	Fact = 1.25	Fact = 1.45	Fact = 1.24
77	Act3 (GCO = 0.5)	Fact = 1.259	Fact = 1.259	Fact = 1.259	Fact = 1.25
78	Act1 (GCO = 0.5)	Fact = 1.1	Fact = 1.26	Fact = 1.26	Fact = 1.25
79	Act2 (GCO = 0.5)	Fact = 1.11	Fact = 1.11	Fact = 1.26	Fact = 1.25
80	Act3 (GCO = 0.5)	Fact = 1.11	Fact = 1.11	Fact = 1.11	Fact = 1.25
81	Act4 (GCO = 0.5)	Fact = 1.12	Fact = 1.12	Fact = 1.12	Fact = 0.93
82	Act1 (GCO = 0.5)	Fact = 0.99	Fact = 1.12	Fact = 1.12	Fact = 0.93
83	Act2 (GCO = 0.5)	Fact = 1	Fact = 1	Fact = 1.12	Fact = 0.93
84	Act3 (GCO = 0.5)	Fact = 1	Fact = 1	Fact = 1	Fact = 0.93
85	Act1 (GCO = 0.5)	Fact = 0.90	Fact = 1	Fact = 1	Fact = 0.93
86	Act2 (GCO = 0.5)	Fact = 0.91	Fact = 0.91	Fact = 0.95	Fact = 0.94
87	Act3 (GCO = 0.5)	Fact = 0.91	Fact = 0.91	Fact = 0.91	Fact = 0.94
88	Act4 (GCO = 0.5)	Fact = 0.91	Fact = 0.91	Fact = 0.91	Fact = 0.74
89	Act1 (GCO = 0.5)	Fact = 0.83	Fact = 0.92	Fact = 0.92	Fact = 0.74
90	Act2 (GCO = 0.5)	Fact = 0.83	Fact = 0.83	Fact = 0.92	Fact = 0.74
91	Act3 (GCO = 0.5)	Fact = 0.83	Fact = 0.83	Fact = 0.83	Fact = 0.75
92	Act1 (GCO = 0.5)	Fact = 0.76	Fact = 0.84	Fact = 0.84	Fact = 0.75
93	Act2 (GCO = 0.5)	Fact = 0.77	Fact = 0.77	Fact = 0.84	Fact = 0.75
94	Act3 (GCO = 0.5)	Fact = 0.77	Fact = 0.77	Fact = 0.77	Fact = 0.75
95	Act1 (GCO = 0.5)	Fact = 0.71	Fact = 0.77	Fact = 0.77	Fact = 0.75
96	Act2 (GCO = 0.5)	Fact = 0.71	Fact = 0.71	Fact = 0.77	Fact = 0.75
97	Act3 (GCO = 0.5)	Fact = 0.71	Fact = 0.71	Fact = 0.71	Fact = 0.75
98	Act4 (GCO = 0.5)	Fact = 0.72	Fact = 0.72	Fact = 0.72	Fact = 0.75
99	Act4 (GCO = 0.5)	Fact = 0.72	Fact = 0.72	Fact = 0.72	Fact = 0.74

Tabla 34. Resultados del MAS Actividad para el Obj1 sólo beneficiando con QB de 31, 31, 31 y 10

Bibliografía

- [Aguilar 2003a] Aguilar R., Bruno A. González C., Moreno L., Muñoz V., Noda A. "Teaching Mathematics to Children with Down's Syndrome". Proceedings of AIED Math Workshop. Australia 2003.
- [Aguilar 2003b] R. M. Aguilar, V. Muñoz, C. González, L. Moreno. "Interfaz Gráfica basada en web para la individualización de las actividades propuestas por un tutorial inteligente". Actas del IV Congreso Multimedia Educativo, Barcelona 2003.
- [Aguilar 2003c] R. M. Aguilar, A. Bruno, C. González, L. Moreno, V. Muñoz, A. Noda. "Tutorial Inteligente para el refuerzo de la suma en alumnos con Síndrome de Down". Actas de las XI Jornadas sobre el Aprendizaje y Enseñanza de las Matemáticas, Tenerife – Gran Canarias, 2003.
- [Aguilar 2004] R. M. Aguilar, V. Muñoz, E. J. González, C. S. González, M. A. Noda, A. Bruno, L. Moreno. "Design of an Instruccional Planner for an Intelligent Tutoring System using Fuzzy Methodology and MAS". WSEAS Transactions on Systems, Vol 3 (10), 2261-2266, 2004.
- [Aguilar 2005a] R. M. Aguilar, V. Muñoz, E. J. González, C. S. González, M. A. Noda, A. Bruno, L. Moreno. "Design of a web-based framework using XML and JavaScript". Proceedings of the 4th WSEAS International Conference on System Science and Engineering, ICOOSSE 05, Brasil, 2005

- [Aguilar 2005b] R. M. Aguilar, V. Muñoz, E. J. González, C. S. González, M. A. Noda, A. Bruno, L. Moreno. "Using Fuzzy and MultiAgent Methodology in an Instructional Planner for an Intelligent Tutorial System". Proceedings of 4th International Workshop on Practical Applications of Agents and MultiAgents Systems, IWPAAMS'05, León, España, 2005.
- [Aguilar 2005c] R. M. Aguilar, V. Muñoz, C. S. González, M. A. Noda, A. Bruno, L. Moreno. "Pensamiento Lógico Matemático en alumnos con Síndrome de Down". Actas de las XII Jornadas sobre el Aprendizaje y Enseñanza de las Matemáticas, Albacete, 2005.
- [Aguilar 2005d] R. M. Aguilar, A. Bruno, C. S. González, V. Muñoz, M. A. Noda, L. Moreno. "Análisis de un Tutorial Inteligente sobre conceptos matemáticos en Alumnos con Síndrome de Down". Actas del III Congreso Internacional da Ensino da Matematica, Brasil, 2005.
- [Aguilar 2006] Aguilar R. M., Bruno A., González C., Moreno L., Muñoz V., Noda A. "Logical-Mathematical Learning for Student with Down's Syndrome". Proceedings of the 30th Conference of the International Group for the Psychology of Mathematics Education, en Novotna J. et al., Vol 1. 369. Praga 2006.
- [Aguilar 2007] R. M. Aguilar, V. Muñoz, M. A. Noda, A. Bruno, L. Moreno. "Teacher strategies by using fuzzy systems". Computer Applications in Engineering Education (en prensa), 2007.

- [Alonso 2000] García Alonso y J. Pavón Mestras. "Introducción al estándar FIPA". Technical Report UCM-DSIP 98-00, Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, 2000.
- [Alonso 2002] C. Alonso, D. Gallego. "Ley de calidad, Tecnologías de la Información y la Comunicación". Revista de Educación del Ministerio de Educación, Cultura y Deportes, 2002.
- [Ammar 2004] Ammar S., Duncombe W., Jump B., Wright R., "Constructing a fuzzy-knowledge-based-system: an application for assesing the financial condition of public schools". Expert System with Applications, Vol 27, 349-364, 2004.
- [Anderson 1990] Anderson, Cobett, Petterson. "Student modeling and tutoring flexibility in the Lisp". Frasson y Gauthier (eds). Intelligent Tutoring Systems. Ablex Publisher. 1990.
- [Apco 2005] Apache Cocoon Project. <http://cocoon.apache.org>. 2005.
- [Bachant 1984] Bachant J., McDermott J "R1 Revisited: Four Years in the Trenches" AI Magazine 5(3), 21-32, 1984.
- [Bellifemine 1999] Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. "JADE. A FIPA-compliant agent framework". Proceedings of PAAM'99, 97-108, 1999.
- [Bevington 1999] Bevington J. and Wishart, J.G. "The influence of classroom peers on cognitive performance in children with behavioural problems". British Journal of Educational Psychology, 69, 19-32, 1999.

- [Bruno 2006] Alicia Bruno, María Noda, Rosa Aguilar, Carina Conzález, Lorenzo Moreno, Vanesa Muñoz, Análisis de un tutorial inteligente sobre conceptos lógico-matemáticos en alumnos con Síndrome de Down. *Relime* Vol. 9, Núm. 2, 211-226, 2006.
- [Brustolini 1991] J. C. Brustolini. "Autonomous agents: characterization and requirements". Technical Report CMU-CS-91-204, Carnegie-Mellon University, 1991.
- [Buchanan 1984] Buchanan B.G., Shortliffe E.H. "Rule-Based Expert Systems, The MYCIN Experiments of the Stanford Heuristic Programming Project" Buchanan and Shortliffe (eds) Addison-Wesley, Reading, MA. 1984.
- [Buchanan 1989] B. G. Buchanan, D. Bobrow, R. W. Davis, J. McDermott, E. H. Shortliffe. "Knowledge-based systems" *Annual Review of Computer Science*, Vol. 4, 395-416, 1989.
- [Burns 1988] Burns, Capps, "Foundation of Intelligent Tutoring Systems: An Introduction" M. Polson & Richardson (eds). *Foundation of Intelligent Tutoring Systems*. Lawrence Erlbaum Associates Publishers, USA. 1988.
- [Burns 1991] Burns H., Parlett J.W., Redfiel C.L., *Intelligent Tutoring System. Evolutions in Design*, Lawrence Erlbaum Associates Publishers, 1991.

- [Carbonell 1970] J. Carbonell, 1970. AI in CAI: An artificial intelligence approach to computer aided instruction. *Science*, 167, 190-202.
- [Cebrián 2000] M. Cebrián de la Serna, J. M. Ríos. Nuevas tecnologías aplicadas a las didácticas especiales. Ed. Pirámide. 2000.
- [Chen 1988] Peter, Pin-Shan Chen. "The entity-relationship model—toward a unified view of data". *Readings in database systems*, 374 – 387. 1988.
- [Coen 1995] M. Coen. "SodaBot: A Software Agent Construction System". 1995.
- [Cohen 1995] Philip R. Cohen and Hector J. Levesque. "Communicative Actions for Artificial Agents". In Victor Lesser and Les Gasser, editors, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95)*, 65-72, San Francisco, CA, USA. The MIT Press: Cambridge, MA, USA. 1995.
- [Cook 1998] John Cook, "PRACMA: Mentoring, Metacognition and Music: Interaction Analyses and Implications for Intelligent Learning Environments" *International Journal of Artificial Intelligence in Education* Vol. 9, 45-87, 1998.
- [Cuevas 1996] Cuevas, "Sistemas Tutoriales Inteligentes. Investigaciones en Matemática Educativa". Grupo Editorial Iberoamericana. 1996.

- [Dimitrova 1998] Vanya Dimitrova, Darina Dicheva. "Who is who: the roles in an intelligent system for foreign language terminology learning". *British Journal of Educational Technology* 29(1), 47-57. 1998.
- [Domingo 1999] M.Domingo, M.Martín-Baranera, F.Sanz, C.Sierra, M.J.Uriz, Validation SPONGIA, an expert system for sponge identification, *Expert Systems with Applications*, vol. 16, 1999.
- [EIS 2006] i2010 - A European Information Society for growth and employment. Europe's Information Society. 2006. http://europa.eu.int/information_society/eeurope/i2010/index_en.htm
- [Elorriaga 1999] Jon A. Elorriaga, "Instructional planning in intelligent evolutive tutoring systems from a case-based reasoning approach". *AI Commun* Vol 12, Núm. 4, 259-260, 1999.
- [Finin 1993] T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML . "A Language and Protocol for Knowledge and Information Exchange" *International Conference on Building and Sharing of Very Large-Scale Knowledge Bases*, Tokyo, 1993.
- [Finin 1994] T. Finin, R. Fritzson, D. McKay, and R. McEntire. "KQML as an Agent Communication Language" In N. Adam, B. Bhargava, and Y. Yesha, editors, *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)*, 456-463, Gaithersburg, MD, USA, ACM Press. 1994.

-
- [Finin 1997] Tim Finin, James May_eld, and Chelliah Thirunavukkarasu. "A Security Architecture for Agent Communication Language", 1997.
- [FIPA 1996] FIPARationale. "Technical report", FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, 1996.
- [FIPA 2001a] FIPA ACL Message Structure Specification. Technical Report XC00061E, FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, 2001.
- [FIPA 2001b] FIPA Agent Management Specification. Technical Report XC00023H, FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, 2001.
- [FIPA 2002] FIPA Abstract Architecture Specification. Technical Report SC00001L, FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, 2002.
- [FIPAOS 2001] "FIPA-OS Developers Guide". <http://fipa-os.sourceforge.net/tutorials.htm>, 2001.
- [Flores 1999] Roberto A. Flores-Mendez. "Towards a Standardization on a Multi-Agent System Frameworks". 1999.
- [Franklin 1996] S. Franklin and A. Graesser. "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents". Intelligent Agents III. Agent Theories, Architectures and Languages (ATAL'96), Vol. 1193. Springer-Verlag. 1996.

- [Genesereth 1997] Michael R. Genesereth and Steven P. Ketchpel. "Software Agents". *Communications of the ACM*, 37. 1997.
- [Gilbert 1997] Don Gilbert. "Intelligent Agents: The Right Information at the Right Time". 1997.
- [González 2004] E. González. "Diseño e Implementación de una Arquitectura Multipropósito basada en Agentes Inteligentes: Aplicación a la Planificación Automática de Agendas y al Control de Procesos". Tesis Doctoral. Universidad de La Laguna, 2004.
- [González 2006] E. J. González, A. Hamilton, L. Moreno, R. Marichal, V. Muñoz. "Software experience when using ontologies in a multi-agent system for automated planning and scheduling". *Software - Practice and Experience*, Vol. 36 (7), 667-688, 2006.
- [Gruber 1993] T. R. Gruber. "A translation approach to portable ontologies". *Knowledge Acquisition*, 5(2), 199-220, 1993.
- [Hallowell 1996] Hallowell and Ratey. "Answers to Distraction". Bantam Trade Pbk. 1996.
- [Hayes 1995] B Hayes-Roth. "An architecture for adaptive intelligent systems". *Artificial Intelligence*, 92, 329-365, 1995.
- [Hewitt 1977] C. Hewitt. "Viewing Control structures as Patterns of Passing Messages". *Artificial Intelligence* Vol 8, 323-364, 1977.

- [Huhns 1999] Michael N. Huhns and Larry M. Stephens. "Multiagent Systems and Societies of Agents". In Gerhard Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pp.: 79--120. The MIT Press, Cambridge, MA, USA, 1999.
- [IRT 2001] Infraestructuras y Redes de Telecomunicaciones en la Comunidad Foral de Navarra. Observatorio de la Sociedad de la Información en Navarra. 2001.
- [Jacobson 1997] I. Jacobson, *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, 1997.
- [Jang 1997] Jang, Sun, Mizutani, "Neuro-Fuzzy and Sof Computing", Prentice Hall, 1997.
- [Java 2006] <http://www.w3.schools.com/js> 2006.
- [Jennings 1998] N.R. Jennings, K. Sycara, and M. Wooldridge. "A Roadmap of Agent Research and Development". *Autonomous Agents and Multi-Agent Systems Journal*, Vol. 1, 7-38, 1998.
- [Jennings 1998] N.R. Jennings, K. Sycara, and M. Wooldridge. "A Roadmap of Agent Research and Development". *Autonomous Agents and Multi-Agent Systems Journal*, Vol , 7-38, 1998.

-
- [Labrou 1999] Yannis Labrou, Tim Finin, and Yun Peng. "The current landscape of Agent Communication Languages". *Intelligent Systems*, N. 14, Vol. 2, 1999.
- [Laukkanen 2000] M. Laukkanen. "Evaluation of FIPA-OS 1.03". Technical report, Cellular System Development, Sonera Mobile Operator, Sonera Ltd., Helsinki, Finlandia, 2000.
- [Lester 1997] James Lester, Sharolyns Converse, Susan Kahler, Todd Barlow, Brian Stone and Ravinder Bhogal. "The Persona Effect: Affective Impact of Animated Pedagogical Agents". *Proceedings of CHI'97*, 359-366. Atlanta 1997.
- [Maes 1994] Pattie Maes. "Modeling adaptive autonomous agents". *Artificial Life*, I, 1994.
- [Maes 1997] P. Maes. "Agents that Reduce Work and Information Overload". In Jeffrey M. Bradshaw editor, *Software Agents*. AAAI Press/MIT Press, 1997.
- [Magoulas 2001] G.D. Magoulas, K.A. Papanikolaou, and M. Grigoriadou, "Neuro-fuzzy Synergism for Planning the Content in a Web-based Course". *Informática* Vol. 25, 2001.
- [Mamdani 1975] Mamdani E. y Assilian S. "An experiment in linguistic synthesis with a fuzzy logic controller". *International Journal of Man-Machine Studies*, 7(1), 1-13. 1975.

- [Matos 1998] N. Matos, C. Sierra, and N. R. Jennings. "Determining successful negotiation strategies: an evolutionary approach". In Y. Demazeau, editor, Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS-98), 182-189, Paris, France. IEEE Press. 1998.
- [Matsuda 2000] Noboru Matsuda, Kurt VanLehn, "Decision Theoretic Instructional Planner for Intelligent Tutoring Systems". ITS2000, 72-83, 2000.
- [Mayer 2000] Mayer R.E. and Moreno R. "Life-like Pedagogical Agents in Constructivist Multimedia Environment: Cognitive Consequences of their Interaction". World Conference on Educational Multimedia, Hypermedia and Telecommunication (ED-Media). Montreal 2000.
- [McEntire 1998] Robin McEntire and Donald McKay. "KQML Lite languages and services". AAAI-98 Workshop on Software Tools for Developing Agents, 1998.
- [Moreno 2003] L. Moreno, R. M. Aguilar, C. González, V. Muñoz, A. Bruno, A. Noda. "The Design of an Intelligent Tutorial for the addition operation in children with learning difficulties. Part I: The Fuzzy Instructional Planner". Proceedings of E-Learn, Phoenix, 2003.
- [Mosquera 2001] E. Mosquera, V. Moret. "Validation of intelligent systems: a critical study and a tool, Expert Systems with Applications", Vol. 18, 2001.

- [Mosquera 2002a] E. Mosquera, V. Moret, Intelligent interpretation of validation data, *Expert Systems with Applications*, vol. 23, 2002.
- [Mosquera 2002b] Mosquera, V. Moret, *Validación de Sistemas Inteligentes*, Tórculo Edicións, 2002.
- [Nwana 1995] H. S. Nwana. "Software Agents: An Overview". *Knowledge Engineering Review*, Vol 11-2, 205-244, 1995.
- [Nwana 1999] Hyacinth S Nwana, Divine T. Ndumu, Lyndon C. Lee, and Jaron C. Collis. "ZEUS: a toolkit and approach for building distributed multi-agent systems". In Oren Etzioni, J. P. Müller, and Jeffrey M. Bradshaw, editors, *Proceedings of the Third International Conference on Autonomous Agents*, Seattle, WA, USA. ACM Press. 1999.
- [OMG 2000] Agent Technology. Green paper. Technical Report agent/00-09-01, Object Management Group, 2000.
- [Passino 1998] Passino, K., Yurkovich, S., "Fuzzy Control". Addison-Wesley, 1998.
- [Piszcz 1998] Alan Piszcz. "A Brief Overview of Software Agent Technology". Technical report, 1998.
- [Reeves 1996] Reeves B. and Nash C. "The Media Equation". Stanford CA: CSLI Publications. 1996.

- [Reigeluth 1987] C. M. Reigeluth, "Instructional Theories in Action: Lessons Illustrating Selected Theories and Models". 1987.
- [Remolina 2002] Emilio Remolina, Sowmya Ramachandran, Daniel Fu, Richard Stottler y William R. Howse, "Intelligent Simulation-Based Tutor for Flight Training". ITSEC 2002.
- [Santally 2005] Mohammad Issack Santally, Alain Senteni. "A Learning Object Approach to Personalized Web-based Instruction". European Journal of Open, Distance and E-Learning Vol I, 2005.
- [Sargent 1998] R.G. Sargent, Verification and Validation of Simulation Models, Proceedings of the 1998 Winter Simulation Conference, 1998.
- [SI 2003] Sociedad de la Información en el siglo XXI: un requisito para el desarrollo. Madrid: Ministerio de Industria, Turismo y Comercio. División de Información, Documentación y Publicaciones. 2003.
- [Sierra 1997] C. Sierra, P. Faratin, and N. Jennings. "A Service-Oriented Negotiation Model between Autonomous Agents". Proceedings of the 8th European Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW-97), 17-35, Ronneby, Sweden, 1997.
- [Silberschatz 2005] A. Silberschatz. "Conceptos de Sistemas Operativos". Addison-Wesley. 2005.

- [Sleeman 1982] D. Sleeman y J. Brown, "Intelligent Tutoring Systems" Academic Press. 1982.
- [Smith 1994] D.C. Smith, A. Cypher, and J. Spohrer. "KidSim: Programming agents without a programming language". Communications of the ACM, Vol 37, 55-67, 1994.
- [Sperberg 1999] C.M. Sperberg-McQueen and Lou Burnard. Guidelines for Electronic Text Encoding and Interchange (TEI Guidelines). TEI P3 Text Encoding Initiative, revised reprint edition, 1999.
- [Stallings 2002] William Stallings. "Sistemas Operativos". Prentice-Hall. 2002.
- [Sturm 2000] J. Sturm. Desarrollo de Soluciones XML. McGrawHill Iberoamericana, 2000.
- [Telefonica 2003] La Sociedad de la Información en España. <http://www.telefonica.es/sociedaddelainformacion>. 2003.
- [Towns 1998] Stuart Towns, Patrick Fitzgerald and James Lester. "Visual Emotive Communication in Lifelike Pedagogical Agents". Proceedings of the Four International Conference on Intelligent Tutoring Systems. 474-483. San Antonio, 1998.
- [Tyler 1975] R. W. Tyler, "Educational benchmarks in retrospect: Educational change since 1915". 1975.

- [Vassileva 1996] Julita Vassileva, Barbara Wasson. "Instructional Planning Approaches". EuroAIED'96. 1996.
- [Vasudevan 1998] Venu Vasudevan. "Comparing Agent Communication Languages". Technical report, Object Services and Consulting, Inc, 1998.
- [Warendorf 1997] Kai Warendorf, Haiping Xu y Antoon Verhoeven, "Case-Based Instructional Planning for Learning in a Context". Proceedings of Joint 1997 Pacific Asia Conference on Expert System and Singapore International Conference on Intelligent Systems. PACES/SPICIS97, Singapore 354-367, 1997.
- [Woldridge 1994] Michael Wooldridge and Nicholas R. Jennings. "Intelligent Agents: Theory and Practice". <http://www.doc.mmu.ac.uk/STAFF/mike/ker95/ker95.html.h> (Hypertext version of Knowledge Engineering Review paper). 1994.
- [Zlatareva 1998] N.P. Zlatareva, A refinement framework to support validation and maintenance of knowledge-based systems, Expert Systems with Applications, vol. 15, 1998.