

## Sistema Genético Difuso con Aplicación en Psicología Infantil

Evelina Meza Diyarza, P. Malaquías Quintero Flores, Federico Ramírez Cruz y Carlos Pérez Corona

Facultad de Ciencias Básicas Ingeniería y Tecnología de la UATx,  
Laboratorio de Investigación en Tecnologías Inteligentes del DSC-JT Apizaco,  
Avenida Instituto Tecnológico S/N, Apizaco Tlaxcala, México, CP. 90300.  
malakaz@prodigy.net.mx

*Paper received on 12/08/08, accepted on 11/09/08.*

**Resumen:** Este trabajo presenta la realización de un sistema genético difuso basado en multimedia aplicado a la detección y diagnóstico de habilidades en infantes, el cual cuenta con un entorno de trabajo formado de tres interfaces de usuario una en multimedia para interactuar con los infantes, otra para la interacción con el especialista en Psicología Infantil, y la tercera para la interacción con el desarrollador del sistema. El sistema genético difuso cuenta con la capacidad de aprendizaje de reglas lingüísticas difusas mediante programación genética. El sistema cuenta también con otras herramientas que le permiten al desarrollador del sistema definir variables lingüísticas, términos lingüísticos, definir y afinar los parámetros de las correspondientes funciones de pertenencia. La efectividad del sistema genético difuso se demostró al aplicarlo en la detección de las habilidades de 49 infantes, los resultados obtenidos fueron comparados con los proporcionados por un especialista en Psicología Infantil.

**Palabras Clave:** sistema genético difuso, programación genética, modelado difuso, multimedia, reglas difusas, modelos de inferencia difusa.

### 1. Introducción

Los sistemas difusos (SD) han demostrado su enorme potencial para la solución de problemas complejos de clasificación, modelado, diagnóstico, toma de decisiones, y control, en un vasto número de aplicaciones [01][02][08]. En muchos de los casos, la clave del éxito de los SD radicó en su habilidad de razonamiento aproximado ante conocimiento impreciso e incierto [03][04]. No obstante dicho éxito, en los noventa los SD fueron criticados por su nula capacidad de aprendizaje, por lo que en el contexto de la inteligencia computacional surgieron dos opciones de hibridación la de modelos neuronales – SD y la de algoritmos evolutivos – SD o sistemas genético difusos (SGD) [11][12].

Este trabajo presenta la realización de un SGD basado en multimedia aplicado a la detección y diagnóstico de habilidades en infantes, con un entorno de trabajo formado de tres interfaces de usuario una en multimedia para interactuar con los infantes, otra para la interacción con el especialista en Psicología Infantil, y la tercera para la interacción con el desarrollador del sistema la cual cuenta con herramientas que

permiten definir variables lingüísticas, términos lingüísticos, definir y afinar los parámetros de las correspondientes funciones de pertenencia, generar la base de reglas lingüísticas difusas mediante programación genética [13][14], y aplicar el mecanismo de inferencia difusa adecuado.

La efectividad del sistema difuso se demostró al aplicarlo en la detección de las habilidades de 49 infantes, los resultados obtenidos fueron comparados con los proporcionados por un especialista en Psicología Infantil.

Habilidad: es hablar de una disposición natural o adquirida en un campo determinado del comportamiento. Una habilidad es una capacitación intelectual que una vez activada facilita el aprendizaje, la ejecución o la retención de una tarea, es decir, para la ejecución como rango definido de una habilidad de aprendizaje [05][06].

La detección oportuna de las habilidades de un infante es de gran utilidad en la toma de decisiones, tanto para sus maestros con el fin de lograr una mejora continua del proceso de enseñanza aprendizaje a lo largo de su instrucción primaria, como para sus padres con el fin de alcanzar un desarrollo integral de su vida familiar y posteriormente profesional [06].

En el contexto de la Psicología Infantil y Pedagogía, se han desarrollado diferentes instrumentos encaminados a dar apoyo en la detección, estudio y desarrollo de habilidades en Infantes, siendo uno de ellos el de los juegos interactivos tales como: Baúl, Laberintos, Rompecabezas, Analogías, Series, entre otros [06][07]. Cada uno de ellos dirigido a identificar en los Infantes respectivamente la habilidad *visual*, *de atención*, *percepción visual espacial e integración*, *abstracción*, etc.

Tradicionalmente la aplicación de tales juegos se realiza de manera manual, lo cual resulta demasiado tedioso, cansado, tardado y poco eficiente, tanto para los Infantes como para el responsable de la identificación de las correspondientes *habilidades*, situación que ha motivado la iniciativa de desarrollar un SGD que basado en multimedia permita por una parte aplicar los juegos de aptitudes en un ambiente agradable y de simple manejo para los Infantes, y por otra parte basado en un sistema de inferencia genética difusa que desempeñe el papel de una útil herramienta de especialistas en Psicología Infantil para la identificación eficiente de habilidades en Infantes [09][10].

## 2. Desarrollo de la Interfaz de Interacción Humano – Máquina

La interfaz de interacción humano – sistema genético difuso se desarrolló para tres tipos de usuario: Infante, Psicólogo y desarrollador del sistema, la interfaz de interacción del infante - sistema difuso consta de siete juegos interactivos, organizados como se ilustra en la Figura 1, mismos que se implementaron en Macromedia Flash. La interfaz Psicólogo – Sistema Genético Difuso programada en Java consta de cuatro cuadros de dialogo que se muestran en la Figura 2. de los cuales los dos primeros le permiten al Psicólogo registrar sus datos personales y dar de alta su correspondiente cuenta de usuario, los dos últimos le permite registrar en MySQL los datos

de los Infantes a evaluar y consultar los datos registrados de los Infantes u observar el desempeño de cada uno de ellos en los juegos de aptitudes. La interacción entre el Sistema Genético Difuso y el Desarrollador de la aplicación de dicho sistema se soporta mediante una interfaz basada en un escritorio de trabajo implementado en Java, en el cual el Desarrollador cuenta con los elementos requeridos para definir las variables lingüísticas de entrada y salida con sus respectivos términos lingüísticos y funciones de pertenencia, el acceso al sistema de inferencia difusa y los medios para la generación automática de la base de reglas lingüísticas mediante programación genética como se muestra en la Figura 3.

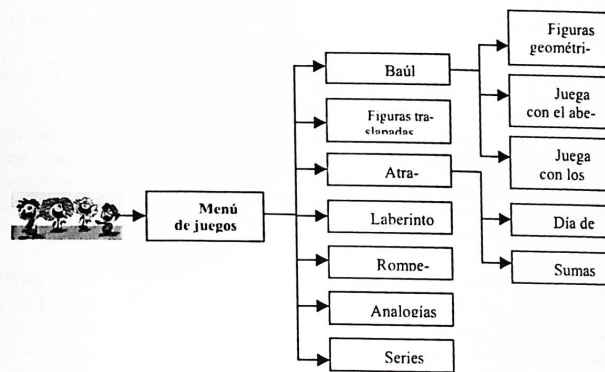


Fig. 1: Estructura de la interfaz infante – sistema difuso.

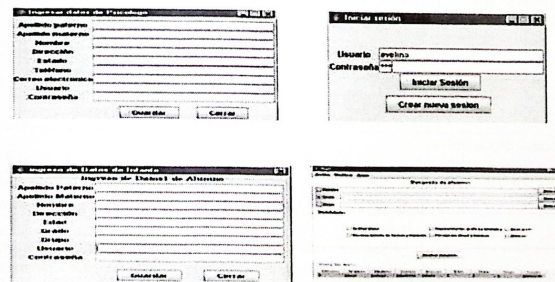


Fig. 2: Registro de datos generales y cuenta de usuario del Psicólogo.

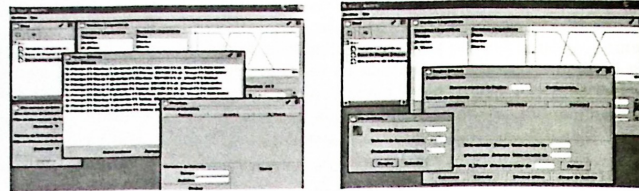


Fig. 3: Escritorio de trabajo del desarrollador de aplicaciones del sistema genético difuso.

### 3 Desarrollo del Sistema Genético Difuso

La generación de la base de reglas lingüísticas difusas se realiza mediante un sistema de programación genética, cuyo modelado e implementación se realizó como se explica a lo largo de la presente sección. Se definieron dos tipos de símbolos terminal (ST) cuya especificación se presenta en la Tabla 1, se definieron cuatro símbolos función (SF) cuyo nombre y descripción se explica en la Tabla 2.

En el algoritmo evolutivo la representación de una población de reglas lingüísticas difusas se hace en una estructura de árbol, como la que se ilustra en la Figura 4. La profundidad mínima permitida para un árbol de reglas difusas se estableció de 3 niveles, que corresponde a reglas difusas con un sólo predicado lingüístico difuso como antecedente. El número máximo de antecedentes por regla se determinó mediante la expresión  $2^{d-3}$ , donde  $d$  es un parámetro de control del algoritmo de programación genética que determina la profundidad máxima permitida del árbol de reglas. Los individuos a evolucionar son reglas lingüísticas difusas con una estructura descrita por una gramática de contexto libre y de expresión regular que se muestran en la Tabla 3.

Tabla 1: Definición de símbolos terminal.

Nombres de símbolos terminal	Descripción
<input type="checkbox"/> <VAR_LIN>	Son nombres de variables lingüísticas de entrada – salida, definidos desde el escritorio de trabajo como se ilustra en la Figura 12.
<input type="checkbox"/> <TERM_LIN>	Son los nombres de los términos lingüísticos - conjuntos difusos en que se ha distribuido el universo de discurso de cada variable lingüística de entrada- salida, definidos desde el escritorio de trabajo como se ilustra en la Figura 12.

Tabla 2: Definición del conjunto de símbolos función.

Nombre	Descripción
ANTE	Representa un predicado lingüístico difuso contenido en la parte de antecedentes de una regla difusa, teniendo como argumentos una variable lingüística de entrada con su respectivo término lingüístico, donde ANTE regresa un valor numérico en el intervalo [0, 1] que representa el grado de verdad de dicho predicado lingüístico.
CONSC	Es análogo al símbolo funcional ANTE, con la diferencia de que CONSC representa un predicado lingüístico difuso contenido en la parte de consecuentes de una regla difusa esto sólo para reglas difusas del tipo Mamdani o Larsen.
OPR_Y	Denota el operador de intersección difusa $\gamma$ ( $\min$ ) de la teoría de conjuntos difusos para realizar la intersección de dos o más ANTE's, y generar un valor numérico en el intervalo [0, 1] que denota el grado de verdad de la regla difusa en turno.
OPR_O	Desempeña el papel de un operador de agregación o unión ( $\max$ ) de la teoría de conjuntos difusos para conformar un solo conjunto difuso desde los conjuntos difusos generados por STR_SI-ENTONCES para cada regla difusa en la base de reglas difusas en evolución, por lo que OPR_O ocupa el lugar del nodo raíz en la representación de árbol de dicha base de reglas.
STR_SI-ENTONCES	Es una primitiva funcional de dos argumentos, siendo el primero el valor numérico generado por ANTE o OPR_Y, mientras que el segundo argumento es el valor numérico obtenido por CONSC, y así dar lugar a un conjunto difuso de salida.

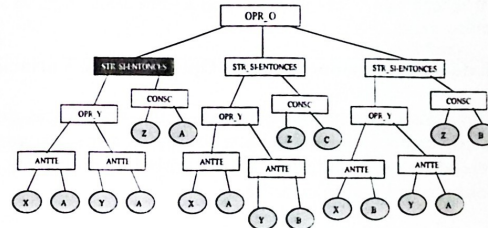


Fig. 4. Representación árbol de una base de reglas lingüísticas difusas (BRD).

Tabla 3: Descriptor gramatical de contexto libre ( $G_{CL}$ ) de la estructura sintáctica de reglas lingüísticas difusas validas.

Conjunto	Definición
$S_1$ :	{SI, ENTONCES, (, ), Y, ES, *, + ;}
$S_{NT}$ :	{<RD>, <ANTDTS>, <CONSC>, <PRED_LD>, <VAR_LIN>, <TERM_LIN>, <POLINOMIO>, <EXP>, <COEF>}
$S_N$ :	{<RD> ;}
$R_s$ :	{<RD> ::= SI <ANTDTS> ENTONCES <CONSC> <ANTDTS> ::= <PRED_LD> [Y <ANTDTS>]}

	$\langle \text{CONSCT} \rangle ::= \langle \text{PRED\_LD} \rangle [\text{Y} \langle \text{ANTDTS} \rangle] \mid \langle \text{POLINOMIO} \rangle$ $\langle \text{PRED\_LD} \rangle ::= \langle \text{VAR\_LIN} \rangle \text{ ES } \langle \text{TERM\_LIN} \rangle$ $\langle \text{POLINOMIO} \rangle ::= \langle \text{VAR\_LIN} \rangle = \langle \text{EXP} \rangle$ $\langle \text{EXP} \rangle ::= (\langle \text{COEF} \rangle * \langle \text{VAR\_LIN} \rangle) \{ [+ (\langle \text{COEF} \rangle * \langle \text{VAR\_LIN} \rangle) ]^* + \langle \text{COEF} \rangle \}$
Conjunto	Definición
$S_T$ :	$\{a, b, \dots, z, A, B, \dots, Z, 0, 1, 2, \dots, 9, .., +, -, \lambda\}$
$S_{NT}$ :	$\{\langle \text{VAR\_LIN} \rangle, \langle \text{TERM\_LIN} \rangle, \langle \text{COEF} \rangle, \langle \text{LETRA} \rangle, \langle \text{DIGITO} \rangle\}$
$S_{IN}$ :	$\{\langle \text{VAR\_LIN} \rangle, \langle \text{TERM\_LIN} \rangle, \langle \text{COEF} \rangle\}$
$R_s$ :	$\{\langle \text{VAR\_LIN} \rangle ::= \langle \text{LETRA} \rangle \langle \text{LETRA} \rangle^* \}$ $\langle \text{TERM\_LIN} \rangle ::= \langle \text{LETRA} \rangle \langle \text{LETRA} \rangle^* \}$ $\langle \text{COEF} \rangle ::= \langle \text{Sig} \rangle \langle \text{DIGITO} \rangle^* \}$ $\langle \text{LETRA} \rangle ::= a \mid b \mid \dots \mid z \mid A \mid B \mid \dots \mid Z$ $\langle \text{DIGITO} \rangle ::= 0 \mid 1 \mid \dots \mid 9$ $\langle \text{Sig} \rangle ::= - \mid \lambda$

La población inicial consiste de dos árboles de reglas difusas contruidos al azar de acuerdo a la estructura de árbol descrita anteriormente, con un número de reglas  $NR_C$  por árbol especificado dentro del intervalo  $[NR_{MIN}, NR_{MAX}]$ , mismo que es fijado durante la configuración del sistema de programación genética.  $NR_{MIN}$ , es interpretado como el tamaño mínimo de la base de reglas con las que el sistema difuso es capaz de dar soluciones satisfactorias al problema para el cual se esté modelando,  $NR_{MAX}$  viene siendo el tamaño máximo de la base de reglas elegido de tal forma que  $NR_{MAX} \geq NR_C$ , donde  $NR_C$  representa el número de reglas requerido para la base de reglas difusas en evolución.

### 3.1 Modelado e Implementación de los Operadores de Variación

Para la identificación evolutiva de la base de reglas lingüísticas difusas, los dos árboles de reglas de la población inicial son sometidos a los operadores de cruza, mutación, y/o reproducción, para formar el tercer árbol de reglas que sustituye únicamente al menos apto de los árboles padre.

En este trabajo el operador de cruza se implemento de acuerdo a un conjunto de lineamientos que se describen en la Tabla 4, desarrollados para preservar la estructura sintáctica correcta del árbol de reglas resultante de la acción de dicho operador genético. Una vez elegidos los padres, se procede a determinar los puntos de cruza en ambos padres, realizar el intercambio de material genético y con ello construir el nuevo árbol de reglas difusas como se describe en los árboles de la Figura 5.

Tabla 4: Lineamientos para la implementación del operador de cruza.

N	Descripción	Propósito
1	La población actual de individuos está conformada de dos árboles de reglas difusas, cada uno conteniendo un número de reglas compren-	Tener en cuenta que las reglas padre se obtienen desde dos bases de

	dido en el intervalo $[NR_{MIN}, NR_{MAX}]$ .	reglas difusas.
2	La primera regla padre se obtiene desde uno de los árboles de reglas Figura 4.9, mientras que la segunda regla padre se obtienen del segundo árbol de reglas Figura 4.10, eligiéndose al azar cual de los dos árboles de reglas será el primero y cual el segundo cada vez que se aplique la cruza.	Aclarar que sólo de un árbol de reglas se puede obtener una sola regla difusa padre identificado por el funcional STR_SI-ENTONCES, de cada par seleccionado.
3	El punto de cruza se define seleccionando al azar cualquier nodo no STR_SI-ENTONCES en el primer padre, restringiéndose el punto de cruza del segundo padre a ser seleccionado al azar siempre que sea del mismo tipo que en el primero, a excepción de los nodos ANTTE, y OPR_Y siempre que no se altere la profundidad permitida del nuevo árbol de reglas.	Asegurar que el intercambio de material genético entre padres sea del mismo tipo, contexto, y significado.
4	De los nodos correspondientes a símbolos terminales, sólo se permite elegir como punto de cruza a nodos <TERM_LIN> siempre que en ambos padres correspondan a conjuntos difusos distintos pero pertenecientes a la misma variable lingüística.	Que la cruza entre nodos terminales sólo se de con términos lingüísticos distintos pero que correspondan en contexto y significado.
5	La cruza se realiza intercambiando el material genético contenido en los puntos de cruza de los dos padres.	Que tenga lugar la cruza.
6	Las dos reglas hijo se insertan en el nuevo árbol de reglas como sub-árboles del nodo raíz OPR_O.	Asegurar que las reglas hijo se incorporen a la nueva población de reglas difusas.

La implementación del operador de mutación se realizó de acuerdo a las cuatro consideraciones que se describen en la Tabla 5 y su aplicación se ilustra en la Figura 6.

Tabla 5: Implementación del operador de mutación.

No	Descripción
1	La regla difusa hijo a ser mutada es seleccionada de acuerdo a un porcentaje bajo definido por el usuario.
2	Se selecciona aleatoriamente el subárbol a ser mutado en la regla difusa hijo que haya sido elegida.
3	Se remplacea el subárbol seleccionado por otro creado aleatoriamente, siempre que éste concuerde con el tipo de símbolos terminal/función del subárbol a sustituir y sin rebasar la profundidad máxima permitida del árbol de reglas difusas, de lo contrario se crea otro subárbol que cumpla con tales condiciones.
4	Los nodos de símbolos función OPR_O y STR_Si-Entonces no pueden ser objeto de mutación.

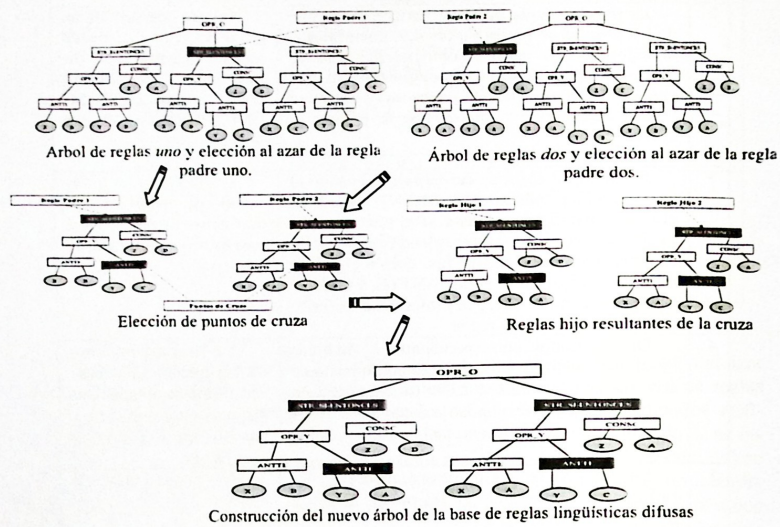


Figura 5: Ilustración de la implementación del operador de cruce..

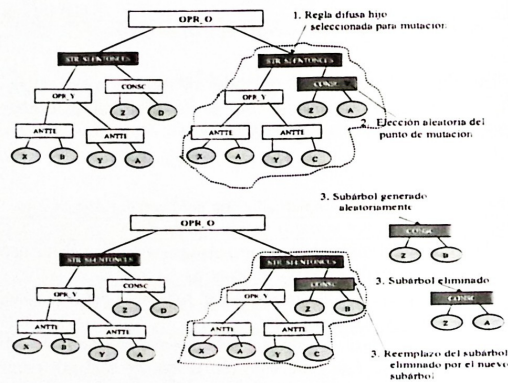


Figura 6: Ilustración de la implementación del operador de mutación.

Para formar la población de la generación siguiente se toman en cuenta sólo los dos árboles de reglas difusas con las aptitudes más altas de los tres árboles de reglas

disponibles los dos árboles padre y el árbol de reglas hijo, lo que significa que siempre por lo menos uno de los árboles de reglas padre por reproducción pasar a formar parte de la población de la siguiente generación. Tal aptitud se obtiene mediante la función de aptitud de error absoluto o distancia Minkowski como se define en (1)

$$f(p) = \sum_{i=1}^n |\inf(v_{e_i}) - v_{s_i}| \quad (1)$$

Donde:

$p$ : es la población de reglas difusas en turno.

$n$ : es el número de datos de entrenamiento/prueba proporcionados.

$\inf()$ : es el resultado de la inferencia difusa aplicada a la base de reglas  $p$  con  $v_e$ .

$v_e$ : es el vector de datos de entrada de entrenamiento/prueba del sistema.

$v_s$ : es el vector de datos de salida esperados de entrenamiento/prueba del sistema.

El criterio de terminación adoptado radica en agotar un determinado número de casos de entrenamiento/prueba por generación o bien hasta obtener una base de reglas con una aptitud esperada menor o igual a la unidad, formalmente expresada en (2).

$$\forall x_i \in v_e, y_i \in v_s \quad |\inf(x_i) - y_i| \leq 1 \quad (2)$$

#### 4 Trabajo Experimental

Las variables lingüísticas de entrada – salida identificadas para cada juego fueron tres cuya definición de términos lingüísticos y FPs se realizó como se ejemplifica en las ilustraciones de la Figura 7 y con los datos de la Tabla 6.

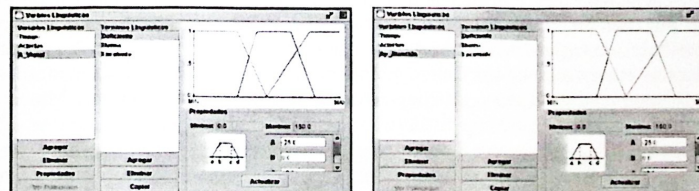


Figura 7: Definición de variables lingüísticas de entrada – salida del sistema difuso para los juegos baúl y laberinto.

**Tabla 6:** Ejemplos de definición de parámetros de FP trapezoidales y triangulares para los juegos baúl y laberinto.

Variable Lingüística	Términos Lingüísticos	Parámetros de FPs			
		a	b	c	d
Tiempo	Poco	-20	0	10	22.5
	Normal	10	15	25	35
	Mucho	22.5	30	45	55
Aciertos	Pocos	-30	0	60	95
	Muchos	50	100	145	193
	Todos	140	193	200	
Aptitud Visual	Deficiente	-25	0	45	75
	Buena	45	65	100	125
	Excelente	75	120	150	160

Variable Lingüística	Términos Lingüísticos	Parámetros de FPs			
		a	b	c	d
Tiempo	Poco	-5	0	3	7.5
	Normal	2.5	5.5	8.5	10
	Mucho	7.5	10	24	25
Aciertos	Pocos	-2	0	5	7
	Muchos	4	6	9	10
	Todos	8	10	12	
Aptitud Atención	Deficiente	-25	0	45	75
	Buena	45	65	100	125
	Excelente	75	120	150	160

El SGD en las etapas de entrenamiento y de prueba, se aplicó para identificar aptitudes en cuarenta y nueve infantes de acuerdo a su desempeño observado en la ejecución de tres juegos de aptitudes el de *baúl*, *laberintos* y *rompecabezas*. En la etapa de entrenamiento el SGD para cada juego de aptitud se sometió a doce corridas con 10 generaciones por corrida y al final obtener una base de 9 reglas lingüísticas difusas por cada juego que se muestran en la Figura 8. Para la etapa de prueba las mismas bases de reglas lingüísticas difusas, se emplearon para identificar las aptitudes de otros cuarenta y nueve infantes. El mecanismo de inferencia difusa empleado para la interpretación de las reglas lingüísticas difusas fue el Mamdani Max-Min.

Reglas Difusas	
Si (Tiempo ES Mucho) Y (Aciertos ES Pocos) ENTONCES (A_Visual ES Deficiente)	
Si (Tiempo ES Normal) Y (Aciertos ES Pocos) ENTONCES (A_Visual ES Deficiente)	
Si (Tiempo ES Poco) Y (Aciertos ES Muchos) ENTONCES (A_Visual ES Buena)	
Si (Tiempo ES Poco) Y (Aciertos ES Pocos) ENTONCES (A_Visual ES Deficiente)	
Si (Tiempo ES Poco) Y (Aciertos ES Todos) ENTONCES (A_Visual ES Excelente)	
Si (Tiempo ES Mucho) Y (Aciertos ES Muchos) ENTONCES (A_Visual ES Deficiente)	
Si (Tiempo ES Mucho) Y (Aciertos ES Todos) ENTONCES (A_Visual ES Buena)	
Si (Tiempo ES Normal) Y (Aciertos ES Muchos) ENTONCES (A_Visual ES Buena)	
Si (Tiempo ES Normal) Y (Aciertos ES Todos) ENTONCES (A_Visual ES Excelente)	

**Figura 8:** Bases de reglas lingüísticas difusas generadas por el SPG.

## 6. Interpretación de Resultados

En los resultados obtenidos por la Psicóloga y el SGD en la detección de las aptitudes visual, atención y abstracción, con los datos de entrenamiento, se observó un desempeño caracterizado por los datos que se resumen en la Tabla 7 y se ilustran en las Figura 9 a), b), y c) respectivamente. En cuanto a los resultados obtenidos con los datos de prueba, se observó un desempeño caracterizado por los datos que se resumen en la Tabla 8 e ilustran en las gráficas de la Figura 10 a), b), y c).

**Tabla 7:** Desempeño del SGD con los datos de entrenamiento.

Juego	Casos	Error Absoluto Total	Error Esperado	Error Máximo	Error Mínimo	Aciertos del SGD	
						Totales	%
Baúl	50	108.62	2.22	4.57	0.06	42	85.71
Laberintos	50	186.9	3.81	22.66	0.04	41	83.60
Rompecabezas	50	273	5.57	27.66	0.77	38	77.55

**Tabla 8:** Evaluación del desempeño del SGD en la fase de prueba.

Juego	Caso	Error Absoluto Total	Error Esperado	Error Máximo	Error Mínimo	Aciertos del SD	
						Totales	Porcentaje
Baúl	49	104.3	2.12	14	0.00	41	84 %
Laberintos	49	159.7	3.26	27.6	0.04	33	67.35 %
Rompecabeza	49	215	4.38	27.6	0.49	35	71.42 %

## 7. Conclusiones y Recomendaciones

En el presente documento se presentó el desarrollo de un novedoso sistema genético difuso basado en multimedia, con la capacidad de desempeñar el papel de una útil herramienta para especialistas en Psicología Infantil y Pedagogía en la realización de la difícil y valiosa tarea de identificar y diagnosticar habilidades intelectuales en Infantes, mediante la interpretación lingüística y numérica de su desempeño en la ejecución de diferentes juegos de aptitudes tales como Baúl, Laberintos, y Rompecabezas. En el documento se explicaron los diferentes componentes que se incorporaron al sistema difuso entre los que se encuentran la interfaz amigable infante – sistema difuso basada en multimedia, la interfaz Psicólogo – sistema genético difuso basada en un escritorio de trabajo implementado en Java que permite dar seguimiento al desempeño de los infantes y diagnóstico de sus aptitudes, la interfaz ingeniero del conocimiento – sistema genético difuso basada en un amigable escritorio de trabajo implementado en Java que permite la definición de variables lingüísticas, términos lingüísticos, el ajuste de parámetros de funciones de pertenencia y la construcción semiautomática o automática de la base de reglas lingüísticas difusas, el modelado difuso del sistema de toma de decisiones basado en un sistema de infe-

rencia difusa Mamdani, el diseño e implementación de un sistema de programación genética para la generación automática de la base de reglas lingüísticas difusas mediante aprendizaje supervisado. El documento finaliza con la explicación de los resultados obtenidos en la fase de entrenamiento y en la fase de prueba a las que fue sometido al sistema genético difuso para diagnosticar las aptitudes de percepción visual, atención y abstracción de 49 infantes mediante los juegos de *baúl*, *laberintos* y *rompecabezas* respectivamente.

Las recomendaciones de trabajo futuro que se sugieren son: a) aplicar el sistema genético difuso en el diagnóstico de más aptitudes de grupos más grandes de infantes, b) incorporar en el sistema difuso las adecuaciones pertinentes para dotarlo de la capacidad ejecución distribuida y con ello dar mayor versatilidad y capacidad de atención a infantes en una red de área local o intranet, y c) incorporar un sistema multiagente en la interfaz infante – sistema difuso.

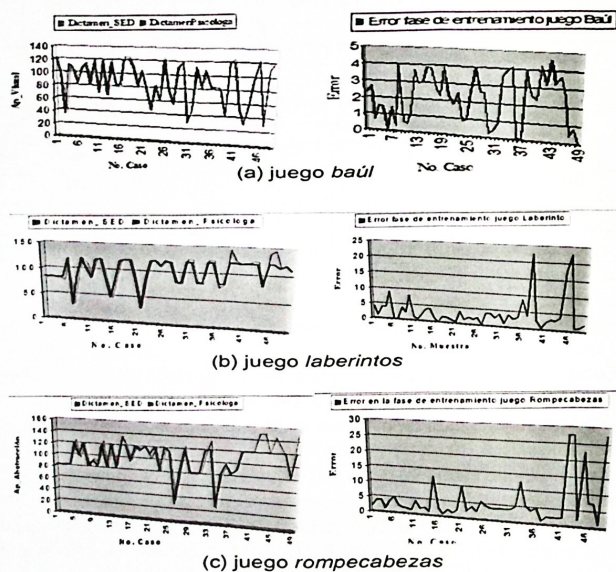


Figura 9: Desempeño del SD – DAI en la fase de entrenamiento para la detección de la aptitud a) visual, b) atención, y c) abstracción.

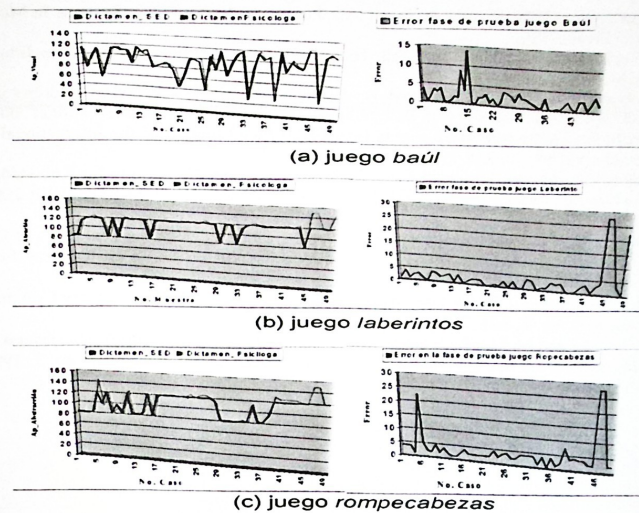


Figura 10: Desempeño del SGD en la fase de prueba para la detección de la aptitud a) visual y b) atención.

## Referencias

- 1 Zargham M. R. y Morgharrebán N. (2005); PORSEL: an expert system for assisting in investing in investment analysis and valuation; En *Soft Computing: a fusion of foundations, methodologies and applications*, Springer, volume 9, number 10, October 2005, pp. 742 – 748.
- 2 Jang J-SR, Sun C-T y Mizutani E. (1997) ; *Neuro – Fuzzy and Soft Computing* ; Prentice Hall.
- 3 Buckley J. J.; *Fuzzy Systems*; En *Soft Computing: a fusion of foundations, methodologies and applications*, Springer, volume 9, number 10, October 2005, pp. 757 – 760.
- 4 Siler, William; *Fuzzy Eexpert Systems and Fuzzy Reasoning*; John Wiley & Sons, Incorporated; 2005.
- 5 Victor Matías Rodríguez, “Psicotécnica pedagógica teoría y practica”, Editorial Porrúa, S. A.
- 6 Yolanda Benito; “Es importante detectar en que sobresalen los infantes”, *Revista Gaceta Universitaria de la Facultad de Ciencias de la Educación, Universidad Autónoma de Tlaxcala*, pp. 6 – 10, julio 1999.
- 7 Gardner, Howard; *La Mente No Escolarizada: Como Piensan Los Niños Y Como Deberían Enseñar Las Escuelas*; Ediciones Paidós Iberica, S.A.; 2002.

- 8 Teodorescu H-N, Kandel A., Jain L. C.; Fuzzy and Neuro-Fuzzy Systems in Medicine; CRC Preee, 1999.
- 9 Joseph Beck, Mia Stern, and Erik Haugsjaa (2001); Applications of AI in Education; ACM.
- 10 Robertson, J., Cross, B., Macleod, H., & Wiemer-Hastings, P. (2004); Children's interactions with animated agents in an Intelligent Tutoring System; en International Journal of Artificial Intelligence in Education (IJAIED), official journal of the International AIED Society, vol 14, pp. 335-357.
- 11 Takeshi Furuhashi; Fusion of Fuzzy/Neuro/Evolutionary Computing for Knowledge Acquisition; en PROCEEDINGS OF THE IEEE. VOL. 89, NO. 9, SEPTEMBER 2001, pp. 1266 – 1274.
- 12 Cerdón O., Herrera F., Hoffmann F., Comide F., y Magdalena L.; Ten Years of Genetic Fuzzy Systems: Current Framework and New Trends; en ELSEVIER Fuzzy Sets and systems, No 141, 2004, pp. 5 – 31.
- 13 M. Russo; Genetic Fuzzy Learning; in IEEE Transactions on Evolutionary Computation, Volume 4, No. 2, July 2000, pp. 259 – 273.
- 14 E. Tunstel and M. Jamshidi, *On Genetic Programming of fuzzy rule-based systems for intelligent control*, in Intelligent Automation and Soft Computing, vol. 2, no. 3, 1996, pp. 271 – 284.