

Lengoaia eta Sistema Informatikoen Saila  
Departamento de Lenguajes y Sistemas Informáticos

*Framework* para el desarrollo y entrenamiento  
de Sistemas de Inferencia Difusa siguiendo  
Métodos de Desarrollo Dirigido por Modelos

por

Unai Segundo Santamaría

Dirigida por Dr. Juan Manuel Pikatza Atxa y Dr. Tomás A. Pérez Fernández

Tesis presentada al Departamento de Lenguajes y Sistemas Informáticos de la  
Universidad del País Vasco / Euskal Herriko Unibertsitatea (UPV/EHU) para  
optar al título de Doctor en Ingeniería Informática

Donostia - San Sebastián, 2017



*Para mi familia*



---

## Agradecimientos

En primer lugar quiero agradecer a mi familia el apoyo y los ánimos que me han infundido en los años que ha durado este proceso. Ver reflejado en sus ojos el orgullo por su hijo, nieto y sobrino ha sido la fuente de energía que me ha permitido no tirar la toalla y haber llegado hasta aquí.

También quiero agradecer a mis directores Juanma y Tomás la oportunidad que me brindaron. He podido realizar un trabajo que me ha gustado y que me ha permitido vivir experiencias que no habría vivido de haber tomado otro camino. De la misma forma, también quiero agradecer a Javilo los esfuerzos realizados y la ayuda que me ha brindado durante este largo proceso. Un especial agradecimiento al equipo médico del grupo, el cual me ha permitido aprender muchas cosas ajenas a mi especialidad y me ha posibilitado tener una visión diferente de la informática.

Por supuesto, estoy muy agradecido a mis compañeros de despacho David, Raúl y Ander por toda la ayuda que me han brindado en el desarrollo de la tesis y el grado de complicidad que hemos adquirido. Es difícil poder imaginar un grupo más competente y profesional a la hora de desarrollar un trabajo. Ojalá en un futuro pueda trabajar junto a compañeros que sean igual de buenos que vosotros. Gracias.

Tampoco puedo olvidar a mis compañeros de facultad, por los ánimos y los consejos, así como por las comidas, las cenas y en general los buenos ratos que hemos pasado. Momentos que espero que se sigan repitiendo.

De todo este grupo, en concreto quiero expresarle mi agradecimiento a Itziar por el cariño que me ha dado y las fuerzas que me ha infundido durante los últimos meses de este proceso, a la postre los que son psicológicamente los más duros, porque la oscuridad siempre es mayor justo antes del amanecer. Muchas gracias.

Por último, no puedo olvidar a mi cuadrilla y mis amigos. Es un placer que me tengáis en tan alta estima y que siempre habléis bien de mí delante de otras personas. Por supuesto, seguro que se me olvida gente a la que extender mi agradecimiento, a todos ellos, gracias.

Milesker denoi.



---

## Índice general

Índice de figuras .....	XIII
Índice de tablas .....	XVII
Resumen .....	XXIII
Laburpena .....	XXV
Abstract .....	XXVII
Abreviaturas y acrónimos .....	XXIX

---

### Parte I Prolegómenos

---

<b>1. Introducción</b> .....	5
1.1. Contexto de trabajo .....	10
1.2. Organización de la tesis .....	11
1.3. Guía de lectura .....	12

---

### Parte II Base Teórica

---

<b>2. Métodos de Desarrollo Dirigido por Modelos</b> .....	19
2.1. Introducción a los Métodos de Desarrollo Dirigido por Modelos	19
2.2. Metamodelado .....	20
2.3. Herramientas para los Métodos de Desarrollo Dirigido por Modelos .....	25
<b>3. Guías Clínicas y Guías Clínicas Informatizadas</b> .....	27
3.1. Guías Clínicas .....	27
3.2. Guías Clínicas Informatizadas .....	29

3.3.	Formalismo de representación de GCI Aide .....	31
3.3.1.	Sintaxis abstracta .....	31
3.3.2.	Sintaxis concreta .....	34
<b>4.</b>	<b>Aide Guideline Technology Platform</b> .....	<b>39</b>
4.1.	Vista general de Aide Guideline Technology Platform .....	39
4.2.	Aide Developer Suite .....	40
4.3.	Aide Guideline Execution Platform .....	42
<b>5.</b>	<b>Sistemas de Inferencia Difusa</b> .....	<b>45</b>
5.1.	Lógica difusa .....	46
5.1.1.	Tipos de funciones de pertenencia .....	50
5.1.2.	Operaciones lógicas difusas .....	51
5.2.	Sistemas de inferencia difusa (Fuzzy inference systems) .....	53
5.2.1.	Estructura .....	53
5.2.2.	Proceso de razonamiento .....	54
5.2.3.	Sistemas Mamdani .....	56
5.2.4.	Sistemas Takagi-Sugeno .....	59
5.2.5.	Otros SID .....	61
5.2.5.1.	Clasificadores difuso relacionales (Fuzzy relational classifiers) .....	61
5.2.5.2.	Sistemas de inferencia híbrida difuso-evidente (Fuzzy-evidential hybrid inference systems) ...	61
5.2.5.3.	Sistemas de inferencia difusa con módulos de reglas de entrada única (Single input rule modules connected fuzzy inference model) ....	62
5.2.5.4.	Sistemas difusos probabilísticos (Probabilistic fuzzy systems) .....	62
5.3.	<i>Frameworks</i> para el desarrollo de Sistemas de Inferencia Difusa	62
<b>6.</b>	<b>Entrenamiento de SID mediante Algoritmos Genéticos</b> ....	<b>65</b>
6.1.	Operación de cruce .....	67
6.1.1.	Cruce simple (Simple crossover) .....	69
6.1.2.	Cruce Max-Min-Aritmético (Max-Min-Arithmetic crossover) .....	69
6.1.3.	Cruce dinámico (Dynamic crossover) .....	70
6.1.4.	Cruce dinámico heurístico (Dynamic heuristic crossover) .....	72
6.1.5.	Cruce <i>Blend Alpha</i> ( $BLX-\alpha$ ) .....	73
6.1.6.	Cruce <i>Parent-centric BLX-<math>\alpha</math></i> ( $PBX-\alpha$ ) .....	73
6.2.	Operación de mutación .....	74
6.2.1.	Mutación aleatoria .....	74
6.2.2.	Mutación gaussiana .....	74
6.2.3.	Mutación de Michalewicz .....	75

---

**Parte III Estado del arte**

---

<b>7. Sistemas de clasificación integrados en formalismos de GCI</b>	81
7.1. Tipo de sistema de clasificación	82
7.1.1. Sistemas basados en reglas	83
7.1.2. Árboles de decisión	83
7.1.3. Redes neuronales artificiales	83
7.1.4. Sistemas de inferencia difusa basados en redes adaptativas	83
7.1.5. Máquinas de soporte vectorial	84
7.1.6. Sistemas basados en razonamiento probabilístico	84
7.1.7. Sistemas basados en votación	85
7.1.8. Mapas cognitivos difusos	86
7.1.9. Sistemas basados en argumentación	86
7.2. Relación entre el formalismo y el sistema de clasificación	86
7.3. Revisión de los formalismos de representación de GCI	87
7.3.1. PROforma	87
7.3.2. GLIF3	88
7.3.3. GLARE	88
7.3.4. NewGuide	89
7.3.5. SAGE	89
7.3.6. Asbru	90
7.4. Síntesis y comparativa	90
<b>8. Sistemas de Inferencia Difusa utilizados en medicina</b>	93
8.1. Análisis de las soluciones	93
8.2. Síntesis y comparativa	102
<b>9. Frameworks para el desarrollo de Sistemas de Inferencia Difusa</b>	107
9.1. Fuzzy Logic Toolbox de Matlab	108
9.2. Xfuzzy 3.0	109
9.3. FuzzyTech	110
9.4. GUAJE FUZZY	111
9.5. Linguistic Fuzzy Logic Controller	112
9.6. Síntesis y comparativa	113

---

**Parte IV Solución propuesta**

---

<b>10. Modelado de DD y SID</b>	121
10.1. Metamodelo de Diagnóstico Diferencial	123
10.2. Metamodelo de Sistema de Inferencia Difusa	124
10.3. Metamodelos de datos de entrada y resultados	128

X	Índice general	
	10.4. Transformaciones de modelo a modelo.....	132
	10.5. Transformaciones de modelo a texto .....	135
<b>11.</b>	<b>Herramientas para la definición de DD y SID.....</b>	<b>137</b>
	11.1. Aide Differential Diagnosis Editor .....	137
	11.2. Aide Fuzzy Inference System Framework .....	140
	11.2.1. Interfaz de Aide-FISF .....	140
	11.2.2. Funcionalidades del menú <i>opciones</i> .....	144
	11.2.2.1. Cambio de tipo de variables de entrada .....	144
	11.2.2.2. Generación automática de Sistemas de Inferencia Difusa ejecutables .....	145
	11.2.2.3. Entrenamiento de modelos de Sistemas de Inferencia Difusa .....	146
	11.2.2.4. Ejecución de modelos de Sistemas de Inferencia Difusa .....	150
<b>12.</b>	<b>Evaluación de la solución propuesta.....</b>	<b>153</b>
	12.1. Evaluación de la eficacia de Aide-DDE y Aide-FISF.....	153
	12.2. Evaluación de la eficacia de los SID generados.....	159
	12.2.1. Diseño de un modelo de SID para el cribado neonatal ..	160
	12.2.2. Pruebas realizadas.....	162
	12.3. Evaluación de los resultados .....	163
	12.3.1. Eficacia de Aide-DDE y Aide-FISF para desarrollar SID en términos de tiempo .....	163
	12.3.2. Eficacia de los SID generados en términos de acierto y tiempo .....	166
<hr/>		
<b>Parte V Conclusiones</b>		
<hr/>		
<b>13.</b>	<b>Principales aportaciones .....</b>	<b>171</b>
<b>14.</b>	<b>Líneas futuras .....</b>	<b>175</b>
<b>15.</b>	<b>Publicaciones generadas .....</b>	<b>177</b>
	<b>Referencias .....</b>	<b>181</b>
<hr/>		
<b>Parte VI Anexos</b>		
<hr/>		
<b>A.</b>	<b>Trastornos evaluados en el DD de cribado neonatal.....</b>	<b>201</b>
<b>B.</b>	<b>Variables clínicas tenidas en cuenta en el DD de cribado neonatal .....</b>	<b>205</b>

Índice general      XI

**C. Diagnóstico Diferencial de cribado neonatal..... 207**

**D. Resultados obtenidos por el SID de cribado neonatal ..... 215**



---

## Índice de figuras

1.1.	Muestra de un algoritmo en la GC para el diagnóstico y tratamiento de la ITU (extraído de (Trapote et al., 2011)) . . . . .	6
1.2.	Muestra de un <i>score</i> de una guía para el tratamiento del asma (extraído de (Rico, 2013)) . . . . .	7
1.3.	Diagnóstico Diferencial para la sospecha de etiología más probable causante de hiperamonemia (extraído de (Couce et al., 2009)) . . . . .	8
2.1.	Metamodelo de un modelo de dominio de un sistema de información . . . . .	21
2.2.	Arquitectura de modelado según el OMG . . . . .	22
3.1.	Muestra de una tabla de DD para la identificación de la etiología más probable en niños con hiperamonemia (extraído y adaptado de la página 33 de (Couce et al., 2009)) . . . . .	28
3.2.	Ejemplo de pregunta exclusiva (izquierda), inclusiva (centro) y de entrada (derecha) . . . . .	32
3.3.	Metamodelo de GCI en el formalismo Aide (parte 2) . . . . .	34
3.4.	Metamodelo de GCI en el formalismo Aide (parte 2) . . . . .	35
3.5.	Situación del A-POM respecto al modelo de arquetipos de OpenEHR . . . . .	36
3.6.	Estructuras <i>Node</i> , <i>Section</i> y <i>Set</i> (izquierda) y tipos de nodos (derecha) . . . . .	37
4.1.	Vista general de la plataforma Aide-GTP . . . . .	40
4.2.	Interfaz de Aide-MAT . . . . .	41
4.3.	Interfaz de Aide-Portal . . . . .	43
5.1.	Ejemplo de un conjunto clásico . . . . .	47
5.2.	Ejemplo de un conjunto difuso . . . . .	47
5.3.	Conjuntos difusos que componen la <i>Temperatura del agua</i> . . . . .	48

5.4.	Conjuntos difusos que componen el <i>Colesterol LDL</i> . . . . .	49
5.5.	Funciones de pertenencia de tipo lineal y constante . . . . .	50
5.6.	Funciones de pertenencia de tipo triangular y trapezoidal . . . . .	50
5.7.	Funciones de pertenencia de tipo gaussiana y de campana . . . . .	51
5.8.	Funciones de pertenencia de tipo sigmoidea y polinómica . . . . .	51
5.9.	Operaciones lógicas <i>AND</i> , <i>OR</i> y <i>NOT</i> para los conjuntos clásicos y difusos . . . . .	52
5.10.	Ejemplo de aplicación de una operación de unión en un antecedente con dos variables . . . . .	54
5.11.	Proceso de implicación simple . . . . .	55
5.12.	Proceso de agregación . . . . .	56
5.13.	Proceso de inferencia de un SID de tipo Mamdani . . . . .	58
5.14.	Proceso de inferencia de un SID de tipo Takagi-Sugeno . . . . .	60
6.1.	Relación entre los métodos de búsqueda y el espacio de búsqueda	69
6.2.	Posición de $h_i^1$ , $h_i^2$ , $h_i^3$ y $h_i^4$ mediante un cruce Max-Min- Aritmético cuando $c_i^1 < c_i^2$ y $\lambda = 0,25$ . . . . .	70
6.3.	Evolución de las posiciones de los cuatro genes creados mediante el cruce dinámico, en función de $t$ , siendo $t \in [1, g_{max}]$	72
6.4.	Evolución de las posiciones de los dos genes creados mediante el cruce dinámico heurístico, en función de $t$ , siendo $t \in [1, g_{max}]$ y suponiendo que $C_1$ tiene mejor aptitud que $C_2$ y siendo $c_i^1 \leq c_i^2$ . . . . .	73
6.5.	Intervalo de valores posibles mediante el cruce BLX- $\alpha$ con un valor de $\alpha = 0,5$ . . . . .	73
6.6.	Rango de valores posibles para un gen del cromosoma $H_1$ mediante el cruce PBX- $\alpha$ con un valor de $\alpha = 0,25$ . . . . .	74
6.7.	Intervalo de valores de una mutación gaussiana . . . . .	75
6.8.	Evolución del intervalo de valores de la mutación de Michalewicz siendo $v = 0,5$ , $u = 0$ y $w = 1$ . . . . .	75
10.1.	Muestra de un DD . . . . .	121
10.2.	Muestra de otro DD . . . . .	122
10.3.	Metamodelos y transformaciones de la solución propuesta . . . . .	123
10.4.	Metamodelo de DD . . . . .	123
10.5.	Metamodelo de SID . . . . .	125
10.6.	Metamodelo de los datos de entrada . . . . .	128
10.7.	Metamodelo de los resultados . . . . .	129
10.8.	Fragmento de los resultados correspondientes al SID sobre la identificación de la etiología más probable causa de hiperamonemia . . . . .	130
10.9.	Índices de proximidad para los distintos valores lingüísticos de la variable clínica <i>Gasometría</i> en relación al conjunto de trastornos <i>Trastornos de la cetogénesis</i> . . . . .	134

10.10 Conjunto de reglas que describen la relación entre la variable de entrada *Gasometría* y la variable de salida *Trastornos de la cetogénesis* ..... 134

10.11 Índices de proximidad para los distintos valores lingüísticos de la variable clínica *Gasometría* en relación al trastorno *Aciduria glutárica tipo 2* ..... 134

11.1. Interfaz principal de Aide-DDE ..... 138

11.2. Interfaz del editor de trastornos ..... 138

11.3. Interfaz del editor de variables clínicas ..... 139

11.4. Interfaz del editor de relaciones ..... 140

11.5. Interfaz principal de Aide-FISF ..... 141

11.6. Interfaz del editor de variables discretas ..... 142

11.7. Interfaz del editor de variables difusas ..... 143

11.8. Interfaz del editor de conjuntos difusos ..... 143

11.9. Previsualización de la variable de entrada *Amonio* ..... 144

11.10 Interfaz del editor de variables complejas ..... 145

11.11 Funcionalidad de generación automática de Aide-FISF ..... 146

11.12 Funcionalidad de entrenamiento de Aide-FISF ..... 146

11.13 Propiedades del algoritmo genético de Aide-FISF ..... 149

11.14 Funcionalidad de ejecución de Aide-FISF ..... 150

12.1. DD de sospecha de la hiperamonemia (extraído y adaptado de la página 33 de (Couce et al., 2009)) ..... 155

12.2. Relación entre el número de SID desarrollados y el tiempo necesario para desarrollarlos, utilizando Aide-DDE y Aide-FISF (línea continua) y manualmente (línea discontinua) . 165



---

## Índice de tablas

7.1. Comparativa sobre los sistemas de clasificación integrados en formalismos de representación de GCI.....	92
8.1. Comparativa sobre los soluciones en medicina que utilizan un SID.....	94
8.2. Comparativa sobre los soluciones en medicina que utilizan un SID (continuación) .....	95
8.3. Comparativa sobre los soluciones en medicina que utilizan un SID (continuación) .....	96
8.4. Resumen de la metodología de creación seguida por las distintas soluciones analizadas .....	103
8.5. Resumen del tipo de SID utilizado por las distintas soluciones analizadas .....	104
8.6. Resumen del tipo de SID utilizado por las distintas soluciones analizadas (continuación) .....	104
8.7. Resumen del tipo de FP utilizado por las distintas soluciones analizadas .....	105
8.8. Resumen del tipo de FP utilizado por las distintas soluciones analizadas (continuación) .....	105
8.9. Resumen del tipo de FP utilizado por las distintas soluciones analizadas (continuación) .....	105
9.1. Comparativa sobre los <i>frameworks</i> para la definición de SID ...	115
9.2. Comparativa sobre los <i>frameworks</i> para la definición de SID (continuación) .....	116
9.3. Comparativa sobre los <i>frameworks</i> para la definición de SID (continuación) .....	116
10.1. PF trapezoidales correspondientes a los valores lingüísticos de la variable <i>Amonio</i> para pacientes menores y mayores de 3 meses (extraído y adaptado de (Couce et al., 2009)) .....	127

XVIII Índice de tablas

10.2. Correspondencias entre los metamodelos de DD y SID .....	132
12.1. Resumen de los DD de la hiperamonemia.....	156
12.2. Tiempo necesario para crear el SID relativo al DD de sospecha de hiperamonemia .....	157
12.3. Tiempo necesario para crear el SID relativo al DD definitivo de hiperamonemia .....	158
12.4. Tiempo necesario para desarrollar la tecnología de la tesis .....	160
12.5. Resumen del DD de cribado neonatal .....	161
12.6. Resumen de las muestras utilizadas para evaluar el SID .....	162
12.7. Resumen de los resultados obtenidos .....	164
A.1. Lista de los trastornos de aminoácidos que se valoran en el DD de cribado neonatal .....	201
A.2. Lista de los trastornos de aminoácidos que se valoran en el DD de cribado neonatal (continuación) .....	202
A.3. Lista de los trastornos de ácidos orgánicos que se valoran en el DD de cribado neonatal .....	202
A.4. Lista de los trastornos de ácidos orgánicos que se valoran en el DD de cribado neonatal (continuación) .....	203
A.5. Lista de los trastornos que se valoran en el DD de cribado neonatal .....	204
B.1. Lista de los aminoácidos que se valoran en el DD de cribado neonatal .....	205
B.2. Lista de las acilcarnitinas que se valoran en el DD de cribado neonatal .....	206
B.3. Lista de las proporciones que se valoran en el DD de cribado neonatal .....	206
C.1. DD de cribado neonatal .....	207
C.2. DD de cribado neonatal (continuación) .....	208
C.3. DD de cribado neonatal (continuación) .....	209
C.4. DD de cribado neonatal (continuación) .....	210
C.5. DD de cribado neonatal (continuación) .....	211
C.6. DD de cribado neonatal (continuación) .....	212
C.7. DD de cribado neonatal (continuación) .....	213
D.1. Resultados obtenidos por el SID de cribado neonatal.....	215
D.2. Resultados obtenidos por el SID de cribado neonatal (continuación) .....	216
D.3. Resultados obtenidos por el SID de cribado neonatal (continuación) .....	217
D.4. Resultados obtenidos por el SID de cribado neonatal (continuación) .....	218

D.5. Resultados obtenidos por el SID de cribado neonatal (continuación) .....	219
D.6. Resultados obtenidos por el SID de cribado neonatal (continuación) .....	220
D.7. Resultados obtenidos por el SID de cribado neonatal (continuación) .....	221
D.8. Resultados obtenidos por el SID de cribado neonatal (continuación) .....	222
D.9. Resultados obtenidos por el SID de cribado neonatal (continuación) .....	223
D.10. Resultados obtenidos por el SID de cribado neonatal (continuación) .....	224



*Hay que reírse cuando dicen que la ciencia fracasa. Tontería: Lo que fracasa es la mentira; la ciencia marcha adelante.” [El árbol de la ciencia]*

*Pío Baroja (1872 - 1956)*



---

## Resumen

Este trabajo de tesis doctoral presenta un modelo independiente de la computación de un Diagnóstico Diferencial (DD), así como un modelo independiente de la plataforma de un Sistema de Inferencia Difusa. Se han utilizado los Métodos de Desarrollo Dirigido por Modelos (MDDM) en la concepción de los modelos, los cuales, además de facilitar la definición de los modelos, ofrecen herramientas para la realización de transformaciones entre ellos. Así, en el presente trabajo también se exponen las transformaciones entre los modelos de DD y SID y las transformaciones para la generación automática de SID expresados en lenguajes concretos a partir de los modelos de SID independientes de la plataforma. Los SID dependientes de la plataforma pueden ser incluidos en el formalismo de representación de Guías Clínicas Informatizadas (GCI) Aide.

Así mismo, en la tesis también se incluye una descripción de las herramientas que facilitan la definición de modelos de DD y SID, así como la generación automática de SID en lenguajes concretos utilizables en distintos motores de razonamiento. Es de reseñar la adición de un módulo de aprendizaje automático mediante un Algoritmo Genético que permite adaptar algunas características de los modelos de SID a los datos reales de entrenamiento.

Las herramientas y modelos se han validado en dos ámbitos. Por un lado, se han utilizado en el cribado neonatal, una prueba diagnóstica dirigida a la identificación presintomática de enfermedades graves con el fin de tratarlas precozmente y así prevenir y minimizar minusvalías neurológicas, orgánicas y psíquicas. Por otro lado, se han utilizado en el diagnóstico de la hiperamoniemia, una Enfermedad Rara que se debe tratar de forma urgente para evitar graves secuelas neurológicas e incluso la muerte. En ambos casos, los SID creados se han integrado en unas GCI para ser evaluados.



---

## Laburpena

Doktorego tesi lan honek konputazio kontzeptuetatik independente den Diagnostiko Diferentzialen (DD) eredu bat eta plataformatik independentea den Inferentzia Lausodun Sistemen (ILS) eredu bat aurkezten du. Ereduen kontzeptualizaziorako Ereduen Bidezko Garapen Metodoak (EBGM) erabili dira. Metodo hauek ereduaren definizioan laguntzeaz gain, ereduaren arteko transformazioak egiteko erremintak eskaintzen dituzte. Hortaz, aurkezten den lan honetan, DD eta ILS ereduaren arteko transformazioak eta plataformatik independente diren ILS ereduetatik lengoia espezifikoetan adierazitako ILS zehatzak lortzeko transformazioak ere azaltzen dira. Plataformatik dependente diren ILS zehatz hauek Aide izeneko Informatizatutako Guida Klinikoen (IGK) errepresentazio formalismoan barneratu daitezke.

Era berean, tesi lan honek DD eta ILS ereduak definitzeko eta arrazona-mendu motore ezberdinetan erabiltzeko ILS zehatzak automatikoki sortzeko erabiltzen diren erremintak deskribatzen ditu. Aipatzekoa da erreminta hauek ILS ereduaren zenbait ezaugarri hobetzeko Algoritmo Genetikoetan oinarritutako ikaste automatikoko sistema bat dutela.

Aipatutako ereduak eta erremintak bi arlo ezberdinetan ebaluatu dira. Alde batetik, jaioberrien beheketaren izeneko prozesuan erabili dira. Prozesu hau sintomak izan aurretik gaixotasun larriak identifikatzeko erabiltzen da. Modu honetan ondorio neurologiko, organiko edota psikiko larriak saihestu edota txikiagotu daitezke. Bestetik, aurkeztutako erreminta eta ereduak hiperamoniemia izeneko Gaixotasun Arraro baten diagnostikoan erabili dira. Gaixotasun honek ondorio neurologiko larriak eta akaso heriotza ekar dezake aurre egiteko tratamendua azkar eskaintzen ez bada. Bi kasuetan sortutako ILSak IGK batzuetan barneratu dira.



---

## Abstract

This PhD thesis work presents a Computational Independent Model (CIM) for Differential Diagnosis (DD) and a Platform Independent Model (PIM) for Fuzzy Inference System (FIS). These models have been built using Model Driven Development Methods (MDDM). These methods, apart from easing the definition process of models, they also offer resources for automatic transformation between them. Thus, the present work explains some transformations between DD and FIS models and some transformations for the automatic generation of FIS expressed in specific languages based in platform independent FIS models. These specific platform dependent FIS can be embedded in Aide, a representation formalism for Computerised Clinical Guidelines (CCG) definition.

Furthermore, this PhD thesis also describes a set of tools for the definition of DD and FIS models, as well as for the automatic generation of FIS expressed in specific languages which can be used in different types of reasoning engines. It is worth noting that these tools include a functionality for improving the characteristics of the defined FIS by means of an automatic learning system, specifically a Genetic Algorithm.

The described models and tools have been tested in two fields. First, they have been used in the process of newborn screening. The objective of this process is the early identification of asymptomatic serious diseases in order to prevent or minimise negative neurologic, organic and psychic effects in newborn population. Second, they have been used for the diagnosis of hyperammonemia, a Rare Disease which has to be treated promptly in order to avoid serious neurological consequences and even to avoid death. In both fields, the created FIS have been embedded in different CCG for their testing.



---

## Abreviaturas y acrónimos

- AG** - Algoritmo Genético
- Aide-AQL** - *Aide Archetype Query Editor* - Editor de Consultas de Arquetipos Aide
- Aide-DDE** - *Aide Differential Diagnosis Editor* - Editor de Diagnósticos Diferenciales Aide
- Aide-DS** - *Aide Developer Suite* - Conjunto de Herramientas de Desarrollo Aide
- Aide-FISF** - *Aide Fuzzy Inference System Framework* - Infraestructura para Sistemas de Inferencia Difusa Aide
- Aide-GEP** - *Aide Guideline Execution Platform* - Plataforma de Ejecución de Guías Aide
- Aide-GTP** - *Aide Guideline Technology Platform* - Plataforma Tecnológica de Guías Aide
- Aide-KAT** - *Aide Knowledge Authoring Tool* - Herramienta de Creación de Conocimiento Aide
- Aide-MAT** - *Aide Medical Authoring Tool* - Herramienta de Creación Médica Aide
- Aide-UMLS** - *Aide UMLS Terminology Finder* - Buscador Terminológico de UMLS Aide
- ATL** - *ATL Transformation Language* - Lenguaje de Transformación ATL
- CIM** - *Computational Independent Model* - Modelo independiente de la computación
- CSV** - *Comma-Separated Values* - Valores Separados por Comas
- DD** - Diagnóstico Diferencial
- CMOF** - *Complete MOF* - Servicio de Meta-Objetos Completo
- EIM** - Error Innato del Metabolismo
- EMF** - *Eclipse Modeling Framework* - Infraestructura de Modelado de Eclipse
- EMOF** - *Essential MOF* - Servicio de Meta-Objetos Esencial
- EMP** - *Eclipse Modeling Project* - Proyecto de Modelado de Eclipse
- FLT** - *Fuzzy Logic Toolbox* - Conjunto de Herramientas para Lógica Difusa

**FP** - Función de Pertenencia  
**GC** - Guía Clínica  
**GCI** - Guía Clínica Informatizada  
**GEG** - Grupo de Elaboración de Guías  
**GEM** - *Guideline Elements Model* - Modelo de Elementos de Guía  
**GLIF3** - *Guideline Interchange Format version 3* - Formato de Intercambio de Guías, versión 3  
**GMF** - *Graphical Modeling Framework* - Infraestructura de Modelado Gráfico  
**HCE** - Historia Clínica Electrónica  
**ICI** - Índice de Calidad de la Información  
**IDE** - *Integrated Development Environment* - Entorno de Desarrollo Integrado  
**LD-1** - Lógica Difusa de nivel 1  
**LD-2** - Lógica Difusa de nivel 2  
**LED** - Lenguaje Específico del Dominio  
**M2M** - *Model-to-Model* - Modelo-a-Modelo  
**M2T** - *Model-to-Text* - Modelo-a-Texto  
**MBE** - Medicina Basada en la Evidencia  
**MCD** - Mapa Cognitivo Difuso  
**MDDM** - Métodos de Desarrollo Dirigido por Modelos  
**MDL** - Modelado Difuso Lingüístico  
**MDP** - Modelado Difuso Preciso  
**MOF** - *Meta-Object Facility* - Servicio de Meta-Objetos  
**MOM** - Modelos Ocultos de Markov  
**MRT** - Modelo de Redes de tareas  
**OMG** - *Object Management Group* - Grupo de Gestión de Objetos  
**OMIM** - *Online Mendelian Inheritance in Man* - Herencia Mendeliana del Humano Online  
**OWL** - *Ontology Web Language* - Lenguaje de Ontologías Web  
**PIM** - *Platform Independent Model* - Modelo independiente de la plataforma  
**RNA** - Red Neuronal Artificial  
**SATD** - Sistema de Ayuda a la Toma de Decisiones  
**SID** - Sistema de Inferencia Difusa  
**SIDRA** - Sistema de Inferencia Difusa basado en Redes Adaptativas  
**SPK** - *Software Prozesuen Kudeaketa* - Gestión de Procesos de Software  
**TMF** - *Textual Modeling Framework* - Infraestructura de Modelado Textual  
**UML** - *Unified Modeling Language* - Lenguaje Unificado de Modelado  
**UMLS** - *Unified Medical Language System* - Sistema de Lenguaje Médico Unificado  
**XMI** - *XML Metadata Interchange* - Intercambio de Metadatos en XML

## Parte I

---

### Prolegómenos



La Parte I presenta los **prolegómenos** de la investigación en los que se enmarca la presente tesis doctoral.

En el capítulo 1 se justifica la realización del trabajo de investigación llevado a cabo y se ofrece el contexto en el que se sitúa la investigación realizada.



## Introducción

Ofrecer el mejor cuidado a los pacientes consiste en utilizar el *conocimiento médico más actual* para ofertar un *cuidado común adaptado a las características del paciente* evitando de esta manera pruebas costosas innecesarias (Rosenfeld y Shiffman, 2009).

El **conocimiento más actual** se recoge en la evidencia de los estudios médicos publicados. Por ejemplo, si un paciente viene aquejado de fiebre y los últimos estudios han demostrado que un nuevo medicamento es más efectivo que los que se recetan habitualmente, que le receten el nuevo medicamento indicará que el cuidado médico está actualizado.

El **cuidado común** es la característica que hace que la respuesta ante unos síntomas comunes conlleve un mismo procedimiento sin depender de la arbitrariedad de diferentes criterios médicos. Por ejemplo, si dos pacientes vienen aquejados de un dolor en la rodilla y para ambos se piden radiografías, eso indica que el cuidado es común.

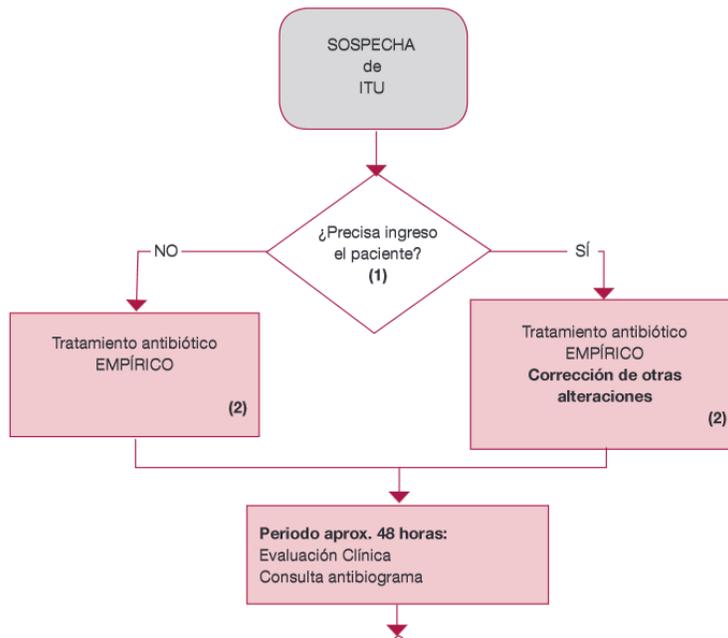
La **adaptación a las características del paciente** consiste en fijar las decisiones del facultativo de acuerdo a las características del paciente. Por ejemplo, dada una dolencia y un paciente alérgico al medicamento más común para tratar esa dolencia, que le receten una alternativa significará que el cuidado ha sido adaptado.

Las Guías Clínicas (GC) recopilan de forma sistemática recomendaciones médicas que sirven de soporte a los profesionales médicos en la toma de decisiones (Field y Lohr, 1990). Las GC son el resultado del trabajo de un Grupo de Elaboración de Guías (GEG), formado por médicos y documentalistas especialistas en una materia. Las GC se actualizan con cierta periodicidad para mantener la actualidad de las recomendaciones que incluyen. En las tareas de elaboración y actualización se utilizan herramientas como MAGIC (Vandvik et al., 2017), que permitan gestionar todo el proceso, o herramientas específicas que ayuden a localizar bibliografía relevante (Iruetaguena et al., 2012).

Para ayudar en la toma de decisiones, las GC califican las recomendaciones de acuerdo a la evidencia que sustentan los estudios médicos. También inclu-

yen otros elementos que facilitan la realización de las tareas médicas, tales como *algoritmos*, *scores* y *Diagnósticos Diferenciales (DD)*.

Los **algoritmos** son representaciones de las tareas a realizar durante algún aspecto del diagnóstico y el tratamiento de la enfermedad. Por ejemplo, en la figura 1.1 se muestra un algoritmo para los pasos iniciales en el diagnóstico y tratamiento de la Infección del Tracto Urinario (ITU) en población pediátrica. El algoritmo que se muestra está formado por cinco pasos y cada forma representa un tipo de acción. Según la representación gráfica utilizada para este algoritmo, los rectángulos con borde redondeado indican una condición de entrada, los rombos o diamantes indican condiciones y los rectángulos acciones. Las acciones se llevan a cabo en el orden establecido por las flechas, de forma que el primer paso es el rectángulo redondeado en la parte superior de la figura y el último paso es el rectángulo mostrado en la parte inferior de la figura.



**Figura 1.1.** Muestra de un algoritmo en la GC para el diagnóstico y tratamiento de la ITU (extraído de (Trapote et al., 2011))

Los **scores**, por su lado, son sistemas de puntuación utilizados para medir la gravedad o intensidad de algunos rasgos de las enfermedades. Estos sistemas asignan puntos a diversos factores de los pacientes y el resultado de la suma de los mismos indica la gravedad. En la figura 1.2 se muestra un *score* utilizado

en las GC sobre el asma. En este *score* los puntos están indicados en las filas de la tabla superior y los factores en las columnas de esa misma tabla. Así, a una *frecuencia respiratoria de 40* para un niño de ocho años le corresponde una puntuación de 2. Si ese mismo niño tiene *sibilancias al final de la espiración (1 punto)* y tiene el uso del *músculo ECM aumentado (2 puntos)* le corresponde una puntuación total de 5 puntos. En la tabla inferior de la figura se muestra la gravedad del paciente en base a la suma de los puntos obtenidos. En la segunda columna se puede ver que a una puntuación de 5 le corresponde una *gravedad moderada*.

**Tabla 7. Pulmonary score (PS) para valoración de la crisis de asma**

Puntuación	Frecuencia respiratoria		Sibilancias	Uso de músculo ECM
	<6 años	≥6 años		
0	<30	<20	No	No
1	31-45	21-35	Final espiración	Incremento leve
2	46-60	36-50	Toda la espiración	Aumentado
3	>60	>50	Inspiración y espiración, sin estetoscopio	Actividad máxima

Se puntúa de 0 a 3 en cada uno de los apartados (mínimo 0, máximo 9).

Gravedad	PS	PEF*	Sat O <sub>2</sub>
Leve	0-3	>80%	>94
Moderada	4-6	60-80%	91-94
Grave	7-9	<60%	<91

\*Tras la administración de una dosis de broncodilatador.  
En caso de discordancia entre PS, PEF y Sat O<sub>2</sub>, se clasificará con el de mayor gravedad.

**Figura 1.2.** Muestra de un *score* de una guía para el tratamiento del asma (extraído de (Rico, 2013))

Por último, los DD relacionan valores de variables clínicas con uno o varios trastornos (o con conjuntos de trastornos que se tratan como uno único en los diagnósticos debido a su similitud) y sirven para facilitar los procedimientos diagnósticos. En la figura 1.3 se muestra un DD para el diagnóstico de sospecha de la etiología más probable causante de hiperamonemia, el cual es parte de una GC para el diagnóstico y tratamiento de la hiperamonemia. En este DD concreto, las variables clínicas se muestran en las columnas y los trastornos en las filas. Los valores de las variables clínicas se muestran en las celdas que relacionan las variables con los trastornos. Así, el trastorno *Atrofia Gyrate (AGy)* tiene un valor de 7,3-7,4 en relación a la variable clínica *Acidosis metabólica*. Los valores de los DD pueden ser tanto numéricos (como el caso expuesto) como lingüísticos. Los valores lingüísticos están más sujetos a interpretaciones subjetivas que a valores concretos objetivos (López-Cuadrado et al., 2015). Por ejemplo, en la relación entre el conjunto de trastornos *Trastornos del ciclo de la urea y/o administración de ácido valproico* y la variable *Amonio* está indicado el valor *Muy elevado*. Sin embargo, en ningún sitio se establece qué significa un valor muy elevado, por lo que éste está sujeto a la valoración personal del profesional médico.

Según (Denekamp y Peleg, 2010) el 12% de las GC contenidas en el repositorio *National Guideline Clearinghouse* de los Estados Unidos (unas 240)

**Tabla 3b. Hiperamonemia. Diagnóstico diferencial.**

	Acidosis metabólica (pH en sangre)	Amonio (N < 50 µmol/L < 90 µg/dl)	Lactato (N < 2,5 mM, < 20 mg/dl)	GOT-GPT (N< 40 UI/L)	CPK (N<190 UI/L)	Ác. úrico (N< 7 mg/dl)	3-OHbutirato > 0,5 mM (hay síntesis de cetónicos)
<b>Si hay Hipoglucemia</b>							
Glucemia venosa < 45 mg/dl							
<b>SACAR INSULINA!!</b>							
Trastornos de cetogénesis	< 6,9	N o E	N o E	E	E	N	< 0,3 mM (bajo)
Oxidación mitocondrial de ácidos grasos	< 7,0	N o E	E	E	E	N o E	Bajo
Glutaricoaciduria tipo II. (MADD)	< 7,0	E	E	E	E	E	Bajo
Defecto múltiple de deshidrogenasas							
Hiperinsulinemia + hiperamonemia	7,3-7,4	< 250 µmol/L	N	N	N	N	Siempre < 0,3
<b>Si no hay Hipoglucemia</b>							
Trastornos del ciclo de la urea y/o administración de ácido valproico	N o < 7,4	Muy E	N o E	E	N	N	Según ayuno
Jarabe de Arce (catabolismo de aa. ramificados; Leu-VaHsoleu)	7,1-7,2	N o leve (< 150µmol/L)	N o E	N	N	N	> 0,5 mM
Acidurias Isoleucérica y 3 metil crotónica (catabolismo de leucina)	< 7,0	200> 700 µmol/L	E	N o E	N	N o E	E (> 1 mM)
Acidemias propiónica y metilmalónica (catabolismo de Met-Treo-VaHsoleu)	< 6,9	Muy E > 700 µmol/L	E	N o E	Puede estar E	N o E	E
Deficiencia múltiple de carboxilasas (MCD)	< 6,9	90-700µmol/L	E	N o E	N	N o en límite alto	E
Atrofia girata (AGy)	7,3-7,4	Puede ser muy E	N	N o E	N	N	Según ayuno
Hiperamonemia-Hiperornitinemia-Homocitrulinuria (HHH)	7,2-7,4	200-350 µmol/L	N	N	N	N	Según ayuno
Defectos de transporte de aminoácidos dibásicos, cistinuria e intolerancia a proteínas con isinuria	pH orina > 7	120 -250 µmol/L	N	N	N	N	Según ayuno

N = normal. E= elevado. µmol/l= micromol/L. mM= milimolar (milimol/L). µg/dl=microgramos/decilitro. Analítica urgente (resultados antes de 2 h) a través de la cual podemos tener una sospecha de la posible etiología. El diagnostico definitivo lo obtendremos con los resultados del aminograma (en plasma/orina), de los ácidos orgánicos (en orina), de la carnitina total y libre (plasma), de las acilcarnitinas (plasma) y de la insulinemia. El 3OH butirato lo podemos determinar en una gota de sangre total en el "OPTIUM EXCEED"

**Figura 1.3.** Diagnóstico Diferencial para la sospecha de etiología más probable causante de hiperamonemia (extraído de (Couce et al., 2009))

comienzan con un DD. Este porcentaje es mayor si se tienen en cuenta aquellas GC que también incluyen DD en su interior.

Las Guías Clínicas Informatizadas (GCI) son el resultado de trasladar la información de las GC a un formato interpretable por un ordenador. Una primera aproximación consiste en expresar la información contenida en la GC en un soporte textual digital (formato PDF, por ejemplo). Otra opción para dotar de mayor versatilidad a una GC es categorizar su información para que se pueda ofrecer un tratamiento automático de los elementos de información que contiene. En esta tesis se utilizan Sistemas de Ayuda a la Toma de Decisiones (SATD) asociados a GCI que modelan los algoritmos y procesos de una GC y se presta especial atención a la representación e interpretación de los DD.

El método para implementar en una GCI los DD incluidos en una GC es mediante Sistemas de Inferencia Difusa (SID), un tipo de sistema de clasificación con capacidad para modelar tanto la incertidumbre de los diagnósticos como la imprecisión de los síntomas (Khatibi y Montazer, 2010). Una de las mayores ventajas de estos sistemas es que el usuario final puede entender fácilmente el razonamiento seguido hasta llegar al resultado.

Debido al gran tamaño de algunos DD, y por consiguiente de los SID resultantes, en ocasiones puede resultar muy difícil diseñar SID de forma manual. En estos casos hasta el menor de los cambios puede ser muy costoso en tér-

minos tanto de tiempo como de esfuerzo. Este problema puede solucionarse utilizando métodos que permitan centrarse más en el diseño del modelo que en la programación manual, como por ejemplo, los Métodos de Desarrollo Dirigido por Modelos (MDDM).

Los SID se pueden enriquecer mediante sistemas de aprendizaje automático que afinen sus prestaciones (Pandey y Mishra, 2009). Por ejemplo, un SID puede recoger información de casos reales para afinar su funcionamiento de forma que el resultado que ofrezca se ajuste más a la realidad. Una de las soluciones más comunes consiste en utilizar Algoritmos Genéticos (AG) como sistema de aprendizaje automático. Los AG son métodos de búsqueda basados en la evolución por selección natural.

La hipótesis de partida de esta tesis es que los SID son un método adecuado para informatizar los DD de una GC e individualizar de esta forma los resultados que se obtienen y que los MDDM permiten la sistematización del desarrollo de los SID y su posterior integración dentro de una GCI, consiguiendo así una reducción de tiempo y esfuerzo.

Los objetivos de esta tesis son **1) la definición de recursos basados en MDDM para crear SID a partir de DD; 2) el diseño de herramientas para la construcción de DD y SID; 3) el uso de técnicas de aprendizaje automático para la mejora de los SID creados.**

Los recursos de MDDM que se definan deberán ser suficientemente flexibles y extensibles como para que se puedan modificar fácilmente. Por lo tanto, se definirá un *metamodelo de DD* independiente de la computación. Esto añadirá un nivel de abstracción mayor y permitirá que los profesionales médicos puedan involucrarse en el proceso de definición de los DD. También se definirá un *metamodelo de SID* independiente de la plataforma. De esta forma, los modelos de SID resultantes de este metamodelo se podrán utilizar en soluciones concretas de distinto tipo. Esto añade mayor libertad a los diseñadores de los modelos de SID, ya que no tendrán que estar pendientes de diseñar un modelo teniendo en cuenta las restricciones de una plataforma o lenguaje concreto. También se definirá un *metamodelo para representar los datos de entrada del SID*. Al fin y al cabo, un SID se puede llegar a utilizar en soluciones diferentes (una aplicación web, una aplicación de escritorio, etc.) y el proceso de introducir los datos de entrada debe ser totalmente transparente para el propio SID. Este metamodelo permitirá crear los recursos necesarios para que el metamodelo de SID siempre obtenga los datos de entrada en el formato que prevé. De la misma forma, también se definirá un *metamodelo para la representación de los resultados*. Al igual que ocurre con los datos de entrada, los formatos de salida son innumerables, por lo que disponer de un metamodelo para representar la capa intermedia entre los resultados del SID y el formato objetivo permitirá ofrecer los resultados como la solución final los requiera. Los metamodelos de datos de entrada y de resultados facilitarán que el metamodelo de SID sea independiente de la plataforma. Una de las grandes ventajas de los MDDM es el uso de transformaciones automáticas para generar automáticamente distintos recursos. En esta tesis se definirán

unas *transformaciones para generar modelos de SID a partir de modelos de DD*. De esta forma, todo el conocimiento incluido en los modelos de DD se trasladará automáticamente a los SID, ahorrando de esta manera esfuerzo a los diseñadores del SID y minimizando los errores innatos de la programación manual. También se definirán unas *transformaciones para generar SID ejecutables a partir de modelos de SID*. Estas transformaciones permitirán generar distintos tipos de SID ejecutables a partir del mismo modelo de SID, lo que facilitará su utilización en distintas soluciones.

En cuanto a las herramientas para la creación de DD y SID, éstas deberán ofrecer un contexto común y englobar todos los recursos de MDDM creados en el objetivo anterior. También deberán, en la medida de lo posible, agilizar el proceso de desarrollo de GCI. Con todo esto, por un lado, se creará un *editor de modelos de DD* para que los profesionales médicos puedan trasladar el conocimiento de los DD a modelos formales. Por otro lado, se creará una *editor de SID* para poder crear modelos de SID. El hecho de que sean dos herramientas (cada una para un conjunto de usuarios diferente) agiliza el proceso de desarrollo y encaja correctamente con el modelo de desarrollo en dos niveles de GCI (Buenestado et al., 2011). Por su lado, el editor de SID deberá contener funcionalidades para *transformar modelos de DD en modelos de SID*, *generar SID ejecutables* a partir de modelos de SID y *simular la ejecución* de los modelos de SID definidos.

Por último, las técnicas de aprendizaje automático que se utilicen deberán *basarse en métodos contrastados* y deberán *ser flexibles*. Por ello, no se incluirá un sistema de aprendizaje cerrado y no modificable, sino que *será parametrizable* de forma que los usuarios puedan adecuar el entrenamiento de los modelos de SID a sus necesidades.

## 1.1. Contexto de trabajo

La presente tesis se incluye dentro del trabajo del grupo de investigación **Erabaki** de la **Universidad del País Vasco / Euskal Herriko Unibertsitatea (UPV/EHU)**. Erabaki es un grupo multidisciplinar formado por 17 investigadores, 12 de los cuales son doctores. De esos 12 doctores 7 son informáticos que pertenecen a la Facultad de Informática y a la Escuela de Ingeniería de Bilbao, 4 son médicos pertenecientes a la Facultad de Medicina y adscritos a los servicios sanitarios de pediatría de los Hospitales Universitarios de Basurto, Cruces y Donostia y 1 es un investigador asociado de la universidad de Western Australia.

Las líneas principales de investigación del grupo Erabaki son la *minería de textos y el aprendizaje a partir de textos no estructurados* y los *SATD*, sobre todo en el área de la medicina.

En la línea de la **minería de textos y el aprendizaje a partir de textos no estructurados**, el grupo creó un motor llamado *Pimiento* para realizar dicha tarea (García y Calvo, 2006). Este motor, se integró junto con

un gestor de corpuses llamado *Lenteja* en una plataforma llamada *Plato*. Esta tecnología se utilizó en un problema de detección de plagios utilizando la herramienta Beagle (García y Pikatza, 2006). En la actualidad, esta línea de investigación se centra en la búsqueda de evidencia y bibliografía de manera automática para su utilización en nuevas versiones de GC (Iruetaguena et al., 2012).

En la línea de los **SATD**, el primer SATD que se construyó dentro del grupo se utilizó para ayudar en los cuidados médicos de la cetoacidosis diabética. Este SATD utilizaba un motor de inferencia construido sobre LISP (Reilly, 2003) que se llamaba MAMUR (Pikatza, 1992).

Posteriormente se creó un motor de inferencia con mejores prestaciones que permitía no sólo realizar cálculos y evaluar expresiones en lenguaje CLIPS (Savely et al., 2017) y FuzzyCLIPS (NRCC, 2015), si no que también permitía definir interfaces y canales de comunicación (Arrieta et al., 1995). Este nuevo motor de inferencia recibió el nombre de EHSIS y se utilizó en un SATD para la atención y el tratamiento de la Diabetes Mellitus de tipo I (Pikatza et al., 1998).

En lo que a SATD basados en GCI se refiere, el grupo Erabaki creó la herramienta MELISIS (Pikatza et al., 2001), un SATD para la ejecución de GCI vía web. Las GCI se definían manualmente y posteriormente se cargaban en el SATD. Utilizando esta tecnología se desarrolló el sistema *Arnasa*, un SATD para el diagnóstico y tratamiento del asma (Sobrado et al., 2004).

Para mejorar la parte de la edición de las GCI, se desarrolló Teseo, una plataforma de edición y ejecución de GCI que utilizaba el lenguaje de representación de GCI Lagre (Flórez et al., 2005). Teseo no se utilizó únicamente en el campo de la medicina, ya que también se utilizó en la gestión de procesos de software en el sistema SPK (*Software Prozesuen Kudeaketa*) (Larburu et al., 2003).

En 2010 el grupo desarrolló la tecnología GuidMex, una plataforma de edición y ejecución de GCI independiente de la plataforma que utilizaba GCI expresadas mediante arquetipos (Buenestado et al., 2010). El motor de inferencia utilizado en esta plataforma era Mairi, una versión de EHSIS accesible mediante servicios web.

En la actualidad el grupo trabaja sobre *Aide Guideline Technology Platform*, una solución integral para la edición y ejecución de GCI independientes de la plataforma y que utiliza el formalismo de representación de GCI *Aide* (Buenestado, 2014).

## 1.2. Organización de la tesis

La presente tesis doctoral está dividida en seis partes. La parte I explica los prolegómenos del trabajo realizado. En este primer capítulo se realiza una introducción a la investigación realizada y se presenta el contexto del trabajo en que ésta se engloba.

En la parte II se presenta la base teórica en que se sustenta la investigación realizada. En el capítulo 2 se explican detalladamente los MDDM centrándose en los conceptos y las herramientas existentes, mientras que el capítulo 3 ofrece una explicación detallada sobre las GC y las GCI, así como sobre el formalismo de representación de GCI *Aide*. El capítulo 4 revisa la plataforma *Aide Guideline Technology Platform* creada por el grupo de investigación. En el capítulo 5, por su lado, se realiza una introducción a la lógica difusa y se presentan de forma detallada los SID. Por último, el capítulo 6 presenta el uso de los AG para el entrenamiento de SID.

La parte III expone el estado del arte previo al trabajo presentado. En el capítulo 7 se realiza una revisión de los sistemas de clasificación utilizados en formalismos de representación de GCI, mientras que en el capítulo 8 se revisan varias soluciones médicas en las que se utilizan SID como componentes principales. Finalmente, en el capítulo 9 se hace una revisión de los *frameworks* existentes para el desarrollo de SID.

La parte IV presenta el trabajo realizado para conseguir los objetivos de la tesis planteados. El capítulo 10 expone el uso de los MDDM para la representación de DD y SID, así como para realizar transformaciones automáticas entre esas representaciones. El capítulo 11, por su lado, presenta las herramientas construidas para definir DD y SID. Finalmente, el capítulo 12 presenta las pruebas realizadas para demostrar la validez de la investigación.

En la parte V se exponen las conclusiones de la tesis. En el capítulo 13 se presentan las aportaciones de la investigación. En el capítulo 14 se presentan las líneas de trabajo futuras y en el capítulo 15 se enumeran las publicaciones generadas durante la investigación. En esta parte también se incluyen las referencias bibliográficas citadas a lo largo de todo el documento.

Finalmente, en la parte VI se incluyen 4 anexos. El anexo A muestra la lista de trastornos evaluados en el DD de cribado neonatal. El anexo B muestra la lista de las variables clínicas tenidas en cuenta en el DD de cribado neonatal. El anexo C, por su lado, presenta el DD de cribado neonatal. Finalmente, el anexo D ofrece los resultados completos sobre las pruebas realizadas con un SID creado a partir del DD de cribado neonatal.

### 1.3. Guía de lectura

Ya que en cada capítulo se citan conceptos explicados en capítulos anteriores, es recomendable leer el presente documento de forma secuencial desde el principio. Sin embargo, dependiendo del grado de conocimiento del lector, éste puede obviar algunos capítulos de la base teórica.

El lector que esté familiarizado con los conceptos de las GC y las GCI no necesita leerse completamente el capítulo 3. Basta con que se lea el apartado 3.3, donde se explica el formalismo de representación de GCI *Aide*. En el caso de que no esté familiarizado con los conceptos de GC y GCI, debería leerse los apartados 3.1 y 3.2. Independientemente de estos apartados, el lector con

perfil médico necesita leerse el capítulo 2, donde se presentan los MDDM, el capítulo 4, donde se presenta la plataforma de edición y ejecución *Aide Guideline Technology Platform*, el capítulo 5, donde se introduce la lógica difusa y se explican en profundidad los SID y el capítulo 6, donde se presentan los AG en su vertiente especializada para el entrenamiento de SID.

En cuanto al lector con perfil informático, si está familiarizado con conceptos de los MDDM, entonces no necesita leer el capítulo correspondiente (capítulo 2). Si su campo está más relacionado con la computación y la matemática quizás no tenga que leer los capítulos sobre los SID y el entrenamiento de SID mediante AG (capítulos 5 y 6). Concretamente, si está familiarizado con la lógica difusa puede obviar el apartado 5.1, si conoce en profundidad la estructura y el razonamiento de los SID puede saltarse los apartados 5.2 y 5.3 y si comprende el funcionamiento de los AG para el entrenamiento de SID puede obviar el capítulo 6. Por último, si el lector conoce algunos de los trabajos previos realizados por el grupo Erabaki, tales como el formalismo de representación de GCI *Aide* y la plataforma de edición y ejecución *Aide Guideline Technology Platform*, puede saltarse el apartado 3.3 y el capítulo 4, donde se presentan el formalismo *Aide* y la plataforma de edición y ejecución, respectivamente.



## Parte II

---

### Base Teórica



La Parte II presenta los **fundamentos teóricos** en los que se basan los conceptos utilizados para llevar a cabo la presente tesis.

Así, en el capítulo 2 se presentan los métodos y recursos que ofrecen los Métodos de Desarrollo Dirigido por Modelos para construir sistemas de software utilizando una visión industrial.

El capítulo 3, por su lado, realiza una introducción a los formalismos de representación de Guías Clínicas Informatizadas, haciendo un especial hincapié en el formalismo de representación Aide.

El capítulo 4 se centra en la plataforma de edición y ejecución de Guías Clínicas Informatizadas *Aide Guideline Technology Platform*, la cual usa el formalismo de representación de Guías Clínicas Informatizadas Aide.

El capítulo 5 introduce la lógica difusa, realiza una taxonomía sobre los distintos sistemas de clasificación existentes y presenta en profundidad el funcionamiento de los Sistemas de Inferencia Difusa.

Por último, el capítulo 6 presenta la aplicación de los Algoritmos Genéticos para el entrenamiento de Sistemas de Inferencia Difusa.



## Métodos de Desarrollo Dirigido por Modelos

En este capítulo se presentan los Métodos de Desarrollo Dirigido por Modelos (MDDM), un conjunto de técnicas y recursos que permiten afrontar el desarrollo de software desde un punto de vista industrial.

En primer lugar se realiza una introducción a los MDDM. Posteriormente se explica lo que son los metamodelos y el metamodelado. De la misma forma, también se introducen las transformaciones que es posible realizar sobre los modelos. Finalmente se presentan las distintas herramientas existentes para el uso de los MDDM.

### 2.1. Introducción a los Métodos de Desarrollo Dirigido por Modelos

Los MDDM constituyen un paradigma de desarrollo de software que se centra en la definición y utilización de modelos abstractos, más cercanos a los conceptos de dominio en lugar de conceptos de computación (Selic, 2003). El objetivo de los MDDM es minimizar la programación manual de código, evitando de esta forma el componente artesanal del desarrollo de software tradicional.

Para lograr este objetivo los MDDM utilizan *lenguajes de modelado* y proveen de herramientas para la *generación automática de código* a partir de los modelos definidos.

Los **lenguajes de modelado** se clasifican como *lenguajes genéricos* y *Lenguajes Específicos del Dominio*. Los **lenguajes genéricos** de modelado se utilizan para expresar información, conocimiento o sistemas de forma estructurada y no tienen un ámbito de uso específico. *Unified Modeling Language (UML)* es un ejemplo de lenguaje genérico de modelado. Los **Lenguajes Específicos del Dominio** o **LED** son lenguajes que se centran en los aspectos del dominio del problema, por lo que permiten definir modelos con distintos niveles de abstracción. La principal diferencia entre los lenguajes de modelado genérico y los LED es que los LED se centran en el dominio de la solución y

ofrecen al desarrollador unas especificaciones más fáciles de entender (Kelly y Tolvanen, 2008). De esta forma, se requiere menos esfuerzo durante el proceso de modelado y se pueden desarrollar aplicaciones del dominio de forma más rápida.

La **generación automática de código**, por su lado, consiste en crear automáticamente secciones de código a partir de las características de los modelos. Aunque esta visión industrial sea una práctica bastante novedosa en el ámbito de la informática, viene siendo una práctica habitual en otras ingenierías, como la mecánica. El uso de los MDDM ofrece una serie de ventajas frente al desarrollo tradicional de software (Hutchinson et al., 2011):

- La velocidad de desarrollo es mayor, gracias a la automatización proveniente del uso de herramientas para la generación automática de código.
- Se incrementa la calidad del software desarrollado gracias al uso de LED y de transformaciones automáticas.
- La mantenibilidad de las soluciones desarrolladas es mayor ya que los cambios se producen a nivel de modelo.
- Se ofrece un mayor nivel de abstracción, lo que permite tratar los problemas desde un punto de vista más conceptual. Esto conlleva que sea posible centrarse en la resolución del problema dejando de lado los aspectos computacionales.
- Se contribuye a mejorar la interoperabilidad y portabilidad. Definiendo distintas transformaciones es posible obtener automáticamente distintas soluciones a partir de un único modelo.

Sin embargo, la utilización de los MDDM conlleva una serie de aspectos que se deben considerar (Stahl et al., 2006):

- Generalmente los detalles de implementación quedan fuera del modelo ya que no están directamente relacionados con el dominio del problema a resolver.
- Normalmente no es posible generar todo el código automáticamente. Por lo tanto, hay que prestar especial atención a la integración del código generado automáticamente y el código creado de forma manual, a fin de que no se produzcan inconsistencias entre ambos.
- Es esencial que el modelo y el código generado estén sincronizados, los cambios realizados en uno deben verse reflejados en el otro y viceversa.

## 2.2. Metamodelado

Dentro de los MDDM, los LED se definen mediante *metamodelos*, conjuntos de elementos que declaran o definen conceptos sobre modelado (Clark et al., 2008). Esto es, un metamodelo describe la posible estructura de cualquier modelo, así como las relaciones entre los elementos, las restricciones de los mismos y las reglas que se pueden aplicar.

Un metamodelo se define especificando tanto su *sintaxis abstracta* como su *semántica abstracta*. Dentro de la **sintaxis abstracta** se incluyen los diferentes elementos necesarios en el lenguaje, mientras que en la **semántica abstracta** se definen las relaciones y restricciones entre dichos elementos. Hay que tener en cuenta que un metamodelo no especifica la sintaxis concreta de los modelos, esto es, no determina qué elementos gráficos o textuales se utilizan para representar dichos modelos. Por ejemplo, en la figura 2.1 se muestra un metamodelo para representar modelos de dominio de sistemas de información, los cuales están formados por entidades, atributos y relaciones con otras entidades. Todas las metaclasses tienen el atributo *nombre*. Además, la metaclass *Atributo* permite guardar su tipo y la metaclass *ReferenciaEntidad* posibilita indicar si la referencia es múltiple o no. Sin embargo, en ningún sitio se especifica la forma que deben tener las entidades del modelo de dominio, por ejemplo, si en el caso de utilizar una representación gráfica, las entidades deban tener forma rectangular o elíptica. De hecho, no está especificado ni que obligatoriamente el modelo de dominio de un sistema de información deba representarse gráficamente. Por lo tanto, una definición textual es igualmente válida.

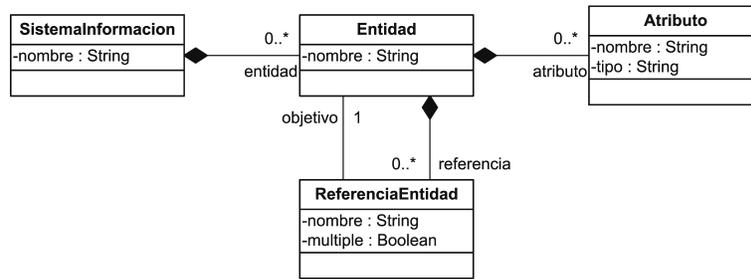
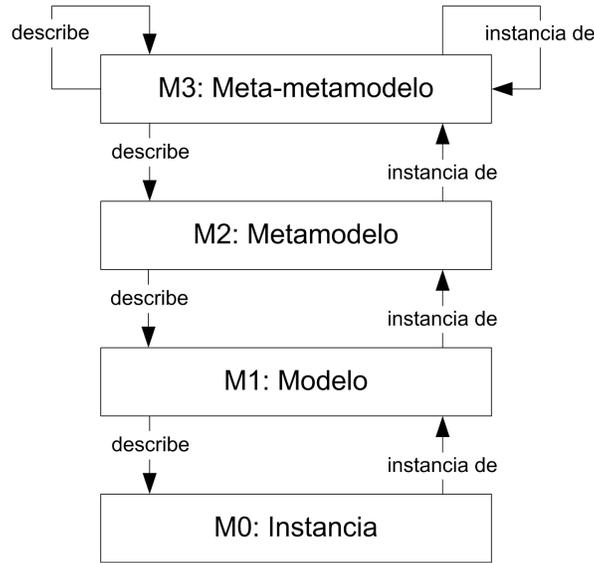


Figura 2.1. Metamodelo de un modelo de dominio de un sistema de información

Para definir un metamodelo es necesario disponer de un lenguaje de metamodelado. Este lenguaje es a su vez un LED específico para la definición de metamodelos y por lo tanto, se puede definir mediante un meta-metamodelo. De la misma forma, ese meta-metamodelo se definiría utilizando un lenguaje de meta-metamodelado, el cual a su vez se podría definir utilizando un meta-meta-metamodelo. En teoría esta sucesión no tiene fin. Sin embargo, el *Object Management Group (OMG)* establece una arquitectura de cuatro niveles de abstracción (ver figura 2.2) en la que el nivel superior se establece en el meta-metamodelo. El lenguaje de nivel más alto propuesto por el OMG (el correspondiente al nivel M3: Meta-metamodelo de la figura 2.2) se denomina *Meta-Object Facility (MOF)*, el cual se considera “un medio universal para definir lenguajes de modelado” (OMG, 2017a).

El OMG ha definido dos variantes del MOF: *Essential MOF (EMOF)* y *Complete MOF (CMOF)*. **EMOF** es un subconjunto de MOF que provee al



**Figura 2.2.** Arquitectura de modelado según el OMG (adaptada de (OMG, 2017b))

desarrollador de las capacidades básicas de MOF. En este lenguaje se ofrecen los elementos mínimos necesarios para realizar modelado orientado a objetos: clases, atributos, operaciones, herencia y paquetes. **CMOF**, por su lado, es una extensión de EMOF y permite definir otros tipos de metamodelos.

Los metamodelos MOF son independientes de la plataforma. Sin embargo, existen correspondencias (*mappings*) de MOF a formatos de serialización concretos, de forma que se permite transformar la sintaxis abstracta de los metamodelos a representaciones concretas. Para este proceso el OMG ofrece el lenguaje *XML Metadata Interchange (XMI)* (OMG, 2017c), un lenguaje de intercambio de metadatos independiente de cualquier plataforma. En general, el uso de distintas correspondencias proporciona interoperabilidad entre distintas herramientas y formatos.

Disponiendo de un LED (formalizado mediante un metamodelo) es posible crear modelos que se ajusten a las especificaciones de dicho metamodelo. En el extracto 2.1 se muestra un modelo que se ajusta al metamodelo de la figura 2.1 expresado en lenguaje XMI. Este modelo está formado por un sistema de información llamado *Datos*, el cual a su vez contiene dos entidades llamadas *Persona* y *Coche*. La primera de ellas tiene tres atributos para guardar información sobre el nombre, la edad y el DNI de una persona, mientras que la segunda tiene dos atributos para guardar la matrícula y el color de un coche. La entidad *Persona* también tiene una relación múltiple con la entidad *Coche*, a fin de poder representar que una persona puede tener más de un coche.

```

<?xml version="1.0" encoding="UTF-8"?>
<ejemploMetamodelo:SistemaInformacion

```

```

xmi:version="2.0"
xmlns:xmi="http://www.omg.org/XMI"
xmlns:ejemploMetamodelo=
  "http://www.ejemplo.org/ejemploMetamodelo"
nombre="Datos">
<entidad nombre="Persona">
  <atributo nombre="nombre" tipo="String"/>
  <atributo nombre="edad" tipo="int"/>
  <atributo nombre="dni" tipo="String"/>
  <referencia nombre="coches" multiple="true"
    objetivo="//@entidad.1"/>
</entidad>
<entidad nombre="Coche">
  <atributo nombre="matricula" tipo="String"/>
  <atributo nombre="color" tipo="String"/>
</entidad>
</ejemploMetamodelo:SistemaInformacion>

```

**Extracto 2.1.** Serialización de un modelo en lenguaje XMI

Posteriormente, y dependiendo de la intención final de la solución modelada, se pueden utilizar las capacidades de generación automática de los MDDM para realizar alguna clase de transformación (Mens y Gorp, 2005). Por lo general, todas estas transformaciones se agrupan en dos tipos principales: *Modelo-a-Modelo* (*Model-to-Model*, *M2M*) y *Modelo-a-Texto* (*Model-to-Text*, *M2T*). Las transformaciones **M2M** se utilizan para transformar la información contenida en un modelo a otro modelo distinto y pueden ser endógenas o exógenas, dependiendo de si la transformación se realiza entre dos modelos pertenecientes al mismo metamodelo o a metamodelos diferentes. En lo que a las transformaciones **M2T** se refiere, a partir de los modelos se pueden conseguir artefactos textuales que pueden ir desde código en un lenguaje de programación hasta documentos serializables en un formato concreto, como XML o SQL. Por ejemplo, mediante transformaciones M2T a partir del modelo del extracto 2.1 se pueden conseguir las clases Java mostradas en el extracto 2.2.

```

public class Persona {
  private String nombre;
  public void setNombre(String value) {
    this.nombre = value;
  }
  public String getNombre() {
    return this.nombre;
  }
}

```

```

    private int edad;
    public void setEdad(int value) {
        this.edad = value;
    }
    public int getEdad() {
        return this.edad;
    }

    private String dni;
    public void setDni(String value) {
        this.dni = value;
    }
    public String getDni() {
        return this.dni;
    }

    private ArrayList<Coche> setCoches;
    public void setCoches(ArrayList<Coche> value) {
        this.coches = value;
    }
    public ArrayList<Coche> getCoches() {
        return this.coches;
    }
}

public class Coche {
    private String matricula;
    public void setMatricula(String value) {
        this.matricula = value;
    }
    public String getMatricula() {
        return this.matricula;
    }

    private String color;
    public void setColor(String value) {
        this.color = value;
    }
    public String getColor() {
        return this.color;
    }
}

```

**Extracto 2.2.** Resultado generado automáticamente realizando transformaciones M2T sobre el modelo del extracto 2.1

### 2.3. Herramientas para los Métodos de Desarrollo Dirigido por Modelos

En cuanto a las herramientas de MDDM, existe una gran variedad. Por un lado están aquéllas que *utilizan lenguajes de modelado genéricos* y aquéllas que *utilizan LED*.

Entre las **soluciones que utilizan lenguajes de modelado genéricos** destaca IBM Rational (IBM, 2017). Esta herramienta utiliza UML como lenguaje de modelado. De la misma forma, también es posible encontrar múltiples soluciones que tomando como entrada modelos definidos mediante UML permiten realizar tareas de validación, transformación y generación automática.

En cuanto a las **herramientas que utilizan LED**, aunque algunas de ellas se basan en el MOF, la mayoría utilizan LED y metamodelos propios. Como ejemplos están las herramientas *MetaEdit+* de MetaCase (MetaCase, 2017), *DSL Tools* de Microsoft (Microsoft, 2017) y el *Eclipse Modeling Project (EMP)* de la iniciativa Eclipse (Gronback, 2009). **MetaEdit+** está compuesto por dos herramientas de nombre *MetaEdit+ Workbench* y *MetaEdit+ Modeler*. Mientras que la primera de ellas se utiliza para definir LED, la segunda permite utilizar esos LED para definir modelos y realizar transformaciones sobre los mismos. En cuanto a la herramienta **DSL Tools**, está incluida dentro del entorno de desarrollo de software *Visual Studio* de Microsoft. Esta herramienta ofrece funcionalidades para la definición de LED, la validación de los LED definidos, la utilización de los LED para crear modelos y la transformación automática de los modelos.

Por su parte, el **EMP** constituye uno de los proyectos más relevantes para difundir y promover el uso de los MDDM. El EMP engloba una serie de proyectos en diferentes líneas: definición de sintaxis abstractas y concretas, transformaciones, soporte a estándares y modelado específico del dominio. El proyecto *Eclipse Modeling Framework (EMF)* es el núcleo del EMP. Concretamente es el encargado de proporcionar las funcionalidades necesarias para poder especificar un LED mediante un metamodelo. El lenguaje utilizado para definir los metamodelos se llama Ecore y mantiene una serie de características comunes con EMOF, aunque no es completamente compatible con el estándar del OMG. A partir de un metamodelo Ecore, EMF proporciona herramientas para poder generar un editor básico de modelos que se ajusten al metamodelo definido, así como funcionalidades para la validación y generación automática. El editor básico especifica una sintaxis concreta para los modelos de forma gráfica siguiendo una estructura de árbol. Sin embargo, ésta puede resultar bastante limitada para la mayoría de los LED.

Es por ello que dentro del EMP también existen otros proyectos para la definición de sintaxis concretas: *Modeling Framework (GMF)* y *Textual Modeling Framework (TMF)*. **GMF** proporciona el soporte necesario para poder desarrollar editores gráficos para la tarea de modelado. Su objetivo es permitir el desarrollo de editores donde los elementos de los modelos definidos

se correspondan con figuras de un diagrama. De esta forma, se puede acercar el proceso de definición del modelo al dominio del problema. En cuanto a **TMF**, este entorno de desarrollo permite la definición de sintaxis concretas de forma textual para aquellos dominios para los cuales una sintaxis gráfica no sea la más adecuada.

Los proyectos para la generación automática incluidos en el EMP dan soporte tanto a transformaciones M2M como M2T. Para las transformaciones M2M, la solución más extendida es la que utiliza el lenguaje de transformación ATL Transformation Language (ATL, 2017), para el cual también ofrecen un entorno de desarrollo. En cuanto a las transformaciones M2T, todas las soluciones que ofrece EMP se basan en el uso de plantillas para la generación de artefactos textuales. Entre las soluciones disponibles destacan Acceleo (Obeo, 2017), una implementación del *MOF Model to Text Language* de OMG y Xpand (Xpand, 2017), un lenguaje basado en plantillas para la generación de artefactos textuales a partir de modelos basados en Ecore. En el extracto 2.3 se muestra la plantilla Xpand utilizada para generar el código mostrado en el extracto 2.2.

```
«DEFINE Root FOR ejemploMetamodelo::SistemaInformacion»
  «EXPAND BloqueEntidad FOREACH entidad»
«ENDDEFINE»

«DEFINE BloqueEntidad FOR ejemploMetamodelo::Entidad»
  «FILE nombre + ".java"»
  public class «nombre» {
    «FOREACH atributo AS a»
      private «a.tipo» «a.nombre»;
    «ENDFOREACH»

    «FOREACH referencia AS r»
      «IF r.multiple»
        private ArrayList<«r.objetivo.nombre»> «r.nombre»;
      «ELSE»
        private «r.objetivo.nombre» «r.nombre»;
      «ENDIF»
    «ENDFOREACH»
  }
«ENDFILE»
«ENDDEFINE»
```

**Extracto 2.3.** Ejemplo de una plantilla en lenguaje Xpand

## Guías Clínicas y Guías Clínicas Informatizadas

En este capítulo se describen tanto las Guías Clínicas (GC) como la versión en formato interpretable por un computador de éstas, esto es, las Guías Clínicas Informatizadas (GCI).

En primer lugar se explica lo que son las GC y los elementos que las componen, para posteriormente describir los distintos tipos de GC existentes e indicar tanto la motivación de la existencia de las GC como sus limitaciones. En segundo lugar se introduce el concepto de GCI como posible solución a las limitaciones de las GC y se presentan los distintos enfoques que es posible adoptar para la informatización de las GC. Finalmente se presenta brevemente el formalismo de representación de GCI Aide.

### 3.1. Guías Clínicas

Formalmente una GC se define como “*un conjunto de declaraciones desarrolladas de manera sistemática para ayudar a los médicos y a los pacientes en la toma de decisiones sobre cuidados sanitarios en circunstancias clínicas específicas*” (Field y Lohr, 1990). Recientes actualizaciones de la definición han añadido que “[las guías clínicas] *se basan en una revisión sistemática de evidencia y una valoración de los beneficios y perjuicios de las distintas opciones alternativas del cuidado sanitario*” (Graham et al., 2011).

Las GC habitualmente contienen una serie de elementos comunes (Shankar et al., 2001):

- Intenciones que han motivado la elaboración de la GC.
- Antecedentes de la GC.
- Criterios de elegibilidad. Esto es, a qué conjunto de pacientes está dirigida la GC.
- Conjunto de recomendaciones que pueden abarcar todas las acciones que deben realizar los médicos respecto a los pacientes. Por ejemplo, realizar una prueba diagnóstica o administrar un medicamento.

- Grafos o algoritmos donde se resumen de forma visual las acciones que deben llevar a cabo los médicos.
- Tablas para la realización de distintos diagnósticos, ya sean Diagnósticos Diferenciales (DD) o de severidad de una enfermedad. Concretamente, los DD son procedimientos médicos utilizados para identificar un trastorno concreto (o un conjunto de trastornos) a partir de una serie de variables clínicas. Estas variables pueden ser síntomas y/o resultados de análisis de laboratorio. Los DD normalmente se expresan por medio de una tabla, como la que se puede ver en la figura 3.1. En esta tabla las columnas muestran las variables clínicas a tener en cuenta y las filas reflejan los posibles trastornos (o conjuntos de trastornos).

Diagnóstico diferencial de la etiología más probable				
	Gasometría (pH sangre)	Amonio	Lactato	GOT-GPT
Trastornos de la cetogénesis	Muy bajo	Normal o discretamente elevado	Normal o discretamente elevado	Elevado
Trastornos de la beta oxidación de los ácidos grasos	Bajo	Normal o discretamente elevado	Elevado	Elevado
Aciduria glutárica tipo 2	Bajo	Elevado	Elevado	Elevado
Hiperinsulinemia + Hiperamonemia	Normal o discretamente elevado	Normal o discretamente elevado	Normal	Normal
Trastornos del ciclo de la urea y/o Intoxicación por administración de ácido valproico	Normal o discretamente elevado	Muy elevado o Crítico	Normal o discretamente elevado	Elevado
Síndrome de Jarabe de Arce	Normal	Normal o discretamente elevado	Normal o discretamente elevado	Normal
Acidemia isovalérica y/o 3-metilcrotonica	Bajo	Elevado o Muy elevado	Elevado	Normal o discretamente elevado

**Figura 3.1.** Muestra de una tabla de DD para la identificación de la etiología más probable en niños con hiperamonemia (extraído y adaptado de la página 33 de (Couce et al., 2009))

Dentro de la denominación genérica de GC se pueden diferenciar distintos tipos de documentos del ámbito clínico, tales como las *Guías de Práctica Clínica*, los *Protocolos de Actuación* y las *Vías Clínicas*. Las **Guías de Práctica Clínica** proporcionan recomendaciones sobre las mejores prácticas en un dominio concreto, pero sin proporcionar detalles sobre la implementación. Los **Protocolos de Actuación**, por su lado, ofrecen un enfoque local sobre una guía e incluyen detalles explícitos de la implementación. Por último, las **Vías Clínicas** son documentos que incluyen acciones recomendadas por más de un protocolo o guía. Aunque cada uno de estos documentos tiene una forma y un alcance distinto (Gooch y Roudsari, 2011), en ocasiones se tratan de forma indistinta (Fox et al., 2006). A lo largo de todo el documento de la tesis se utilizará la denominación GC para referirse al concepto que engloba todos los tipos de guías clínicas.

Las GC constituyen un instrumento fundamental para la difusión de la *medicina basada en la evidencia* (MBE). La MBE utiliza estimaciones en términos de beneficios y riesgos derivados de investigaciones de alta calidad sobre muestras poblacionales (Greenhalgh, 2010) y mediante su uso se pretende contribuir a la estandarización de la práctica clínica, disminuyendo la

variabilidad de los diagnósticos y aumentando la efectividad de las terapias. De la misma forma, la comunidad científica intenta mediante la formalización de las GC hacer frente a la necesidad de estandarizar las decisiones médicas de cara a la práctica (Morris, 2000). Sin embargo, la necesidad de estandarización contrasta con la de ofrecer un tratamiento individualizado y específico al paciente.

### 3.2. Guías Clínicas Informatizadas

Las GCI son el resultado de trasladar el conocimiento de las GC a un formato interpretable por un ordenador. Por lo tanto, mantienen las mismas características de las GC, esto es, son explícitas y detalladas. Aparte de esto las GCI están guiadas por datos del paciente, por lo que permiten, por un lado, la estandarización de los cuidados, y por otro lado, la individualización de los cuidados sanitarios del paciente (Latoszek-Berendsen et al., 2010).

Existen varios enfoques a la hora de informatizar una GC: *creación de documentos digitales*, *creación de documentos web*, *informatización centrada en el documento* e *informatización centrada en el modelo*. De todas formas, aunque los resultados de los enfoques relativos a la creación de documentos digitales y la creación de documentos web utilicen ordenadores para la difusión de las GCI, no ofrecen un verdadero soporte a la decisión. En estos dos casos la guías resultantes no están guiadas por los datos de los pacientes, por lo que la individualización del tratamiento no se logra.

La **creación de documentos digitales** es el proceso de informatización más sencillo ya que consiste únicamente en trasladar el texto de una GC a un documento digital (formato PDF, por ejemplo). En la actualidad existen múltiples repositorio donde poder acceder a este tipo de documentos. Ejemplo de esto son los casos de la *National Guideline Clearinghouse* (AHRQ, 2017) en Estados Unidos, el *National Institute for Health and Clinical Excellence* (NICE, 2017) en el Reino Unido y *GuiaSalud* (GuiaSalud, 2017) en España. En estos repositorios es habitual disponer de herramientas para poder trabajar con las GC allí publicadas.

La **creación de documentos web**, por su lado, consiste en disponer los textos de la GC en un entorno web de forma que mediante enlaces hipertextuales se pueda navegar entre las distintas secciones que componen la guía, siendo este sistema más cómodo que tener que pasar las páginas una a una. Además, este formalismo permite acceder a otros recursos externos ajenos a la guía. Un ejemplo de este tipo de solución es la del portal *eTG* (TGL, 2017) de la organización australiana *Therapeutic Guideline Limited*.

La **informatización centrada en el documento** consiste en el uso de métodos y tecnologías de marcado semántico de GC textuales para la creación de GCI. Así, en un primer paso, es posible seleccionar los elementos relevantes del texto para facilitar la exploración del texto por parte del usuario médico

y posteriormente, en un segundo paso, se pueden convertir esos elementos relevantes en sentencias computables para que puedan ser utilizadas a la hora de realizar tareas de soporte a la decisión. El marcado se realiza utilizando un lenguaje específico, para lo cual se pueden utilizar lenguajes específicos de marcado como *Guideline Elements Model (GEM)* (Shiffman et al., 2000) o lenguajes para la definición de ontologías como *Ontology Web Language (OWL)*. **GEM** es un modelo de documento digital basado en XML que contiene más de 100 etiquetas para poder marcar el texto de una GC con varios niveles de abstracción (Shiffman et al., 2004). **OWL**, por su lado, permite aprovechar el poder de razonamiento automático que ofrecen los recursos de la Web Semántica. Para que sirva de ejemplo, la arquitectura *Digital Electronic Guideline Library* (DeGel) permite convertir de forma gradual una GC textual a una representación formal acorde a una ontología de elección (Shahar et al., 2004). También existen soluciones que unen los dos acercamientos mencionados y que permiten transformar una GCI marcada con GEM a lenguaje OWL (Tran et al., 2009).

La **informatización centrada en el modelo** consiste en utilizar un modelo conceptual de GCI que recoge todas las características contenidas en una GC textual, pero sin que haya una relación directa entre el modelo y la GC textual (Sonnenberg y Hagerty, 2006). Posteriormente, se puede utilizar un Sistema de Ayuda a la Toma de Decisiones (SATD) basado en esas GCI para poder particularizar el cuidado de los pacientes. En este enfoque hay que diferenciar entre el *modelo subyacente* y el *formalismo de representación* (Latoszek-Berendsen et al., 2010). El **modelo subyacente** recoge la representación abstracta o las características principales de una GC, mientras que el **formalismo de representación** establece la representación concreta o cómo deben representarse esas características.

El modelo subyacente debe ser capaz de representar todas las características de cualquier GC, por lo que puede ser una estructura de considerable complejidad. En (Clercq et al., 2008) se incluyen varios elementos comunes que conviene tener en cuenta en el modelo de GCI. Por ejemplo, indican que es conveniente proporcionar un conjunto de bloques genéricos para la construcción de las GCI, ofrecer la opción de añadir repeticiones y bucles e incluir múltiples niveles de abstracción, entre otros aspectos. En (Sonnenberg y Hagerty, 2006) añaden dos requisitos a ese modelo. Concretamente indican que el modelo debe posibilitar la conexión a Historias Clínicas Electrónicas (HCE) y ofrecer acceso a conocimiento específico del dominio, como puede ser terminología, información farmacéutica, etc.

La mayoría de los formalismos para la informatización de GCI centradas en el modelo parten de la base de que las tareas de las GC se pueden entender como redes jerárquicas de tareas que se desarrollan en el tiempo. Este paradigma se conoce como Modelo de Redes de Tareas (MRT) (Peleg et al., 2003) y surge, no como un intento de estandarización, sino como resultado de que diversos formalismos de representación de GCI tuviesen un modelo subyacente parecido. Estos formalismos de representación permiten definir GCI

utilizando los elementos especificados en el modelo subyacente, y por norma general, constan de un *lenguaje de control de flujo* y de un *lenguaje de expresión* (Clercq et al., 2008). El **lenguaje de control de flujo** se encarga de especificar cómo se estructura una GCI y cómo se relacionan los elementos del modelo entre ellos. El **lenguaje de expresión**, por su lado, se utiliza para especificar criterios de decisión y condiciones (por ejemplo, “El paciente tiene 39º de fiebre” y “Si el paciente tiene más de 38º, entonces administrarle ...”).

En lo que a los elementos del modelo se refiere, un formalismo de representación basado en el MRT debe soportar como mínimo dos tareas: las *acciones* y las *decisiones*. Las **acciones** están dirigidas tanto a ofrecer recomendaciones a los usuarios y asignarles tareas (por ejemplo, “Administrar X medicación al paciente”), como a tareas de computación (por ejemplo, “Obtener automáticamente los datos del último análisis de sangre del paciente del HCE”). Las **decisiones**, por su lado, se utilizan para dirigir el flujo del proceso (por ejemplo, “Si el paciente tiene más de 38º, entonces hacer A. En caso contrario hacer B”). Las decisiones pueden ser muy complejas, e incluso pueden utilizarse para diseñar sistemas de clasificación que permitan realizar cálculos complejos dentro de la GCI. Entre otros, en PROforma (Sutton y Fox, 2003) se utiliza su lenguaje de expresión para implementar un *sistema de clasificación basado en argumentación* y en GLARE (Terenziani et al., 2001) se utiliza para crear un *sistema de clasificación basado en votación*.

### 3.3. Formalismo de representación de GCI Aide

Aide es un formalismo de representación de GCI independiente de la plataforma de ejecución final, lo que facilita tanto la difusión como la reutilización de las GCI (Buenestado, 2014). Aide se basa en el uso de los MDDM para representar la *sintaxis abstracta* de una GCI. Adicionalmente, Aide también propone una *sintaxis concreta*, que describe el formalismo de representación concreto utilizado para cumplir los requisitos especificados en la sintaxis abstracta.

#### 3.3.1. Sintaxis abstracta

Aide está basado en el paradigma MRT. Por ello, las GCI representadas mediante Aide tienen una estructura de grafo dirigido de tareas ordenadas jerárquicamente. El nivel más bajo de abstracción lo establecen las tareas simples, las cuales se corresponden con los *nodos* (*Nodes*) del grafo. Las *transiciones* (*Transitions*) entre los nodos son las encargadas de dirigir el flujo del proceso. Aide dispone de seis tipos de nodos diferentes: *Formulario de preguntas* (*QuestionForm*), *Acción* (*Action*), *Recomendación* (*Recommendation*), *Cálculo* (*Calculation*), *Decisión* (*Decision*) y *Final* (*Final*).

Los nodos de **formulario de preguntas** se utilizan para conseguir información sobre el paciente al que se le está aplicando la GCI. La información

se puede obtener mediante petición directa al usuario o mediante el acceso a un HCE. En el primero de los casos el usuario introduce respuestas a las preguntas que el nodo le plantea. Existen tres tipos diferentes de preguntas (figura 3.2): *exclusivas*, *inclusivas* y *de entrada*. Las preguntas **exclusivas** admiten una única respuesta de una lista de respuestas posibles, mientras que las preguntas **inclusivas** admiten más de una respuesta. En las preguntas **de entrada**, por su lado, el usuario debe introducir un dato de entrada.

The figure shows three distinct question formats:

- Exclusiva (Left):** A question box titled "¿Cuál es el sexo del paciente?" with two radio button options: "Hombre" (selected) and "Mujer".
- Inclusiva (Center):** A question box titled "¿Cuáles son los síntomas clínicos que presenta el paciente?" with four checkbox options: "Buen estado general" (unchecked), "Intolerancia digestiva" (checked), "Hipotonía" (unchecked), and "Quejidos" (checked). Below these are two more checked options: "Mareos" and "Mareos".
- Entrada (Right):** A question box titled "¿Cuál es el nivel de amonio en sangre?" with a text input field containing the number "350" and the unit "μmol/L".

**Figura 3.2.** Ejemplo de pregunta exclusiva (izquierda), inclusiva (centro) y de entrada (derecha)

Los nodos de **acción** y **recomendación** se utilizan para indicar al usuario las acciones que se deben llevar a cabo. La diferencia entre estos dos tipos de nodos es semántica. Así, mientras que en los nodos de recomendación realizar la acción es opcional y siempre se requerirá la confirmación del usuario, las acciones indicadas en los nodos de acción son imprescindibles para poder continuar a la siguiente tarea y el sistema siempre supondrá que el usuario la ha realizado.

El cometido de los nodos de **cálculo**, por su lado, consiste en evaluar las expresiones lógicas o matemáticas o los sistemas de clasificación que tengan asociados. El formalismo no marca un formato determinado para estas expresiones, pudiendo utilizarse cualquier lenguaje para el que se disponga de soporte en ejecución.

En lo que a los nodos de **decisión** se refiere, son los únicos nodos que tienen más de una transición de salida. Aíde incluye dos tipos diferentes: *decisión exclusiva* y *decisión inclusiva*. Los nodos de **decisión exclusiva** se utilizan para discriminar entre diferentes alternativas, mientras que los nodos de **decisión inclusiva** se utilizan para habilitar secciones de ejecución paralela. Los transiciones de salida de los nodos de decisión exclusiva contienen condiciones, de las cuales la que se cumpla indicará cuál es el siguiente nodo a ejecutar. Estas condiciones tienen la cualidad de que son exclusivas entre ellas, esto es, nunca se puede dar el caso de que con la información disponible se cumpla más de una condición a la vez. De esta forma se puede discriminar entre diferentes alternativas. Los nodos de decisión inclusiva, por su lado, no disponen de condiciones en sus transiciones de salida. De esta forma, cuando la ejecución llega a uno de estos nodos, se crea un hilo de ejecución por cada transición de salida. Así, se habilita una sección de ejecución paralela, esto

es, una sección donde no se indica explícitamente el orden en que se deben realizar las tareas indicadas en los nodos incluidos en esa sección.

Por último, los nodos de **final** no disponen de transiciones de salida y se utilizan para especificar las últimas acciones que se deben realizar antes de dar por terminada la ejecución de la GCI.

Las tareas que forman la GCI se pueden agrupar para crear tareas complejas. Estas agrupaciones de tareas permiten simplificar la visión que el usuario tiene de la GCI, añadiendo diferentes niveles de abstracción. Las tareas complejas también se pueden utilizar para definir patrones reutilizables. De esta forma, una tarea compleja definida en una GCI también se podrá utilizar en otras GCI posteriores. Para poder definir estas tareas complejas se han definido dos nuevos elementos de representación: *sección* (*Section*) y *conjunto* (*Set*). La estructura **sección** está formada exclusivamente por nodos y la estructura **conjunto** puede contener tanto secciones como conjuntos. El nivel más alto de una GCI se representa mediante un conjunto.

Aparte de todos estos elementos organizativos y centrados en las tareas de la GCI, Aide también ofrece distintos elementos para poder representar tanto información específica del dominio (terminología médica, información farmacológica, etc.) como los propios metadatos de la guía (información didáctica y de mantenimiento, autores, versionado, etc.).

Todas las características sintácticas de las CGI descritas hasta este punto se han formalizado mediante el metamodelo que se puede observar en las figuras 3.3 y 3.4. En ellas se pueden distinguir todos los componentes que forman una GCI en el formalismo Aide, así como las diversas asociaciones y restricciones existentes entre los mismos.

En el centro de la figura 3.3 dispuestas de abajo arriba se muestran las metaclases correspondientes a los principales componentes de una GCI, esto es, los elementos *nodo*, *sección* y *conjunto*, representados en el metamodelo mediante las metaclases *Node*, *Section* y *Set* respectivamente. Todas ellas son especializaciones de la metaclase que representa cualquier elemento de la GCI (metaclase *GCIComponent* a la derecha de la metaclase *Section*). La metaclase *GCIComponent*, por su parte, está relacionada con las metaclases utilizadas para representar información específica del dominio: las metaclases *Resource*, *Reference*, *Substance*, *Evidence* y *Concept*. La metaclase *GCI* permite representar una GCI y se sitúa en la figura 3.3 encima de la metaclase *Set*, de la que es especialización. Por último, la metaclase *GCI* está relacionada con la metaclase encargada de modelar los metadatos de la guía (*COGS*). Esta última metaclase se sitúa encima de *GCI*.

En la figura 3.4, por su lado, se muestra la metaclase *Node*, la cual se especializa en todos los tipos de nodos soportados en Aide. Así, de izquierda a derecha, se pueden observar las metaclases *Action*, *Calculation*, *Question-Form*, *Decision*, *Recommendation* y *Final*.

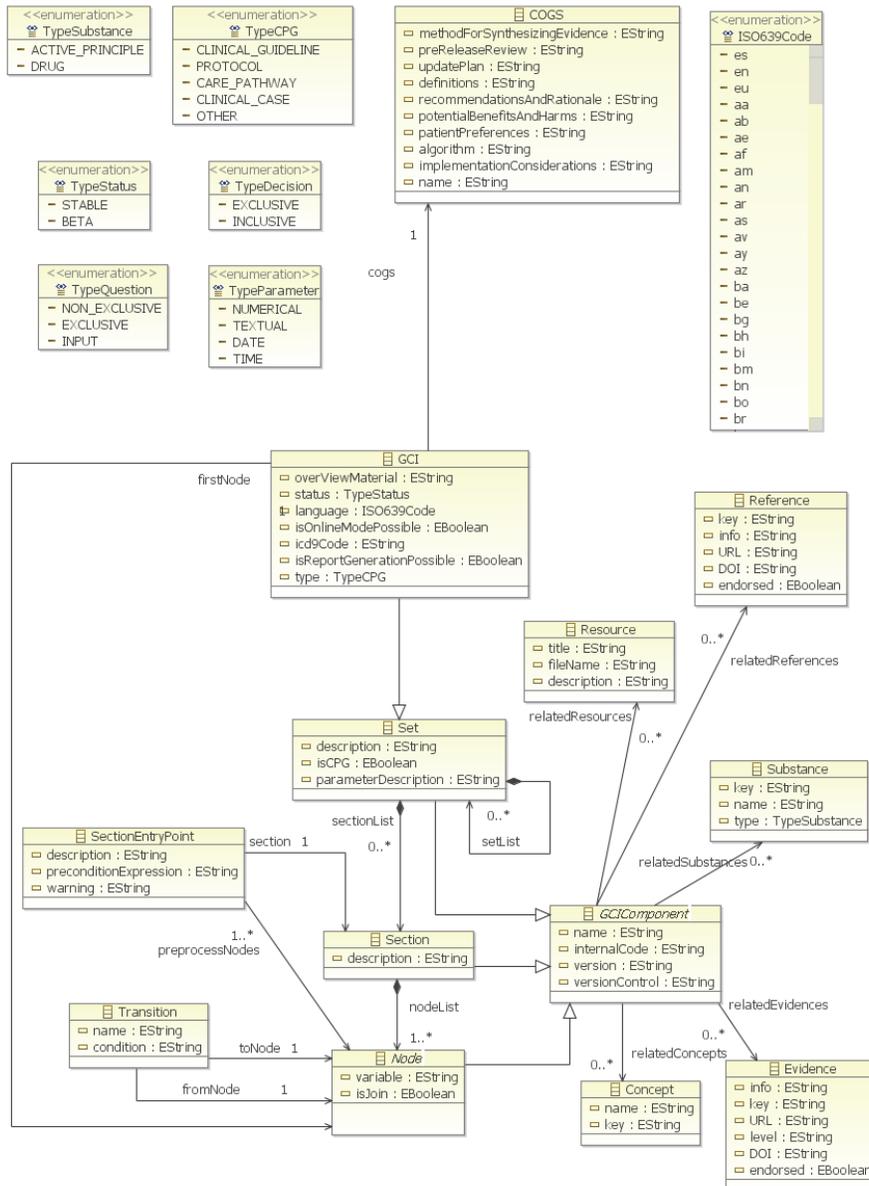


Figura 3.3. Metamodelo de GCI en el formalismo Aide (parte 1)

### 3.3.2. Sintaxis concreta

La sintaxis concreta de representación de GCI de Aide se basa en una arquitectura de arquetipos OpenEHR. OpenEHR es una iniciativa que se cen-

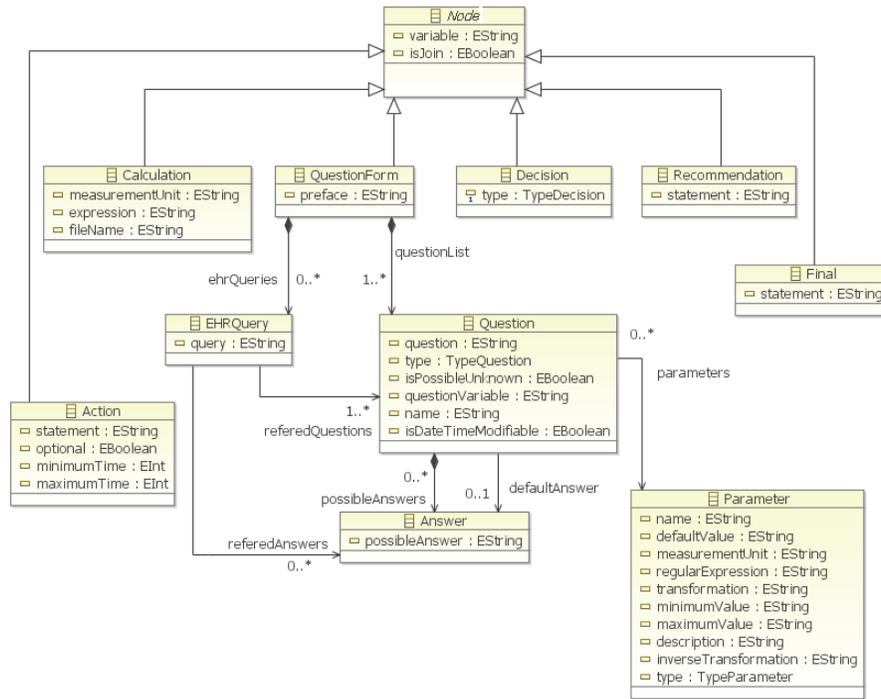
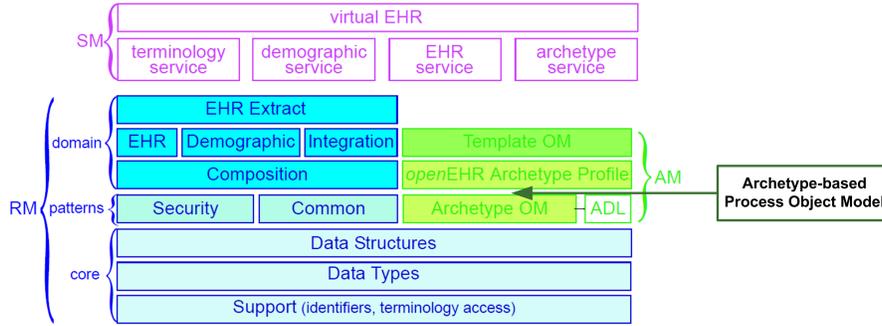


Figura 3.4. Metamodelo de GCI en el formalismo Aide (parte 2)

tra en la definición de una arquitectura de sistemas HCE adaptables (Beale y Heard, 2008). Un arquetipo se define como “una expresión computable de un modelo de dominio, basada en restricciones y que tiene su origen en un modelo de referencia” (Beale, 2002). Más informalmente, un arquetipo establece una serie de atributos que caracteriza un concepto en un dominio determinado. Los arquetipos se utilizan para que los expertos de un dominio representen o modelen conceptos de ese dominio de un modo formal. Existen varios proyectos que utilizan arquetipos en el dominio clínico y uno de ellos es, justamente, OpenEHR. OpenEHR define un modelo de referencia a partir del cual se pueden especificar arquetipos para el modelado de conceptos médicos.

La arquitectura de arquetipos de Aide se ha denominado Modelo de Objetos de Proceso basado en Arquetipos (*Archetype-based Process Object Model, A-POM*) y se ha construido sobre el Modelo de Arquetipos (*Archetype Model, AM*) de OpenEHR (ver figura 3.5). Aide utiliza 11 arquetipos diferentes para modelar la información de las principales clases de la sintaxis abstracta, a saber *GCIComponent*, *GCI*, *Set*, *Section*, *Node* y cada una de las clases en que se especializa *Node*.

Aide también dispone de una sintaxis concreta para la representación gráfica de las GCI. Ésta se basa en la ampliamente utilizada notación algorítmica



**Figura 3.5.** Situación del A-POM respecto al modelo de arquetipos de OpenEHR

de las GC textuales. De esta forma, a cada elemento de la GCI le corresponde una forma y color diferente.

En la parte derecha de la figura 3.6 se muestra la fracción inicial de una GCI para el diagnóstico y tratamiento de la hiperamonemia. En esta GCI el primer paso consiste en realizar un análisis de sangre. Esta tarea se corresponde con un nodo de *acción* y en la figura se representa mediante un rectángulo de color rojo. Posteriormente se introducen los resultados de ese análisis y unos datos iniciales del paciente (fecha de nacimiento y género). Esto se realiza mediante un nodo de *formulario de preguntas*, el cual se representa mediante un rectángulo azul de tres dimensiones. Tras esto, la GCI realiza dos *cálculos* (representados mediante rectángulos de color marrón en la figura 3.6) y seguidamente realiza una recomendación. El nodo de *recomendación* se representa mediante un rectángulo de color amarillo en la figura. Dependiendo de la recomendación la GCI puede tomar distintas alternativas. Para ello se utiliza un nodo de *decisión exclusiva*, de forma que únicamente se va a activar una de sus transiciones de salida. Este nodo se representa mediante un rombo verde en la figura 3.6 y cada una de sus transiciones de salida indica la condición que se debe cumplir para que pueda activarse. La primera de las transiciones de salida (a la izquierda) del nodo de *decisión* lleva a un final de la GCI. Los nodos de *final* se representan mediante elipses de color rojo.

En la parte izquierda de la figura 3.6, por su lado, se muestra la relación gráfica entre los elementos de la GCI *nodo*, *sección* y *conjunto*. El rectángulo que engloba todas las demás figura se corresponde con un *conjunto*. Este *conjunto* está formado por dos *secciones*, los dos rectángulos contenidos (arriba y abajo) dentro de la figura del *conjunto*. Por último, el rectángulo correspondiente a la *sección* superior muestra una estructura parecida a la de la parte derecha de la figura 3.6. De esta forma, cada una de las pequeñas figuras en ese rectángulo se corresponde con un *nodo*.

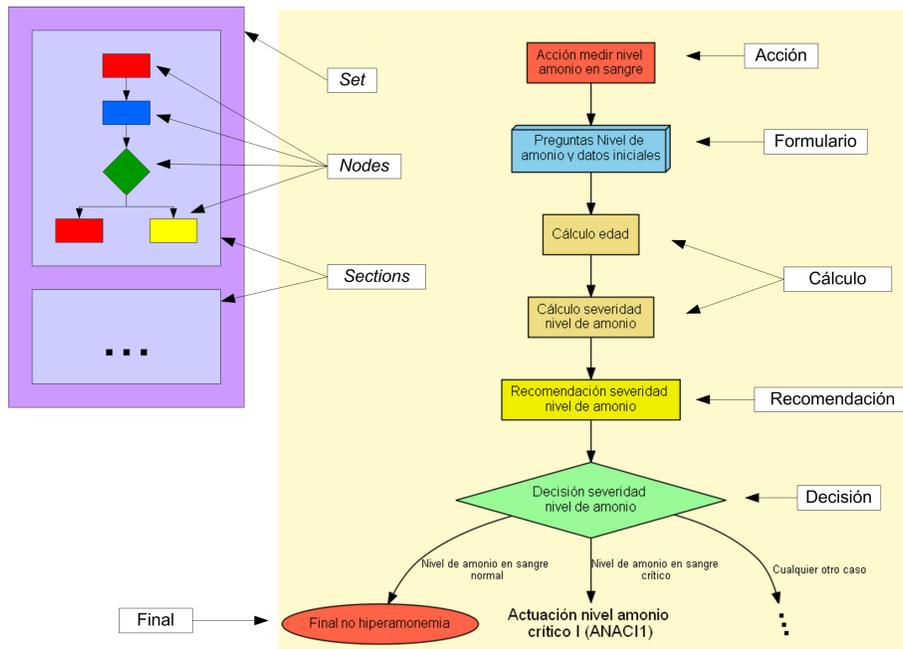


Figura 3.6. Estructuras *Node*, *Section* y *Set* (izquierda) y tipos de nodos (derecha)



## Aide Guideline Technology Platform

*Aide Guideline Technology Platform* (Aide-GTP) es una plataforma tecnológica creada para dar soporte a todo el ciclo de vida de las GCI, desde su definición hasta su ejecución (Buenestado, 2014). Estas GCI están representadas siguiendo el formalismo Aide, el cual se ha presentado en el capítulo 3.

En primer lugar se ofrece una vista general de la plataforma para luego centrarse en sus módulos específicos para la edición y ejecución de GCI.

### 4.1. Vista general de Aide Guideline Technology Platform

La arquitectura general de Aide-GTP se puede observar en la figura 4.1. En esta figura se distinguen dos bloques principales. El bloque de la izquierda representa la plataforma de edición *Aide Developer Suite* (*Aide-DS*), mientras que el de la derecha representa la plataforma de ejecución *Aide Guideline Execution Platform* (*Aide-GEP*). Las GCI (representadas mediante el rectángulo en el centro de la figura 4.1) se desarrollan utilizando los componentes incluidos en Aide-DS y se despliegan y ejecutan en Aide-GEP.

En la parte baja de la figura 4.1 se muestran los distintos actores que interactúan con los diferentes componentes de Aide-GTP. Así, el *ingeniero del conocimiento* y el *experto médico* utilizan Aide-DS, mientras que el *experto médico*, el *médico general* y el *administrador* utilizan Aide-GEP. El **ingeniero del conocimiento** tiene un amplio conocimiento informático y sobre Aide-GTP. Aunque su conocimiento médico es limitado, es capaz de modelar ese conocimiento en un soporte informático. El **experto médico**, por su lado, es un actor con amplios conocimientos en el ámbito de la GCI, pero con limitado conocimiento informático. Es el actor responsable de la corrección clínica de la GCI. El ingeniero de conocimiento y el experto médico son los encargados de definir las GCI. El experto médico también es el encargado de contestar las dudas que puedan surgirle a los usuarios durante la ejecución de las GCI,

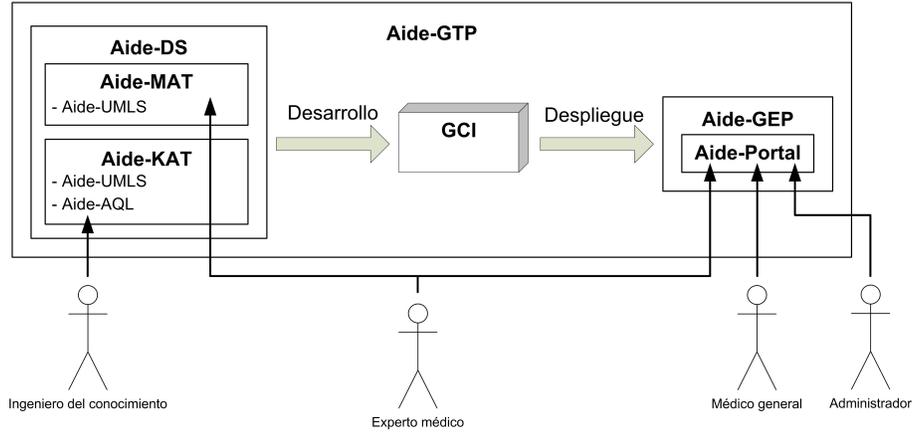


Figura 4.1. Vista general de la plataforma Aide-GTP

de ahí su relación con la plataforma Aide-GEP. El **médico general** es un actor que dispone de conocimiento médico general, pero que no llega a tener un conocimiento sobre el área de la GCI tan amplio como el experto médico. La funcionalidad principal del rol de médico general consiste en ejecutar una GCI. Por último, el **administrador** puede ser cualquier persona con amplio conocimiento sobre Aide-GEP, ya sea médico, administrativo o ingeniero del conocimiento, ya que su tarea consiste en gestionar los contenidos y los usuarios de Aide-GEP.

## 4.2. Aide Developer Suite

La plataforma de edición Aide-DS está formada por dos editores (Buenestado et al., 2011): *Aide Medical Authoring Tool (Aide-MAT)* y *Aide Knowledge Authoring Tool (Aide-KAT)*. Estos editores se han creado siguiendo el formalismo de representación Aide, de forma que las GCI que permiten crear cumplen las restricciones del metamodelo de Aide.

**Aide-MAT** es un editor básico que permite definir GCI preliminares no directamente ejecutables y ha sido especialmente diseñado para su uso por parte de los expertos médicos. Las GCI se describen utilizando una paleta de elementos que sigue la representación gráfica de Aide y un área de dibujo (ver figura 4.2). La descripción de la guía se realiza desplazando los distintos elementos de la paleta a la derecha de la figura 4.2 al área de dibujo. Las propiedades de los elementos se establecen en el menú inferior, mientras que en las ventanas a la izquierda de la figura 4.2 se muestra la estructura general de la guía en forma de árbol y de forma gráfica, arriba y abajo respectivamente.

Todos los elementos se comportan de manera coherente a como se han definido en su metamodelo. Aide-MAT, además, ofrece varias utilidades típi-

cas de un *Integrated Development Environment (IDE)* avanzado, como hacer zoom en la guía, ofrecer vistas previas o permitir la edición de textos. También incluye otras funcionalidades, entre las que destaca un editor para realizar búsquedas terminológicas en *Unified Medical Language System (UMLS)* llamado *Aide UMLS Terminology Finder (Aide-UMLS)* (ver el componente Aide-MAT en la figura 4.1). UMLS es un repositorio terminológico estándar que ofrece servicios para la creación de sistemas médicos interoperables (UMLS, 2017). Aide-UMLS permite buscar términos en el repositorio de UMLS mediante servicios web y asignar éstos a las entidades de tipo *Node*, *Section* o *Set* de la GCI que se esté desarrollando.

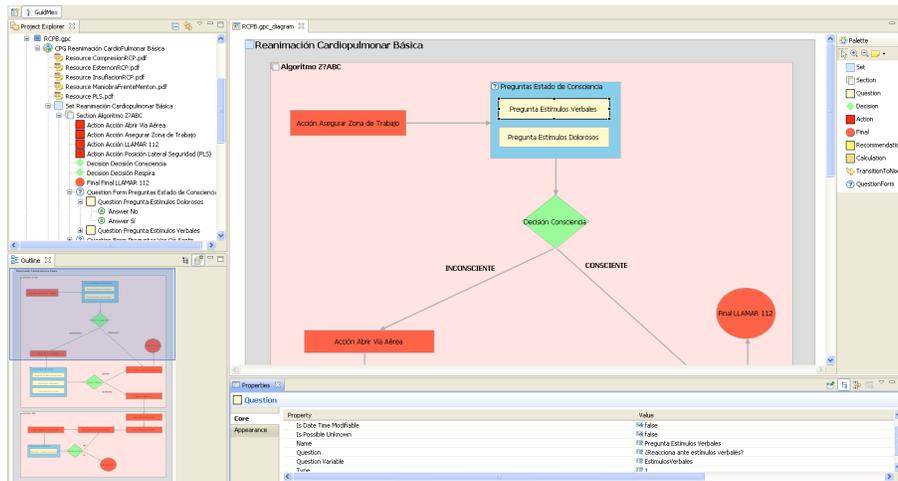


Figura 4.2. Interfaz de Aide-MAT

**Aide-KAT** es una extensión del editor Aide-MAT y permite completar los modelos creados mediante Aide-MAT, añadiéndoles las características técnicas necesarias para que las GCI sean ejecutables. Aide-KAT ha sido diseñado para ser utilizado por ingenieros del conocimiento, debido a su capacidad para comprender y completar los modelos de GCI creados por los expertos médicos utilizando Aide-MAT.

Las características que se establecen en Aide-KAT están más cerca de los conceptos computacionales que de los conceptos médicos de la GCI. Así, por ejemplo, en Aide-KAT se definen las expresiones formales correspondientes a las condiciones de las transiciones de salida en los nodos de decisión exclusiva. Visualmente Aide-KAT es parecido a Aide-MAT, con la diferencia de que Aide-KAT ofrece la posibilidad de establecer los valores de más propiedades de los componentes que forman la GCI. Con Aide-KAT, por ejemplo, se pueden establecer las expresiones mencionadas o los códigos de identificación de los nodos, características que no se pueden establecer con Aide-MAT. Aide-KAT

también incluye el editor Aide-UMLS, así como un editor para la definición de consultas en repositorios de HCE llamado *Aide Archetype Query Editor* (*Aide-AQL*) (ver el componente Aide-KAT en la figura 4.1).

### 4.3. Aide Guideline Execution Platform

Aide-GEP es una plataforma de ejecución que permite ejecutar GCI en formato Aide. Esta plataforma está compuesta por un motor de ejecución encargado de dirigir el proceso que se llama *Aide Guideline Execution Engine* (Aide-Engine) y un portal web, cuya última versión es Aide-Portal (accesible en <http://www.e-guidesmed.ehu.es>).

Cada uno de los tres diferentes roles del portal, *Administrador*, *médico general* y *experto médico*, puede realizar diferentes acciones. Así, el **administrador** se encarga de gestionar tanto las GCI como los usuarios. Por lo tanto, se encarga de desplegar guías, eliminar guías obsoletas, crear nuevos usuarios, etc. El **médico experto**, por su lado, realiza su tarea de gestionar las dudas médicas de los usuarios mediante el intercambio de mensajes en el propio portal. Por último, el **médico general** ejecuta las GCI siguiendo paso a paso las tareas definidas en la misma y consultando los recursos adicionales que se ofrecen en cada uno de los pasos. Así, aparte de obtener la información relativa a un paso de la GCI, el usuario médico también puede consultar información bibliográfica, terminológica, de evidencia y, cada vez que se recomienda administrar un principio activo, información sobre el mismo. De la misma manera, en cualquier momento puede consultar el grafo de la GCI, así como todos los metadatos relativos a la misma. El grafo es la representación gráfica de los pasos de la guía y los metadatos incluyen información sobre el mantenimiento, los autores, el versionado, etc. Por otro lado, las ejecuciones pueden ser pausadas y reanudadas en cualquier punto de la GCI. Por último, se guarda un registro de todas las ejecuciones realizadas para que puedan ser consultadas o compartidas por los usuarios que las hayan ejecutado.

En la figura 4.3 se muestra la interfaz del portal durante la ejecución de una GCI, así como de las diferentes fuentes de información adicional que se pueden consultar. En la parte superior izquierda se puede observar la interfaz de Aide-Portal durante la ejecución de una GCI. En la parte inferior izquierda, por su lado, se muestra el detalle de los resultados de un cálculo realizado en una GCI. Finalmente, en la parte derecha y de arriba abajo se muestra la imagen del grafo, un detalle sobre la información bibliográfica y un detalle de la información farmacéutica de una GCI.

**e-GuidesMed**  
 Ejecutable Medical Guidelines Portal  
 Portal de Guías Médicas Ejecutables  
 Medikuntzako Gaiak Egikargarriren Ataria

**DIAGNÓSTICO Y TRATAMIENTO DE LA HIPERKALEMIA**

La actuación terapéutica será común para la hiperkalemia de cualquier etiología hasta conocer el diagnóstico. Para ello se administrará un conjunto de catálisis que tendrá efecto en una serie de enfermedades o deficiencias enzimáticas:

- **Etacryn** (20-80 mg/día): Cofactor de los carboxilasa
- **Hidroxibutirato** (1 mg/día): Precursor de los cofactores de la metiltransferasa con metasa y la renetación de la homocisteína.
- **Trisofato** (500-600 mg/día): Cofactor de las transaminasa.
- **Etilbutila** (100-300 mg/día): Cofactor de las deshidrogenasa.
- **Tiamina** (300 mg/día): Cofactor de las descarboxilasa.

**Tabla importante** - Lista de medicamentos prohibidos:

- TOTALMENTE: **Valerato, Midazolam, Ácido acetilsalicílico (aspirina), Fluorocitilo y ICT** (hasta diagnóstico). En caso de que se esté administrando alguno de estos medicamentos, dicho tratamiento se deberá interrumpir inmediatamente.
- CON PRECAUCIÓN si consideramos: **Enofilo, Gabapenteno, Topiramato y Fosfobital**

**La recomendación realizada tiene un coeficiente de calidad de información (88.89 %) Trastornos de la cetogénesis - Índice de calidad de información (QoI, Quality of Information) de 85.70 %**

Valores introducidos	Interpretación de los valores introducidos	
- pH: 6.8	- Muy bajo (50.0 %) - Bajo (50.0 %)	- Muy bajo [5.0, 7.0]
- Amonio (mmol/L): 120	- Discretamente elevado (100.0 %)	- Normal [5.0, 25.0]
- Lactato (mmol/L): 2	- Normal (100.0 %)	- Discretamente elevado [10.0, 20.0]
- GOT-GPT (U/L): -	- Discretamente elevado (16.67 %) - Elevado (83.33 %)	- Elevado [70.0, 150.0]
- CPK (U/L): 400	- Normal (100.0 %)	- Elevado [350.0, 1000.0]
- Ácido úrico (mg/dl): 3	- Normal (100.0 %)	- Normal [0.0, 7.0]
- 3-Hidroxibutirato (mmol/L): 0.15	- Bajo (100.0 %)	- Bajo [0.0, 0.0, 0.25, 0.35]

**ASCÓRBICO ÁC. + HIDROXICOBALAMINA**

Mecanismo de acción:  
 Activador del metabolismo y tiene notables propiedades como desintoxicante y protector frente a las lesiones (epilepsia, hepatitis, etc.).

Indicaciones terapéuticas:  
 déficit de Vit. A y Vit. B12. Afecciones gastrointestinales, cardiovasculares, hepáticas, renales, de la piel, con antibióticos. Debilidad general, fatiga, insomnio, migrañas, neuralgias, astenia, procesos infecciosos, dermatitis, urticaria y alergias. Tius, pre y postoperatorios.

Posología:  
 Oral: Ascórbico ác./hidroxicoBALAMINA. 1 g/500 mcg. 2 g/1000 mcg al día.

Figura 4.3. Interfaz de Aide-Portal



## Sistemas de Inferencia Difusa

Los sistemas de clasificación son métodos algorítmicos cuya función es clasificar una serie de casos de entrada en un número predeterminado de clases o categorías siguiendo unos criterios específicos. Los sistemas de clasificación tienen múltiples usos en ámbitos muy diferentes. Así, existen aplicaciones para el reconocimiento de imágenes y voz, controles industriales, predicciones en el mercado financiero, predicciones del tiempo atmosférico, etc.

Los **sistemas basados en reglas** tienen como objetivo capturar el conocimiento humano en un dominio concreto para poder crear sistemas que emulen el razonamiento de las personas (Frye et al., 1995). Para ello utilizan reglas proposicionales de tipo *if-then*, donde la parte *if* se denomina *antecedente* y la parte *then* se denomina *consecuente*. El **antecedente** establece la condición que debe cumplirse para afirmar que el **consecuente** también se cumple. Las reglas se evalúan sobre conjuntos de **hechos**, siendo los hechos sentencias que describen situaciones que son ciertas y verificables. Por ejemplo, *la Tierra es redonda* y *los coches tienen cuatro ruedas* son dos hechos.

La **lógica difusa** se basa en que el uso de términos lingüísticos en lugar de números permite un acercamiento más intuitivo a los problemas sin entrar en complejidades matemáticas. Por ejemplo, para definir la altura de una persona, utilizar términos como *bastante alto* o *un poco bajo* es más sencillo que indicar exactamente con un número la altura en cuestión de esa persona. El uso de la lógica difusa es útil en el momento en el que se empieza a trabajar con *cómo* percibe la gente un concepto, en oposición a un interés únicamente clasificatorio o de contabilidad, ya que la lógica difusa permite contestar a preguntas de tipo *sí-no* utilizando respuestas de tipo *un-poco-sí-un-poco-no*. Volviendo al ejemplo anterior, la lógica difusa permite contestar a la pregunta *¿Es esa persona alta?* utilizando respuestas del tipo *Es bastante alta* o *No es muy alta*, en lugar de las monosilábicas *Sí* y *No*.

Dentro de los sistemas de clasificación se encuentran los Sistemas de Inferencia Difusa (SID). Los SID son el resultado de integrar la *lógica difusa* en los *sistemas basados en reglas*. Esta integración propicia que los hechos del sistema basado en reglas tengan un grado de certeza, esto es, los hechos dejan

de ser verdades absolutas y pueden cumplirse parcialmente (Bezdek et al., 1999). De esta forma se consigue que los SID sean capaces de funcionar en escenarios con incertidumbre. En concreto, los SID tienen una gran aceptación en el ámbito de la medicina debido a su capacidad para representar la incertidumbre de los diagnósticos y la imprecisión de los síntomas (Khatibi y Montazer, 2010).

Este capítulo se divide en tres secciones. En la primera de ellas se explica el concepto de la lógica difusa, ya que es necesario para comprender el funcionamiento de los SID. En la segunda sección se presentan en profundidad los SID. En la última sección se introducen los *framework* para el desarrollo de SID.

## 5.1. Lógica difusa

La lógica difusa fue formulada por Lofti Zadeh en 1965 (Zadeh, 1965) y desde su primera aparición, el número y la variedad de aplicaciones de la lógica difusa se ha incrementado considerablemente (Cabrera et al., 2009) gracias a la incapacidad de los acercamientos convencionales para trabajar de forma efectiva en escenarios con incertidumbre (Alayón et al., 2007).

Una variable difusa está formada por un número determinado y concreto de valores lingüísticos. Así, por ejemplo, se puede definir una variable difusa de nombre *Altura de una persona* cuyos valores lingüísticos sean  $\{Muy\ baja, Baja, Normal, Alta, Muy\ Alta\}$ .

La lógica difusa está interrelacionada con la teoría de conjuntos. De esta forma, a cada valor lingüístico de una variable difusa le corresponde un *conjunto difuso*. Los conjuntos difusos son una generalización de los *conjuntos clásicos*. Mientras que en los **conjuntos clásicos** los elementos están bien totalmente dentro bien totalmente fuera, en los **conjuntos difusos** los elementos puedan estar a la vez dentro y fuera. En el ámbito de la teoría de conjuntos, el **universo de discurso** se define como un conjunto que contiene todos los elementos sobre los que se habla en un determinado contexto. Por ejemplo, para el universo de discurso de los *Animales* todos los animales existentes son elementos de ese conjunto.

Los elementos del universo de discurso se pueden clasificar en conjuntos más pequeños en base a ciertas reglas o criterios. Así, en el universo de discurso de los *Animales*, se puede definir un conjunto llamado *Peces* que incluya todos los elementos del universo de discurso que sean peces. Se puede afirmar que este conjunto es clásico, ya que contiene todos los animales que son peces y no contiene ninguno que no lo sea (ver figura 5.1). La **pertenencia** de un elemento a un conjunto indica “cuánto” se puede considerar dentro de ese conjunto al elemento en cuestión y se representa mediante el símbolo  $\mu$ . Siendo el  $0$  el valor que indica la total no-pertenencia de un elemento al conjunto y  $1$  la total pertenencia al mismo, en los conjuntos clásicos la pertenencia sólo puede tener uno de esos dos valores. Por ejemplo, el elemento *Perro* tiene

una pertenencia de 0 y el elemento *Salmón* tiene una pertenencia de 1 en el conjunto de la figura 5.1.

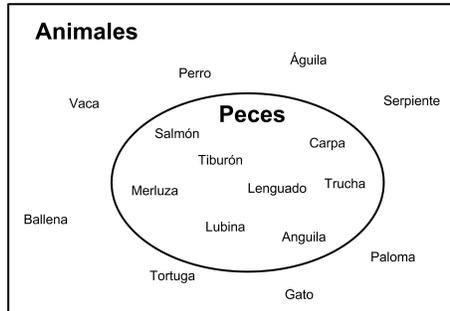


Figura 5.1. Ejemplo de un conjunto clásico

Supóngase ahora el conjunto *Peces de agua dulce* dentro del universo de discurso *Peces* (ver figura 5.2). Aunque la mayoría de los peces se pueden clasificar claramente dentro o fuera de este conjunto, no ocurre lo mismo con el *Salmón* y la *Anguila*, ya que ambos son peces que viven tanto en agua dulce como salada. Por ello se puede afirmar que el conjunto *Peces de agua dulce* es un conjunto difuso. En estos conjuntos la pertenencia se indica mediante un valor en el rango  $[0, 1]$ . Por ejemplo, se podría afirmar que el elemento *Carpa* tiene una pertenencia de 1 en el conjunto de la figura 5.2, el elemento *Merluza* tiene una pertenencia de 0 y el elemento *Anguila* tiene una pertenencia alrededor del 0.5, ya que vive en agua dulce y salada.

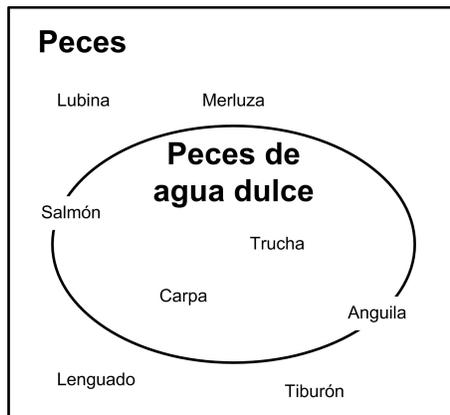
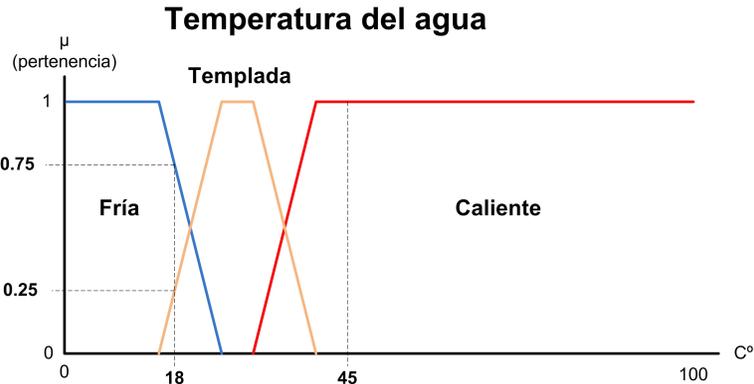


Figura 5.2. Ejemplo de un conjunto difuso

Las **funciones de pertenencia (FP)** son curvas que describen la forma de los conjuntos difusos de forma que sirven para calcular la pertenencia de los elementos del universo de discurso en los conjuntos que describen. Cualquier curva puede ser una FP, pero normalmente se utilizan funciones matemáticas estándares, tales como las funciones trapezoidales o las gaussianas. Eso sí, todas cumplen la cualidad de que para cualquier valor de entrada (un elemento concreto) su valor de salida (la pertenencia del elemento) será un número en el rango  $[0, 1]$ . En el apartado 5.1.1 se describen varios tipos de FP.

En la figura 5.3 se muestra la variable difusa *Temperatura del agua*, la cual está compuesta por tres valores lingüísticos: *Fría*, *Templada* y *Caliente*. En el eje X se representan los elementos del universo del discurso, los cuales para la temperatura del agua abarcan todos los valores desde los  $0^{\circ}$  centígrados hasta los  $100^{\circ}$  centígrados. En el eje Y se representa la pertenencia. Cada uno de los valores lingüísticos está representado por un conjunto difuso que sigue una función trapezoidal y los conjuntos se superponen de forma que algunos elementos del universo del discurso pertenecen a más de un conjunto. En la misma figura también se ha situado la posición del valor de  $45^{\circ}$  centígrados, el cual obtiene un grado de pertenencia de 1 en el conjunto difuso *Caliente*, mientras que obtiene un grado de pertenencia de 0 para los otros dos conjuntos difusos. El valor  $18^{\circ}$  centígrados, por su lado, obtiene un grado de pertenencia de 0.75 en el conjunto *Fría* y de 0.25 en el conjunto difuso *Templada*.



**Figura 5.3.** Conjuntos difusos que componen la *Temperatura del agua*

Una variable que expresa sus valores lingüísticos mediante conjuntos difusos se considera *completa* cuando el grado de pertenencia acumulado para todos sus valores del universo de discurso en los conjuntos difusos es 1. Por ejemplo, la variable *Temperatura del agua* de la figura 5.3 es completa, ya que cualquier valor de su universo de discurso cumple esa característica. Como se ha visto en el ejemplo anterior, tanto una temperatura de  $45^{\circ}$  como una de  $18^{\circ}$  obtienen una pertenencia acumulada de 1. Para el caso de los  $45^{\circ}$ , obtiene

esa pertenencia de un único conjunto (*Caliente*) y tiene una pertenencia de 0 en el resto de conjuntos. En el caso de la temperatura de 18º, la suma de sus pertenencias a los conjuntos *Fría*, *Templada* y *Caliente* también da un valor de 1, siendo la pertenencia en el último conjunto de 0.

En la figura 5.4 se muestra la variable *Colesterol LDL*. No se puede considerar esta variable completa, ya que no se cumple la característica mencionada. Por ejemplo, un valor de 155 obtiene una pertenencia de 0.85 en los conjuntos *Discretamente alto* y *Alto*, lo que hace una pertenencia acumulada de 1.70. De forma parecida, un valor de 185 obtiene una pertenencia de 0.35 en los conjuntos *Alto* y *Muy alto*, por lo que su pertenencia acumulada es de 0.70.

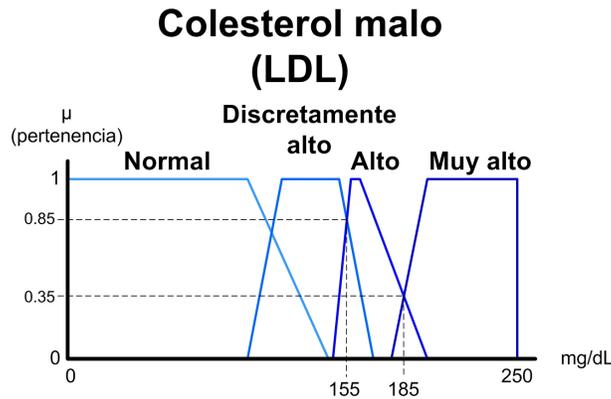


Figura 5.4. Conjuntos difusos que componen el *Colesterol LDL*

La lógica difusa se divide en dos grupos, la *lógica difusa de nivel 1 (LD-1)* y la *lógica difusa de nivel 2 (LD-2)*. En la primera la pertenencia de cualquier valor de entrada en cualquier conjunto difuso se expresa mediante un valor en el rango [0, 1]. En la LD-2, sin embargo, la pertenencia se expresa mediante un conjunto difuso (Castillo y Melin, 2008). Esto añade un mayor nivel de incertidumbre, al mismo tiempo que también añade mayor complejidad a los cálculos.

Al igual que los conjuntos difusos generalizan el funcionamiento de los conjuntos clásicos, las operaciones entre conjuntos difusos también generalizan las operaciones de los conjuntos clásicos. Por lo tanto, es perfectamente posible realizar operaciones de unión o intersección entre conjuntos difusos. Mientras que el resultado de estas operaciones entre conjuntos clásicos es un conjunto clásico, el resultado de las operaciones con conjuntos difusos será otro conjunto difuso. En el apartado 5.1.2 se concretan varias operaciones lógicas entre conjuntos difusos.

### 5.1.1. Tipos de funciones de pertenencia

Existen múltiples tipos de FP. Aunque la mayoría están basadas en funciones matemáticas básicas (Jang et al., 1997), cualquier función creada *ad hoc* sirve como FP siempre que se cumpla la condición de que todos los valores del universo de discurso tengan un grado de pertenencia en el rango  $[0, 1]$ .

Las funciones más sencillas son aquéllas que se pueden representar mediante una única línea recta. Un ejemplo de estas funciones son la **función lineal** y la **función constante o singleton** (ver figura 5.5).

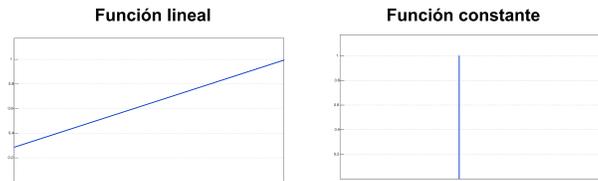


Figura 5.5. Funciones de pertenencia de tipo lineal y constante

Otras funciones se expresan mediante la combinación de varias líneas rectas. Las más comunes son la **función triangular** y la **función trapezoidal** (ver figura 5.6). La principal ventaja de estas FP es su simplicidad. Para expresar la función triangular sólo es necesario indicar tres puntos y para expresar la función trapezoidal cuatro. Estos puntos encajan con los puntos en el universo de discurso en que las funciones cambian de pendiente.

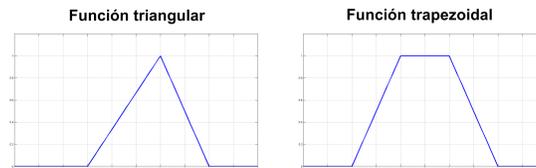


Figura 5.6. Funciones de pertenencia de tipo triangular y trapezoidal

La **función gaussiana** y la **función de campana** (ver figura 5.7) tienen una notación concisa y la cualidad de tener pendientes suaves y valores diferentes de 0 en todo el universo de discurso. Para expresar la función gaussiana únicamente es necesario indicar el punto medio de la campana que forma la función y la desviación estándar de la misma. La función de campana necesita un parámetro más para poder indicar el radio de la curva de la campana que crea la función. La correcta afinación de los parámetros de esta última función permite crear una función muy cercana a un conjunto clásico o no difuso.

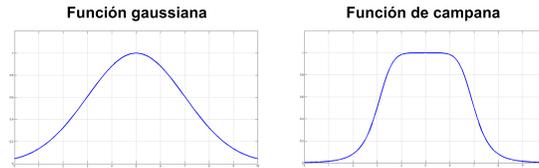


Figura 5.7. Funciones de pertenencia de tipo gaussiana y de campana

La **función sigmoidea** y la **función polinómica** (ver figura 5.8), por último, permiten construir funciones abiertas tanto a derecha como a izquierda o construir curvas asimétricas.

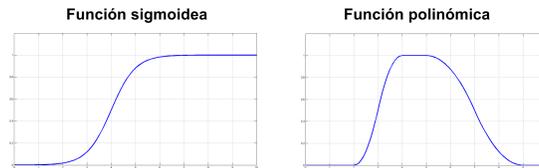


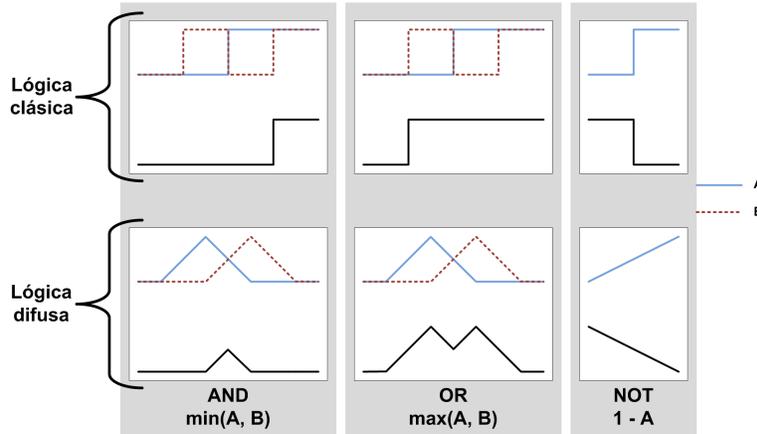
Figura 5.8. Funciones de pertenencia de tipo sigmoidea y polinómica

### 5.1.2. Operaciones lógicas difusas

En los conjuntos difusos se pueden realizar las mismas operaciones lógicas que en los conjuntos clásicos: *intersección*, *unión* y *negación*.

La operación de **intersección** entre dos conjuntos  $A$  y  $B$  se representa como  $A \text{ AND } B$ . Esta operación se realiza mediante unos operadores denominados *normas triangulares* o *t-normas* (Klement et al., 2000). Las t-normas son un conjunto de funciones matemáticas definidas en el rango  $[0, 1]$  ( $[0, 1] \times [0, 1] \rightarrow [0, 1]$ ). Comparten las propiedades de conmutatividad, monotonicidad y asociatividad y su elemento de identidad es el 1 (dado un valor  $x \in [0, 1]$ ,  $x \text{ AND } 1 = x$ ). Las t-normas más conocidas son la función de **mínimo** y la de **producto** (Wang, 1997). La función de **mínimo** consiste en la elección del valor más bajo entre todos los valores de entrada. La función de **producto**, por su lado, realiza una multiplicación o un escalado entre los valores de entrada. Existen otras muchas funciones de intersección, tales como *Sienes-Rescher*, *Mizumoto*, *Lukasiewicz*, *Dubois-Prade*, *Zadeh*, *Goguen*, *Gödel* e *implicación afilada*. Sin embargo, la explicación sobre estas funciones está fuera del alcance de la tesis. Se puede encontrar más información sobre las mismas en (Baturone et al., 2000). En la primera columna de la figura 5.9 se muestra la operación de intersección realizada mediante una función de mínimo tanto para un conjunto clásico (arriba) como para un conjunto difuso

(abajo). En ambos casos se muestra la intersección entre dos conjuntos denominados  $A$  y  $B$  y representados mediante una línea continua y una línea a trazos respectivamente y el resultado de la intersección (inmediatamente debajo de la intersección y representado mediante una línea continua de trazo grueso).



**Figura 5.9.** Operaciones lógicas  $AND$ ,  $OR$  y  $NOT$  para los conjuntos clásicos y difusos

La operación de **unión** entre dos conjuntos  $A$  y  $B$  se representa como  $A$   $OR$   $B$ . Los operadores para realizar esta operación se denominan *conormas triangulares*, *t-conormas* o *s-normas* (Klement et al., 2000). En las t-conormas el elemento de identidad es el 0 (dado un valor  $x \in [0, 1]$ ,  $x$   $OR$   $0 = x$ ). Entre las posibles funciones que se pueden utilizar, están las funciones de *máximo*, *suma probabilística* y *suma acotada* (Pedrycz y Gomide, 1998). Mientras que la función de **máximo** obtiene el valor máximo de entre todos los valores de entrada, la función de **suma probabilística** calcula la suma de los valores de entrada y le resta el producto de los valores de entrada y la **suma acotada** devuelve el mínimo entre la suma de los valores de entrada y la unidad. Existen otras funciones de unión, tales como la *suma normalizada* y *suma drástica*, pero su explicación está fuera del alcance de la tesis. Se puede obtener más información sobre ellas en (Grabisch et al., 2009). En la segunda columna de la figura 5.9 está representada la función de máximo para conjuntos clásicos (arriba) y conjuntos difusos (abajo). Para los dos casos se muestra la operación de unión entre los conjuntos  $A$  y  $B$  y el resultado de la operación.

La operación de **negación** de un conjunto  $A$  se representa como  $NOT$   $A$  y se utiliza para obtener todos los valores que no estén en un conjunto. Como la pertenencia se mide en el rango  $[0, 1]$  esta operación es equivalente a restar a la unidad la pertenencia de los valores de entrada. O lo que es lo mismo:  $NOT$   $A \equiv 1 - A$ . En la tercera columna de la figura 5.9 se muestra la representación de

esta función para un conjunto clásico (arriba) y un conjunto difuso (abajo). En ambos casos se ofrece la representación del conjunto original y del resultado.

## 5.2. Sistemas de inferencia difusa (Fuzzy inference systems)

En este apartado primero se detalla la *estructura* de los SID, para posteriormente explicar su *proceso de razonamiento* y mostrar dos variantes de los SID, los *sistemas Mamdani* y los *sistemas Takagi-Sugeno*, así como otras variantes más actuales como los *sistemas difuso relacionales*, los *sistemas de inferencia híbrida difuso-evidente*, los *sistemas de inferencia difusa con módulos de reglas de entrada única* y los *sistemas difusos probabilísticos*.

### 5.2.1. Estructura

Los SID están compuestos por una *base de conocimiento*, un *conjunto de hechos de entrada* y un *motor de inferencia*.

La **base de conocimiento** guarda las reglas sobre las que funciona el sistema. Los antecedentes y los consecuentes están formados por uno o más pares de variables difusas y valores lingüísticos. En los antecedentes, las variables expresadas son de entrada, mientras que en los consecuentes se indican variables de salida.

Por ejemplo, en la regla *if Temperatura del agua is Fría then Válvula de agua caliente is Abrir*, la *Temperatura del agua* y la *Válvula de agua caliente* son variables de entrada y salida respectivamente. El término *Fría* es un valor lingüístico de la variable *Temperatura del agua* y *Abrir* es un valor lingüístico de la variable *Válvula de agua caliente*. Esta regla refleja que cuando la temperatura del agua es fría la válvula del agua caliente debe abrirse.

En los SID los valores lingüísticos de las variables expresadas en los antecedentes y los consecuentes están representados mediante conjuntos difusos y los valores de entrada son valores numéricos en el universo de discurso de las variables de los antecedentes. En la figura 5.3 se muestran los conjuntos difusos relativos a los valores lingüísticos de la variable *Temperatura del agua*.

La base de conocimiento es escalable, esto es, se pueden añadir y modificar reglas de forma muy sencilla. Sin embargo, cuando hay muchas reglas, la gestión manual eficiente de la base de conocimiento no resulta escalable.

El **conjunto de hechos de entrada** es el conjunto de valores sobre el que se evalúan las reglas de la base de conocimiento. Cada uno de los hechos de entrada está formado por una variable de entrada y uno de los valores lingüísticos de esa variable.

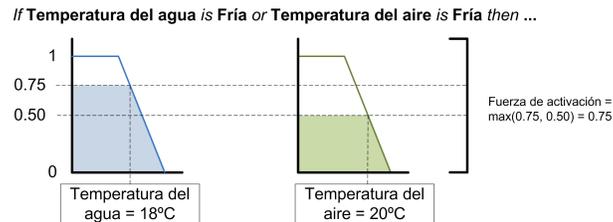
El **motor de inferencia** es el encargado de llevar a cabo el proceso de razonamiento del sistema basado en reglas.

### 5.2.2. Proceso de razonamiento

El proceso de razonamiento se denomina *proceso de inferencia* y consiste en obtener unos valores de salida numéricos a partir de unos valores de entrada numéricos. El proceso se divide en cinco pasos: *fuzzificación* de los valores de entrada, *aplicación de operadores lógicos* (paso no siempre necesario), *implicación*, *agregación* y *defuzzificación*.

La **fuzzificación** consiste en crear los hechos de entrada así como en calcular el grado de certeza de esos hechos. El grado de certeza se obtiene a partir de la pertenencia de los valores de entrada en los conjuntos difusos que componen las variables de los antecedentes. Tomando el ejemplo de la temperatura del agua mostrado en la sección 5.1.1 e ilustrado en la figura 5.3 un valor de entrada de 18°C crea los hechos *Temperatura del agua fría* y *Temperatura del agua templada* con unos grados de certeza de 0,75 y 0,25 respectivamente.

La **fuerza de activación** de una regla es la certeza con la que se activa, lo que quiere decir que las reglas de los SID se pueden activar aunque no se cumpla el antecedente completamente. La fuerza de activación viene dada por el grado de certeza del antecedente. Si el antecedente tiene una única variable, su grado de certeza será igual a la fuerza de activación. Teniendo en cuenta los valores del párrafo anterior, la fuerza de activación de la regla *if Temperatura del agua is Fría then ...* es de 0,75, ya que tiene una única variable en el antecedente. Sin embargo, si el antecedente tiene más de una variable, hay que realizar un paso de **aplicación de operadores lógicos** para calcular la fuerza de activación de la regla. Para ello se utilizan las operaciones lógicas de intersección, unión y negación explicadas en la sección 5.1.2. Por ejemplo, el antecedente de la regla *if Temperatura del agua is Fría or Temperatura del aire is Fría then ...* tiene dos variables unidas por una operación lógica de unión. Póngase que un valor de entrada para la temperatura del agua de 18°C y un valor de entrada para la temperatura del aire de 20°C obtienen unos grados de pertenencia de 0,75 y 0,50 en sus respectivos conjuntos difusos (ver figura 5.10). Utilizando una t-conorma de *máximo* para realizar la operación de unión entre las dos variables se obtiene una fuerza de activación de la regla de 0,75.



**Figura 5.10.** Ejemplo de aplicación de una operación de unión en un antecedente con dos variables

El paso de **implicación** consiste en identificar las reglas que se activan, esto es, aquéllas reglas cuyos antecedentes se cumplen y en calcular sus respectivos hechos de salida. Aparte de esto también se encarga de calcular el grado de certeza de esos hechos de salida. El grado de certeza del consecuente se calcula aplicando una t-norma entre el valor de la fuerza de activación de la regla y el propio conjunto difuso del consecuente. En la figura 5.11 se muestra el proceso de implicación de la regla *if Temperatura del agua is Fría then Válvula de agua caliente is Abrir*. El valor de entrada de  $18^{\circ}\text{C}$  para la temperatura del agua obtiene un grado de pertenencia de 0,75. Como el antecedente tiene una única variable (*Temperatura del agua*), no hay que aplicar ningún operador, por lo que la fuerza de activación de la regla es igual al grado de pertenencia del valor de entrada. En el paso de implicación se ha utilizado la función de *mínimo*, de forma que se ha comparado la fuerza de activación de la regla con todos los valores del conjunto difuso del consecuente. El resultado es la creación del hecho indicado en el consecuente (*Válvula de agua caliente abierta*) con un grado de certeza de 0.75, lo que gráficamente se refleja mediante un conjunto difuso truncado en la parte derecha de la figura 5.11.

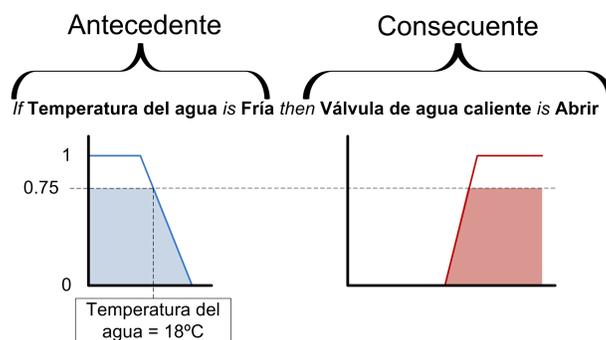


Figura 5.11. Proceso de implicación simple

El paso de **agregación** consiste en combinar para cada variable de salida los conjuntos difusos que describen los hechos de salida de las reglas activadas para esa variable de salida. El resultado es un hecho para cada variable de salida cuyo valor lingüístico está descrito mediante un único conjunto difuso.

Los valores de salida de los SID son valores numéricos en el universo de discurso de las variables de salida. En el paso de **desfuzzificación** o concreción se transforman los hechos agregados de salida en valores de salida. Supone el paso inverso al paso de fuzzificación.

### 5.2.3. Sistemas Mamdani

Estos sistemas fueron propuestos por primera vez por Mamdani y Assilian en 1975 como sistema de control de un motor de vapor (Mamdani y Assilian, 1975).

En los sistemas Mamdani el paso de agregación se realiza aplicando una t-conorma. En la figura 5.12 se muestra el paso de agregación correspondiente a dos reglas que afectan a la variable de salida *Válvula de agua caliente*. Esta variable tiene dos valores lingüísticos, [*Abrir*, *Cerrar*]. Mediante la regla superior (a la izquierda de la figura) se ha obtenido un grado de certeza de 0,75 en el hecho correspondiente al valor *Abrir*, mientras que con la activación de la regla inferior se ha obtenido un grado de certeza de 0,25 en el conjunto difuso del valor lingüístico *Cerrar*. En la parte central de la figura se puede observar la aplicación de la operación de unión, que en este caso ha sido la función de máximo. Por último, en la parte derecha de la figura 5.12 se puede ver el resultado de la agregación.

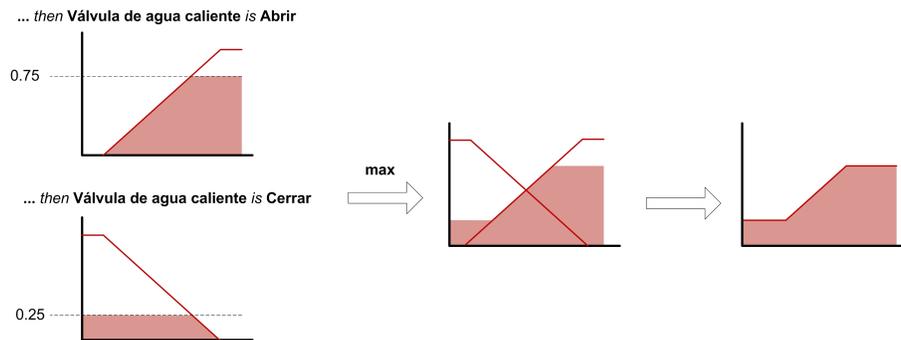


Figura 5.12. Proceso de agregación

La desfuzzificación se realiza utilizando diversos métodos, tales como el *cálculo del centroide*, el *cálculo del centro de área mayor*, el *bisector* y el *máximo* (Baturone et al., 2000). El **cálculo del centroide** devuelve el centro del área bajo la curva que forma la función que define el conjunto difuso del hecho de salida. El **cálculo del centro de área mayor**, por su lado, consiste en dividir el conjunto difuso en zonas convexas y calcular el centroide de la zona de mayor área. El **bisector** determina el valor que divide en dos partes iguales el área del conjunto difuso. Por último, el **máximo** consiste en escoger como valor representativo el valor numérico de salida cuyo grado de pertenencia sea máximo en el conjunto difuso del hecho de salida. Cuando hay muchos puntos con ese grado de pertenencia este método resulta ambiguo, de manera que normalmente se utiliza alguna variante como *primer máximo*, *último máximo*, *media de los máximos*, *mayor de los máximos* y *mínimo de los máximos*. Existen otras muchas funciones para realizar el paso de desfuzzi-

ficación, tales como *media difusa*, *media difusa ponderada*, *método de calidad*, *método de calidad gamma*, *etiqueta mayor* y *Takagi-Sugeno*. La explicación de estos métodos queda fuera del alcance de la tesis, pero se puede obtener más información sobre las funciones enumeradas en (van Leekwijck y Kerre, 1999).

En la figura 5.13 se muestra un SID de tipo Mamdani para regular la temperatura del agua de un recipiente y mantenerla siempre en una temperatura templada. El SID tiene dos variables de entrada, *Temperatura del agua* y *Temperatura del aire*, las cuales se pueden ver en la parte inferior izquierda de la imagen junto con sus valores numéricos de entrada. La primera de ellas introduce en el SID la temperatura del agua del recipiente en grados centígrados, mientras que la segunda introduce la temperatura del aire alrededor del recipiente, también en grados centígrados. El universo de discurso de la temperatura del agua se define en el rango  $[0, 100]$  y tiene tres valores lingüísticos,  $[Fría, Templada, Caliente]$ . Mientras tanto, el universo de discurso de la temperatura del aire se define en el rango  $[-10, 50]$  y la variable tiene los valores lingüísticos  $[Fría, Caliente]$ . Los valores lingüísticos de las dos variables de entrada están descritas mediante unos conjuntos difusos definidos utilizando FP trapezoidales.

El SID también tiene dos variables de salida, *Válvula del agua caliente* y *Válvula del agua fría*. Ambas se pueden observar en la parte inferior derecha de la figura con sus correspondientes valores numéricos de salida. Cada una de las variables de salida controla una válvula de entrada de agua en el recipiente. Una de las válvulas controla la entrada de agua fría, mientras que la otra controla la entrada de agua caliente en el recipiente. El universo de discurso de ambas variables se define en el rango  $[0, 1]$  y se divide en los valores lingüísticos  $[Cerrar, Abrir]$ , representando el 0 que la válvula está completamente cerrada y 1 que está completamente abierta. Los valores lingüísticos de estas variables, al igual que los valores lingüísticos de las variables de entrada, están descritos mediante conjuntos difusos definidos utilizando FP trapezoidales.

El SID dispone de tres reglas para regular el agua del recipiente (mostradas de forma horizontal en la figura 5.13). Cada regla tiene su interpretación gráfica encima. Por ejemplo, la primera regla es **if** *Temperatura del agua is Fría or Temperatura del aire is Fría then* *Válvula de agua caliente is Abrir and Válvula de agua fría is Cerrar*. Encima de la regla se muestran los conjuntos difusos correspondientes a los valores lingüísticos de los antecedentes y los consecuentes de las reglas, esto es, los conjuntos correspondientes a los valores *Fría* y *Fría* en los antecedentes y a los valores *Abrir* y *Cerrar* en los consecuentes.

La primera y la tercera regla tienen dos antecedentes que están unidos mediante un operador de unión (*OR*). La segunda regla, por su lado, tiene un único antecedente.

Los valores de entrada de las temperaturas del agua y el aire son 18°C y 20°C respectivamente. El proceso de fuzzificación se refleja en la figura 5.13 mediante una flecha de abajo hacia arriba a la izquierda de la figura y unas

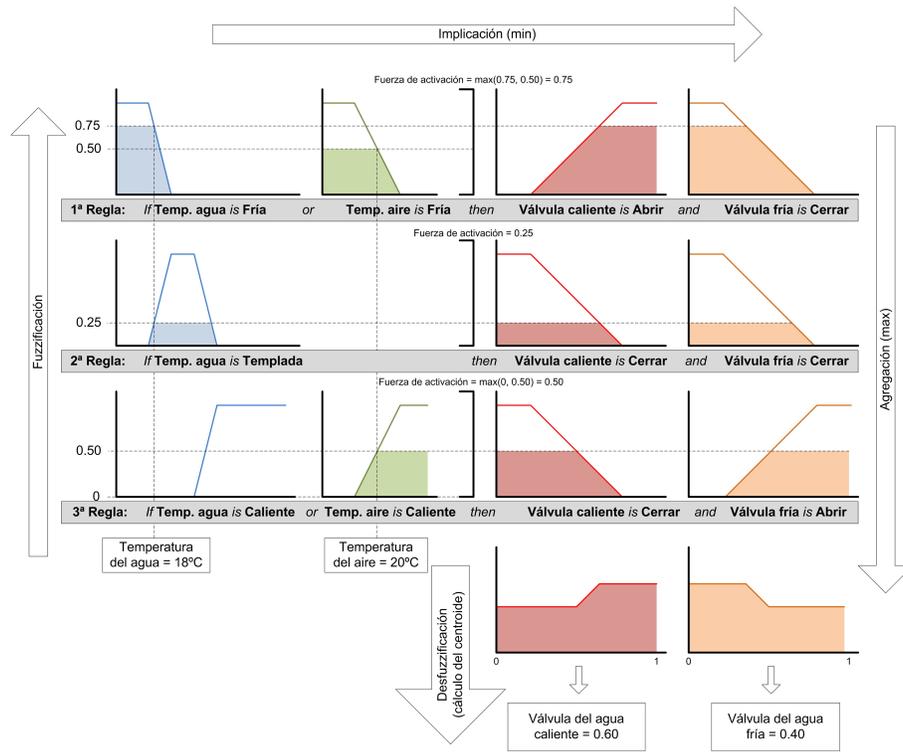


Figura 5.13. Proceso de inferencia de un SID de tipo Mamdani

líneas no continuas que nacen de los valores de entrada hacia los conjuntos difusos de los antecedentes de las reglas. Como resultado de la fuzzificación se obtienen cuatro hechos de entrada: *Temperatura de agua fría* con un grado de certeza de 0,75, *Temperatura de agua templada* con un grado de certeza de 0,25, *Temperatura de aire fría* con un grado de certeza de 0,50 y *Temperatura de aire caliente* con un grado de certeza de 0,50. Con estos hechos se activan las tres reglas del SID.

Para calcular la fuerza de activación de la primera y la tercera regla se utiliza la función t-conorma de *máximo*. Esto da como resultado una fuerza de activación de 0,75 para la primera regla ( $\max(0,75, 0,50) = 0,75$ ) y de 0,50 para la tercera regla ( $\max(0,0, 0,50) = 0,50$ ). La fuerza de activación de la segunda regla, por su lado, es igual a la pertenencia del valor de entrada en el conjunto difuso del antecedente, esto es, 0,25.

La implicación se realiza mediante la función t-norma de *mínimo*. El resultado de este paso es la creación de seis hechos de salida, uno por cada consecuente de las reglas activadas. Los grados de certeza son el resultados de aplicar la función de mínimo entre la fuerza de aplicación de las reglas y los conjuntos difusos de los consecuentes. El paso de implicación gráficamente se

refleja mediante la flecha en la parte de arriba de la figura 5.13 y el resultado se refleja en los trapecios truncados sobre los consecuentes de las reglas.

El paso de agregación se refleja mediante una flecha de arriba hacia abajo a la derecha de la figura 5.13. Este paso aplica la función de *máximo* sobre todos los hechos de salida correspondientes a las mismas variables de salida. Esto da como resultado dos hechos representados mediante los dos conjuntos difusos agregados en la parte inferior derecha de la figura 5.13 (uno para cada una de las variables de salida).

Por último, el paso de desfuzzificación (reflejado mediante una flecha en la parte central inferior de la figura 5.13) aplica la función del *cálculo del centroide* para obtener los dos valores de salida numéricos correspondientes a los hechos de salida, 0,60 para la variable *Válvula del agua caliente* y 0,40 para la variable de salida *Válvula del agua fría*. Esto, a efectos prácticos indica que hay que abrir ligeramente la válvula del agua calientes, mientras que hay que cerrar ligeramente la válvula del agua fría.

#### 5.2.4. Sistemas Takagi-Sugeno

Los sistemas Takagi-Sugeno se presentaron por primera vez en el año 1985 como solución en una planta de procesamiento de agua y en un proceso de creación de acero (Takagi y Sugeno, 1985).

En estos sistemas los valores lingüísticos de los consecuentes están descritos mediante conjuntos difusos expresados usando únicamente FP lineales o constantes. Por ejemplo, una regla de dos antecedentes en un sistema de este tipo tiene la forma de **If** *Entrada1* =  $x$  **and** *Entrada2* =  $y$  **then** *Salida* **is**  $z = ax + by + c$ , donde  $a$ ,  $b$  y  $c$  son valores constantes.

Cuando los valores de  $a$  y  $b$  son nulos ( $a = b = 0$ ) el sistema se denomina Takagi-Sugeno de *orden-cero* y el consecuente sigue una función constante dada por el valor de  $c$ . En cualquier otro caso ( $a \neq 0$  ó  $b \neq 0$ ), el consecuente de ese sistema Takagi-Sugeno sigue una función lineal.

En los sistemas de Takagi-Sugeno los pasos de agregación y desfuzzificación están unificados y el valor de salida numérico correspondiente a una variable de salida se calcula utilizando una suma ponderada utilizando la ecuación

$$\text{Variable salida} = \frac{\sum_{i=0}^N w_i z_i}{\sum_{i=0}^N w_i} \quad (5.1)$$

donde  $N$  es el número de reglas del sistema que afectan a la variable de salida,  $w_i$  es la fuerza de activación de la regla  $i$  y  $z_i$  es el resultado de la función de salida de la regla  $i$ .

En la figura 5.14 se muestra un SID de tipo Takagi-Sugeno de *Orden-cero* para regular la temperatura del agua de un recipiente. Las variables de entrada son la *Temperatura del agua* y la *Temperatura del aire*, las mismas que se han utilizado en el ejemplo del apartado anterior. Las variables de salida

son la *Válvula del agua caliente* y *Válvula del agua fría*, y al igual que en el ejemplo del apartado anterior, tienen dos valores lingüísticos [*Cerrar*, *Abrir*] y están definidas en el rango [0,1]. Sin embargo, sus valores lingüísticos no están descritos mediante conjuntos difusos trapezoidales, sino constantes. Así, el valor lingüístico *Cerrar* se describe mediante la función  $z = 0$  y el valor *Abrir* se describe mediante la función  $z = 1$ .

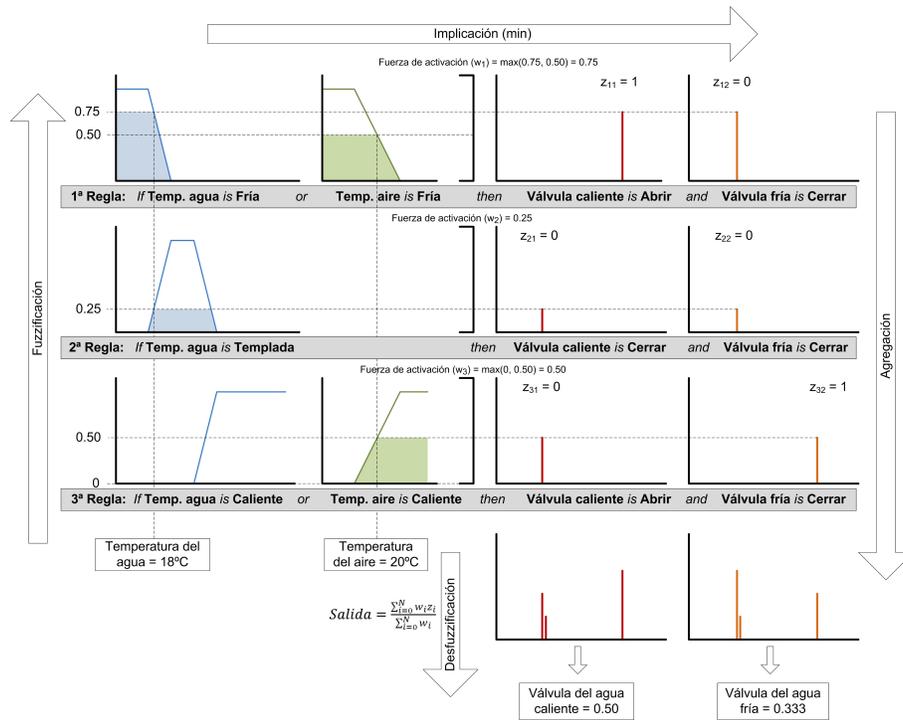


Figura 5.14. Proceso de inferencia de un SID de tipo Takagi-Sugeno

Las reglas del SID son las del ejemplo de la sección anterior. Sin embargo, gráficamente difieren, ya que los consecuentes de las reglas del SID de la figura 5.14 están representados mediante unas líneas verticales en los puntos  $z = 0$  y  $z = 1$ .

Los pasos de fuzzificación, aplicación de operadores lógicos e implicación se realizan de la misma forma que en el ejemplo de la sección anterior. El cambio se produce en los pasos de agregación y defuzzificación, que en el SID de la figura 5.14 se resuelven utilizando la ecuación 5.1. En este ejemplo, las tres reglas del SID afectan a las dos variables de salida, por lo que  $N = 3$ .

$$\begin{aligned}
 \text{Válvula caliente} &= \frac{w_1 z_{11} + w_2 z_{21} + w_3 z_{31}}{w_1 + w_2 + w_3} \\
 &= \frac{0,75 \cdot 1 + 0,25 \cdot 0 + 0,5 \cdot 0}{0,75 + 0,25 + 0,5} = 0,5
 \end{aligned} \tag{5.2}$$

$$\begin{aligned}
 \text{Válvula fría} &= \frac{w_1 z_{12} + w_2 z_{22} + w_3 z_{32}}{w_1 + w_2 + w_3} \\
 &= \frac{0,75 \cdot 0 + 0,25 \cdot 0 + 0,5 \cdot 1}{0,75 + 0,25 + 0,5} = 0,333
 \end{aligned} \tag{5.3}$$

Así, para calcular el valor de salida de la variable *Válvula del agua caliente* se tienen en cuenta las fuerzas de activación y los consecuentes de las tres reglas que afectan a esta variable. Así, las fuerzas de activación son 0,75, 0,25 y 0,50 y los valores de los consecuentes 1, 0 y 0 respectivamente.

Con estos datos se obtienen unos resultados de 0,50 y 0,333 para las variables de salida *Válvula de agua caliente* y *Válvula de agua fría* respectivamente. Esto significa que hay que dejar la válvula de agua caliente en un punto intermedio entre abierta y cerrada y hay que cerrar ligeramente la válvula de agua fría.

### 5.2.5. Otros SID

Además de los SID de tipo Mamdani y Takagi-Sugeno, con el tiempo han ido surgiendo otras variantes más actuales de los SID, las cuales añaden nuevas características o modifican algunas existentes, todo ello con la intención de mejorar la eficiencia. A continuación se presentan algunas de ellas:

#### 5.2.5.1. Clasificadores difuso relacionales (Fuzzy relational classifiers)

Estos clasificadores no utilizan reglas de tipo *if-then* (Kuncheva, 2000). El paso de implicación se realiza utilizando una matriz de pesos que relaciona las variables de entrada con las de salida. Este proceso se lleva a cabo utilizando una suma ponderada. Por lo demás, los clasificadores difuso relacionales sí disponen de pasos de fuzzificación, aplicación de operadores, agregación y desfuzzificación.

#### 5.2.5.2. Sistemas de inferencia híbrida difuso-evidente (Fuzzy-evidential hybrid inference systems)

Los sistemas de inferencia híbrida difuso-evidente son una combinación de los SID con la teoría de evidencia Dempster-Shafer (Khatibi y Montazer, 2010). Esta teoría indica que todo conocimiento es impreciso y que el nivel de certeza del conocimiento se puede medir en grados de *creencia* y *verosimilitud*

(Dempster, 1967). La **creencia** es el grado de certeza real y total, mientras que la **verosimilitud** incluye también el concepto de posibilidad (aunque sea con baja probabilidad). El nivel de certeza se expresa en forma de probabilidad, cuyo límite inferior y superior son el grado de creencia y de verosimilitud respectivamente.

El funcionamiento de un sistema de inferencia híbrida difuso-evidente se divide en dos fases. En la primera de ellas se fuzzifican unos valores de entrada y los hechos de entrada resultantes activan una serie de reglas. En la segunda fase se evalúa el nivel de certeza de los resultados de las reglas activadas utilizando la creencia y verosimilitud.

#### 5.2.5.3. Sistemas de inferencia difusa con módulos de reglas de entrada única (Single input rule modules connected fuzzy inference model)

Estos sistemas tienen dos características principales (Yubazaki et al., 1997). Por un lado, todas las reglas del sistema tienen antecedentes con una única variable. Esto hace que sea innecesario disponer de un paso de aplicación de operadores lógicos ya que la fuerza de activación de las reglas es igual a la pertenencia de los valores de entrada en los conjuntos difusos de los valores lingüísticos expresados en los antecedentes. Por otro lado, cada variable de entrada tiene un peso en el rango  $[0, 1]$  respecto a cada una de las variables de salida. La suma de todos los pesos para una variable de salida es la unidad. De esta forma se permite premiar o perjudicar a aquellas variables de entrada que sean más importantes o menos importantes respectivamente en relación a las variables de salida.

#### 5.2.5.4. Sistemas difusos probabilísticos (Probabilistic fuzzy systems)

Los sistemas difusos probabilísticos son SID donde los antecedentes se representan mediante conjuntos difusos y los consecuentes mediante distribuciones probabilísticas (van den Berg et al., 2002). Cada una de las reglas activa un consecuente con una probabilidad determinada. La probabilidad total de una salida se calcula siguiendo una suma ponderada de todas las probabilidades de esa salida.

### 5.3. Frameworks para el desarrollo de Sistemas de Inferencia Difusa

Existen dos corrientes para el desarrollo de SID (Gacto et al., 2011): el *Modelado Difuso Lingüístico (MDL)* y el *Modelado Difuso Preciso (MDP)*. El objetivo del **MDL** es conseguir modelos que se centren más en la interpretabilidad del sistema y de los resultados que en la precisión o eficiencia.

En el **MDP**, por el contrario, se pretende conseguir una mayor efectividad, aún a expensas de la interpretabilidad. Entre los sistemas que se construyen siguiendo el MDL están aquéllos que se construyen manualmente a partir de la información de expertos y de la bibliografía, mientras que normalmente, los sistemas que siguen el MDP suelen estar contruidos automáticamente a partir de datos utilizando algoritmos de aprendizaje automático.

Existen *frameworks* y herramientas para facilitar el trabajo de los desarrolladores sigan la corriente que sigan para la construcción de SID. Sin embargo, los *frameworks* para el desarrollo de SID siguiendo la metodología MDP, tales como FisPro (Guillaume y Charnomordic, 2011), están fuera del alcance de esta tesis. A continuación se explica brevemente el proceso de desarrollo de SID siguiendo la metodología MDL y utilizando *frameworks* específicos.

La definición de un SID se puede dividir en 5 pasos: primero hay que *definir las variables de entrada y salida*, posteriormente hay que *especificar las funciones de implicación, agregación y desfuzzificación*, en el siguiente paso hay que *definir las reglas*, después hay que *simular* o probar el prototipo y finalmente hay que *exportar* el sistema resultante para que sea integrado en una solución final.

La **definición de las variables de entrada y salida** es un proceso muy ligado a los tipos de SID que permiten diseñar los *frameworks*, así como a los tipos de FP que se pueden utilizar para definir las variables. Dependiendo de la complejidad de las variables que se quieran diseñar (número de valores lingüísticos, FP de cada uno de los valores, puntos de cruce de las FP...) el funcionamiento del sistema resultante puede no ser el esperado. En cualquier caso, la definición de las variables es un proceso con bastante peso matemático. Aunque los *frameworks* permiten la definición manual de variables y FPs, también suelen ofrecer herramientas específicas que facilitan la labor de los desarrolladores interpretando gráficamente las variables que se van definiendo.

El paso de **especificación de las funciones de implicación, agregación y desfuzzificación** no suele revestir una mayor complejidad, ya que suele consistir en la selección de las funciones a utilizar dentro del abanico de funciones que provee cada *framework*.

La **definición de las reglas** de un SID es probablemente el trabajo más costoso y complejo del diseño del SID. Si se opta por establecer reglas sencillas el número de éstas puede aumentar de forma exponencial en base al número de variables de entrada y salida. Si se opta por utilizar un número mínimo de reglas, la complejidad de éstas aumentará si aumenta el número de variables. Tanto por cantidad como por complejidad es necesario el uso de alguna herramienta que facilite la definición y revisión de las reglas. Los *frameworks* para la construcción de SID normalmente incluyen funcionalidades específicas para la definición de reglas que permiten especificar éstas mediante tablas, grafos o formularios.

Una vez construido el SID hay que **simular** y comprobar que su funcionamiento sea el que se espera de él. La mayoría de *frameworks* incluyen funcionalidades para realizar esta comprobación, aunque pueden diferir en có-

mo la ofrecen. Así, los hay que permiten probar el SID en sí mismo, mientras que hay otros que permiten probar el SID como parte de un sistema más complejo. En el caso de que el SID no tenga el comportamiento deseado habrá que modificar ciertos aspectos del diseño y volver a realizar pruebas.

Cuando ya se ha comprobado que el SID tiene el comportamiento adecuado, hay que **exportarlo** de forma que pueda ser integrado en una solución final. Las variantes que ofrecen los *frameworks* para esta tarea son múltiples, desde soluciones autoejecutables hasta módulos integrables en otros sistemas.

Por último, y dado que los SID construidos siguiendo la metodología MDL a veces adolecen de una eficiencia menor que los SID creados siguiendo el MDP, bastantes *frameworks* incluyen *sistemas de aprendizaje automático* para la mejora de ciertas características de los SID diseñados.

## Entrenamiento de SID mediante Algoritmos Genéticos

Un SID debe ofrecer una buena efectividad al mismo tiempo que ser interpretable idealmente. Sin embargo, estos objetivos son contradictorios (Gacto et al., 2011). Un método para conseguir que los SID sean efectivos e interpretables consiste en diseñar los SID manualmente a partir de la información de los expertos o de la bibliografía para después afinar su funcionamiento utilizando algún sistema de aprendizaje automático. Para afinar el comportamiento de los SID se puede hacer uso de los algoritmos evolutivos (Herrera, 2008) y más concretamente, de los algoritmos genéticos (AG).

Los AG son métodos de búsqueda de soluciones basados en la evolución por selección natural (Holland, 1992). Los AG están inspirados en técnicas de la genética, como el cruce de individuos, la herencia y las mutaciones. Se da prioridad o mayor importancia a las mejores soluciones. Para ello, cada solución tiene un valor de *aptitud* o *fitness* que mide su calidad. Cuanto más alto sea este valor, más probable será que esta solución sobreviva.

Los individuos que forman la población son *cromosomas*. Cada cromosoma tiene un *gen* por cada parámetro a entrenar. Los cromosomas están codificados mediante un código numérico, de forma que a cada gen le corresponde un dígito. Lo más común y sencillo es que los cromosomas sean binarios, esto es, que los genes sólo puedan tener valores de 0 y 1. Sin embargo, cuando los AG se aplican a problemas numéricos reales de alta precisión surgen problemas de computabilidad (Michalewicz, 1998). Al fin y al cabo, cada uno de los individuos estará formado por una cadena binaria demasiado larga. Por ejemplo, el estándar 754 del IEEE establece una longitud de 32 bits para representar un número real. Suponiendo que haya que entrenar 50 parámetros reales, los cromosomas tendrán una longitud de 1600 dígitos. Para solucionar esto existen AG específicos para números reales (*Real-Coded Genetic Algorithms*), en los cuales, como su mismo nombre indica, los genes se especifican mediante números reales.

La búsqueda de nuevos cromosomas se realiza sobre un *espacio de búsqueda* que incluye todas las soluciones posibles para el problema. El espacio de búsqueda se recorre utilizando procesos de *exploración* y *explotación*. La

**exploración** se refiere a la tarea de examinar eficientemente toda región del espacio de búsqueda y encontrar nuevas soluciones potencialmente mejores. Por otro lado, la **explotación** es el proceso de mejorar y combinar los rasgos de las mejores soluciones encontradas durante el proceso de exploración, con el objetivo de encontrar soluciones de mayor calidad. En los AG la explotación se realiza mediante la operación de *cruce*, mientras que la exploración se realiza mediante la operación de *mutación*. Mediante la operación de **cruce** se combinan dos o más cromosomas de una población para crear nuevos cromosomas que mantengan ciertas características de los cromosomas progenitores (lo que se puede considerar herencia). La operación de **mutación**, por su lado, consiste en realizar pequeñas variaciones en algunos cromosomas de la población de forma que se puedan introducir nuevas soluciones no alcanzables mediante operaciones de explotación como el cruce. De esta forma, se puede evitar caer en mínimos locales.

En el extracto 6.1 se muestra el pseudocódigo de un AG. El primer paso consiste en establecer unas probabilidades para las operaciones de cruce y mutación ( $Prob_c$  y  $Prob_m$  en la línea 1), así como definir una función de aptitud que permita medir la calidad de los cromosomas (línea 2). Primero se crea una población de cromosomas de forma aleatoria (línea 4) y se calcula la aptitud de todos ellos (línea 5). A partir de este punto se entra en un ciclo hasta que se llegue a una *condición de finalización*. Entre las condiciones de finalización más extendidas están el que se haya creado un número determinado de generaciones, el haber llegado a una solución muy cercana a la óptima y la no mejoría de la aptitud de la población durante un número determinado de generaciones. En el caso de utilizar una *aproximación elitista*, la cual permite que el mejor o los mejores individuos se mantengan inalterables en las nuevas poblaciones que se generan de forma que no se pierda su información, otra condición de finalización es la no mejoría de la aptitud del mejor individuo de la población durante un número determinado de generaciones. En cada generación se eligen los cromosomas más aptos (con valor de aptitud más alto) para cruzarse utilizando la probabilidad indicada al inicio del algoritmo (líneas 8 y 9). De esta forma se introducen nuevos cromosomas en la población (línea 10). Posteriormente se aplican mutaciones en los cromosomas de la población la probabilidad establecida al comienzo del algoritmo (línea 11). El siguiente paso consiste en volver a calcular la aptitud de los cromosomas de la población (línea 12). Por último, y como la población debe mantener un número constante de cromosomas, se eliminan aquéllos que sean menos aptos (línea 13). Llegados a este punto, se vuelven a evaluar las condiciones de finalización y si no se cumplen, se vuelve a ejecutar el ciclo.

```

1: Definir los parámetros Prob_c y Prob_m;
2: Definir la función de aptitud;
3: t = 0;
4: Crear la población inicial P(0);
5: Evaluar la aptitud de los cromosomas de la
   población P(0);

```

```

6: Hasta que no se llegue a una condición de
   finalización:
7: {
8:   P_1(t) = Seleccionar cromosomas para la operación
   de cruce en P(t);
9:   P_2(t) = Operación de cruce sobre P_1(t) con la
   probabilidad Prob_c;
10:  P_3(t) = Introducir nuevos cromosomas P_2(t) en la
   población P(t);
11:  P_4(t) = Operación de mutación sobre la población
   P_3(t) con la probabilidad Prob_m;
12:  Evaluar la aptitud de los cromosomas de la
   población P_4(t);
13:  P(t + 1) = Eliminar los cromosomas menos aptos de la
   población P_4(t);
14:  t = t + 1;
15: }

```

**Extracto 6.1.** Pseudocódigo de un algoritmo genético

Existen diversas formas de utilizar los AG para mejorar las prestaciones de los SID (Herrera, 2008). Por un lado, los AG se pueden utilizar como método para la creación automática de SID a partir de datos. Por otro lado, se pueden utilizar para afinar distintas características de los SID. En este último aspecto destacan tres corrientes. La primera de ellas consiste en el afinamiento de los parámetros necesarios para definir las FP que describen los conjuntos difusos del SID, sin modificar las reglas del sistema (Herrera et al., 1995) (Karr, 1991). La segunda corriente consiste en adaptar los parámetros del proceso de inferencia, sin modificar el conocimiento del SID (Crockett et al., 2006). Por último, la tercera línea engloba los métodos de desfuzzificación adaptados mediante AG (Kim et al., 2002).

En el conocimiento médico normalmente la incertidumbre no viene marcada por reglas generales del tipo "*si el paciente tiene fiebre alta, puede tener gripe*", sino por cómo cuantificar la "*fiebre alta*". Por lo tanto, en los SID creados a partir de información bibliográfica y de expertos el afinamiento de los SID se debe dar en relación a los valores de las FP que describen los conjuntos difusos. Por ello, en adelante se detalla la primera de las corrientes mencionadas anteriormente. De la misma forma, el entrenamiento se centra en los AG para números reales, ya que se prevé intratable el entrenamiento de variables médicas representadas mediante cromosomas binarios.

A continuación se ahonda más en las operaciones de cruce y mutación y se muestran varios métodos para realizar dichas operaciones.

## 6.1. Operación de cruce

Antes de poder realizar la operación de cruce hay que seleccionar un número determinado de cromosomas de la población para utilizarlos en dicha

operación. Existen múltiples métodos para seleccionar cromosomas de una población (Blickle y Thiele, 1995) y en ocasiones se utiliza una combinación de varios de ellos. A continuación se explican algunos de los métodos de selección:

- **Selección de ruleta (Roulette-wheel selection):** Este método también es conocido como *muestreo estocástico con reemplazo* (Baker, 1987). Consiste en seleccionar aleatoriamente un número determinado de cromosomas a cruzar en base a una probabilidad obtenida a partir de la aptitud de los mismos (a mayor aptitud mayor probabilidad). El problema de la selección de ruleta es que un mismo cromosoma puede ser seleccionado varias veces aunque no sea el más apto.
- **Muestreo estocástico universal (Stochastic universal sampling):** Este método consiste en asignar probabilidades a los cromosomas de una población en base a su aptitud y seleccionar un número determinado de ellos en base a un único intervalo aleatorio (Baker, 1987). Este método evita que se seleccione un cromosoma con baja aptitud varias veces.
- **Selección por torneo (Tournament selection):** Este método consiste en realizar competiciones entre conjuntos aleatorios de cromosomas, donde el cromosoma con mejor aptitud es el ganador (Goldberg y Deb, 1991). Esto se consigue seleccionando tantos conjuntos aleatorios de la población como cromosomas se quieran utilizar en el cruce. Cuanto mayor sea el tamaño del conjunto seleccionado más probable será que se seleccionen para cruzar los mejores individuos. Por el contrario, seleccionar conjuntos de un único cromosoma equivale a realizar una selección aleatoria.
- **Selección por truncamiento (Truncation selection):** Este método consiste en seleccionar un porcentaje de los mejores cromosomas y posteriormente hacer una selección aleatoria en esa selección. Este método se considera artificial, ya que pierde parte del componente de aleatoriedad existente en la selección natural, debido a que los cromosomas menos aptos nunca tienen la opción de reproducirse. Este método se suele usar cuando las poblaciones son muy grandes y se necesita seleccionar grandes conjuntos de cromosomas para cruzar.

Existen múltiples algoritmos para realizar la operación de cruce entre cromosomas de números reales (Herrera et al., 2003). Dependiendo del método pueden haber diferentes grados de explotación. En adelante, se asume que los cromosomas a cruzar son  $C_1 = (c_1^1, c_2^1, c_i^1, \dots, c_n^1)$  y  $C_2 = (c_1^2, c_2^2, c_i^2, \dots, c_n^2)$ , donde  $c_i^1$  se refiere al gen  $i$  del cromosoma  $C_1$  y  $n$  es el número de genes de los cromosomas. Así, se establece que  $c_i^1, c_i^2 \in [a_i, b_i]$ , siendo  $a_i$  y  $b_i$  los límites mínimo y máximo respectivamente del espacio de búsqueda de los genes  $c_i^1$  y  $c_i^2$ . Ahora se pueden definir  $\alpha_i = \min(c_i^1, c_i^2)$  y  $\beta_i = \max(c_i^1, c_i^2)$ . De esta forma, en el intervalo  $[a_i, b_i]$  se diferencian tres intervalos  $[a_i, \alpha_i]$ ,  $[\alpha_i, \beta_i]$  y  $[\beta_i, b_i]$ , los cuales delimitan las regiones de exploración y explotación (ver figura 6.1).

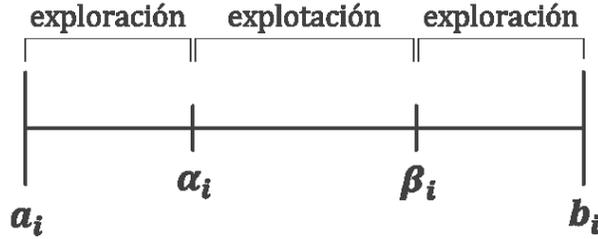


Figura 6.1. Relación entre los métodos de búsqueda y el espacio de búsqueda

A continuación se detallan los métodos de cruce que se han utilizado en la presente tesis.

### 6.1.1. Cruce simple (Simple crossover)

Consiste en seleccionar un punto  $i \in \{1, 2, \dots, n - 1\}$  de forma aleatoria. Posteriormente se cortan los dos cromosomas por ese punto y se intercambian las secciones (Goldberg, 1989). De esta forma, los dos nuevos cromosomas tienen la siguiente estructura:

$$H_1 = (c_1^1, c_2^1, \dots, c_i^1, c_{i+1}^2, \dots, c_n^2) \quad (6.1)$$

$$H_2 = (c_1^2, c_2^2, \dots, c_i^2, c_{i+1}^1, \dots, c_n^1) \quad (6.2)$$

### 6.1.2. Cruce Max-Min-Aritmético (Max-Min-Arithmetic crossover)

En este caso se crean cuatro cromosomas  $H_k = (h_1^k, h_2^k, \dots, h_i^k, \dots, h_n^k)$ , donde  $k = 1, 2, 3, 4$ . Los genes se crean siguiendo las siguientes expresiones:

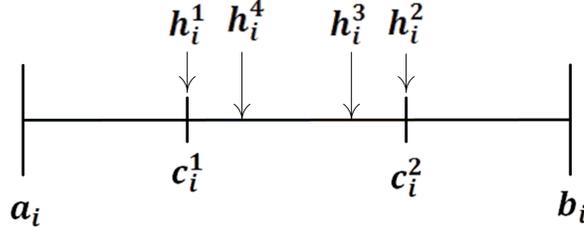
$$h_i^1 = \min(c_i^1, c_i^2) \quad (6.3)$$

$$h_i^2 = \max(c_i^1, c_i^2) \quad (6.4)$$

$$h_i^3 = \lambda \cdot c_i^1 + (1 - \lambda) \cdot c_i^2 \quad (6.5)$$

$$h_i^4 = \lambda \cdot c_i^2 + (1 - \lambda) \cdot c_i^1 \quad (6.6)$$

donde  $\lambda$  es un número elegido por el usuario en el rango  $\in [0, 1]$ . De los cuatro nuevos cromosomas se eligen los dos más aptos para introducirlos en la población (Herrera et al., 1997). En la figura 6.2 se muestra gráficamente la posición de los cuatro genes creados, cuando  $c_i^1 < c_i^2$ .



**Figura 6.2.** Posición de  $h_i^1$ ,  $h_i^2$ ,  $h_i^3$  y  $h_i^4$  mediante un cruce Max-Min-Aritmético cuando  $c_i^1 < c_i^2$  y  $\lambda = 0,25$

### 6.1.3. Cruce dinámico (Dynamic crossover)

Cada cruce dinámico de los cromosomas  $C_1$  y  $C_2$  produce cuatro nuevos cromosomas  $H_k = (h_1^k, h_2^k, \dots, h_i^k, \dots, h_n^k)$ , donde  $k = 1, 2, 3, 4$ . Cada uno de los nuevos cromosomas  $H_k$  se crea siguiendo una de las familias de funciones  $\mathcal{F}(t)$ ,  $\mathcal{S}(t)$ ,  $\mathcal{M}^+(t)$  y  $\mathcal{M}^-(t)$ , siendo  $t \in [1, g_{max}]$  y  $g_{max}$  el número máximo de generaciones del algoritmo (Herrera et al., 1996). De estos cuatro, se eligen los dos mejores para introducirse en la población.

Las familias de funciones anteriores se definen de la siguiente forma:  $\mathcal{F} = (F^1, \dots, F^t, \dots, F^{g_{max}})$ ,  $\mathcal{S} = (S^1, \dots, S^t, \dots, S^{g_{max}})$  y  $\mathcal{M} = (M^1, \dots, M^t, \dots, M^{g_{max}})$ . Para la generación  $t$ , siendo  $1 \leq t \leq g_{max} - 1$  y los dos genes a cruzar  $c_i^1$  y  $c_i^2$ , esas familias de funciones cumplen las siguientes propiedades:

$$\begin{aligned} \forall c_i^1, c_i^2 \in [a_i, b_i] \quad & F^t(c_i^1, c_i^2) \leq F^{t+1}(c_i^1, c_i^2) \\ & \text{y } F^{g_{max}}(c_i^1, c_i^2) \approx \min(c_i^1, c_i^2) \end{aligned} \quad (6.7)$$

$$\begin{aligned} \forall c_i^1, c_i^2 \in [a_i, b_i] \quad & S^t(c_i^1, c_i^2) \geq S^{t+1}(c_i^1, c_i^2) \\ & \text{y } S^{g_{max}}(c_i^1, c_i^2) \approx \max(c_i^1, c_i^2) \end{aligned} \quad (6.8)$$

$$\begin{aligned} \forall c_i^1, c_i^2 \in [a_i, b_i] \quad & M^t(c_i^1, c_i^2) \geq M^{t+1}(c_i^1, c_i^2) \\ & \text{o } M^t(c_i^1, c_i^2) \leq M^{t+1}(c_i^1, c_i^2) \\ & \text{y } M^{g_{max}}(c_i^1, c_i^2) \approx M_{lim}(c_i^1, c_i^2) \\ & \text{siendo } M_{lim} = \frac{c_i^1 + c_i^2}{2} \end{aligned} \quad (6.9)$$

Dado que la propiedad de la expresión 6.9 es disyuntiva, la familia de funciones  $\mathcal{M}$  se puede dividir en dos familias de funciones diferentes, siendo  $\mathcal{M}^+$  la que cumple la primera de las propiedades y  $\mathcal{M}^-$  la que cumple la segunda.

Para la construcción de las familias de funciones  $\mathcal{F}$  y  $\mathcal{S}$  se suelen utilizar las variaciones Dubois de las funciones *t-norma*  $T^q$  y *t-conorma*  $G^q$ , respectivamente (Dubois y Prade, 1980). Estas variaciones tienen la siguiente forma:

$$T^q(x, y) = \frac{x \cdot y}{x \vee y \vee q}, \text{ siendo } x, y \in [0, 1] \text{ y } 0 \leq q \leq 1 \quad (6.10)$$

$$G^q(x, y) = 1 - \frac{(1-x) \cdot (1-y)}{(1-x) \vee (1-y) \vee q}, \text{ siendo } x, y \in [0, 1] \text{ y } 0 \leq q \leq 1 \quad (6.11)$$

Estas funciones están definidas en el intervalo  $[0, 1]$ , mientras que los genes están definidos en el rango  $[a, b]$ . Por ello, antes de cruzar los genes hay que transformarlos al intervalo  $[0, 1]$ , y después del cálculo habrá que volver a transformarlos a su intervalo real. Por otro lado, también se necesita una función  $\delta(\cdot)$  para transformar los valores de  $t \in [1, g_{max}]$  al rango de  $q$ . En resumen, las familias de funciones  $\mathcal{F}$  y  $\mathcal{S}$  se construyen de la siguiente manera:

$$\begin{aligned} \forall c_i^1, c_i^2 \in [a_i, b_i], \quad 1 \leq t \leq g_{max}, \\ F^t(c_i^1, c_i^2) = a_i + (b_i - a_i) \cdot T^{\delta_F(t)}\left(\frac{c_i^1 - a_i}{b_i - a_i}, \frac{c_i^2 - a_i}{b_i - a_i}\right) \end{aligned} \quad (6.12)$$

$$\begin{aligned} \forall c_i^1, c_i^2 \in [a_i, b_i], \quad 1 \leq t \leq g_{max}, \\ S^t(c_i^1, c_i^2) = a_i + (b_i - a_i) \cdot G^{\delta_S(t)}\left(\frac{c_i^1 - a_i}{b_i - a_i}, \frac{c_i^2 - a_i}{b_i - a_i}\right) \end{aligned} \quad (6.13)$$

Las funciones de transformación de  $t$  se definen como  $\delta_F(t) = \delta_S(t) = \frac{1}{t}$ .

En cuanto a las familias de funciones  $\mathcal{M}$ , se definen mediante la siguiente función de media:

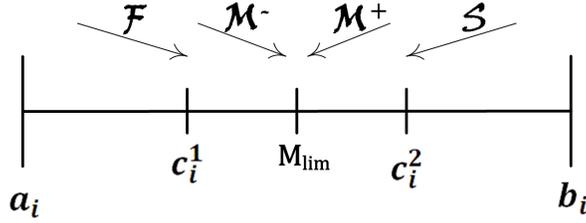
$$P^q(x, y) = \sqrt[q]{\frac{x^q + y^q}{2}}, \text{ siendo } x, y \in [0, 1] \text{ y } -\infty \leq q \leq \infty \quad (6.14)$$

Esta función también está definida en el intervalo  $[0, 1]$ , por lo que hay que transformar los valores de los genes antes de utilizarla. Al final, la función queda de la siguiente manera:

$$\begin{aligned} \forall c_i^1, c_i^2 \in [a_i, b_i], \quad 1 \leq t \leq g_{max}, \\ M^t(c_i^1, c_i^2) = a_i + (b_i - a_i) \cdot P^{\delta_M(t)}\left(\frac{c_i^1 - a_i}{b_i - a_i}, \frac{c_i^2 - a_i}{b_i - a_i}\right) \end{aligned} \quad (6.15)$$

Concretamente, las familias de funciones  $\mathcal{M}^+$  y  $\mathcal{M}^-$  se definen utilizando las funciones de transformación  $\delta_{M^+}(t) = 1 + \ln(g_{max}/t)$  y  $\delta_{M^-}(t) = 1 + \ln(t/g_{max})$  respectivamente.

En la figura 6.3 se muestran gráficamente los intervalos de valores posibles para cada una de las familias de funciones en función de  $t$ .



**Figura 6.3.** Evolución de las posiciones de los cuatro genes creados mediante el cruce dinámico, en función de  $t$ , siendo  $t \in [1, g_{max}]$

#### 6.1.4. Cruce dinámico heurístico (Dynamic heuristic crossover)

Este método es una variante del método anterior. Consiste en utilizar características heurísticas para mejorar la creación de los nuevos cromosomas, centrándose en aquél que sea el mejor de los cromosomas progenitores (Herrera et al., 1996). Así, suponiendo que se tienen que cruzar los cromosomas  $C_1$  y  $C_2$ , donde el primero es más apto que el segundo, se crea un nuevo cromosoma  $H^t = (h_1^t, h_2^t, \dots, h_i^t, \dots, h_n^t)$  en la generación  $t$ . Los genes se combinan siguiendo la siguiente expresión:

$$h_i^t = \begin{cases} F^t(c_i^{1t}, c_i^{2t}) & c_i^{1t} \leq c_i^{2t} \\ S^t(c_i^{1t}, c_i^{2t}) & \text{cualquier otro caso} \end{cases} \quad (6.16)$$

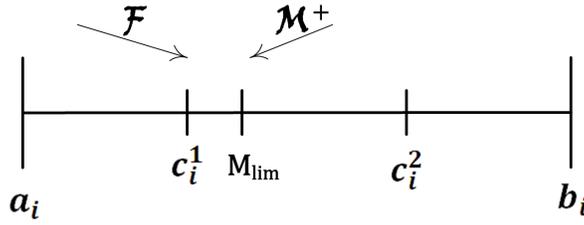
Las funciones  $F^t$  y  $S^t$  pertenecen a las familias de funciones  $\mathcal{F}$  y  $\mathcal{S}$  presentadas en el apartado anterior. Por otro lado, se crea otro cromosoma  $H^t = (h_1^t, h_2^t, \dots, h_i^t, \dots, h_n^t)$ , donde los genes se combinan siguiendo la expresión  $h_i^t = M^t(c_i^{1t}, c_i^{2t})$ . La función  $M^t$  pertenece a la familia de funciones  $\mathcal{M}^+$  si  $c_i^{1t} \leq c_i^{2t}$  y a la familia  $\mathcal{M}^-$  en caso contrario. De todas formas, en ambos casos se utiliza la siguiente función de media:

$$P^q(x, y) = q \cdot x + (1 - q) \cdot y, \text{ siendo } x, y \in [0, 1] \text{ y } 0 \leq q \leq 1 \quad (6.17)$$

El valor de  $q$  se obtiene a partir de las funciones de aptitud de los cromosomas progenitores:

$$q = \frac{f(C_1)}{f(C_1) + f(C_2)}, \text{ donde } f(\cdot) \text{ se refiere a la función de aptitud} \quad (6.18)$$

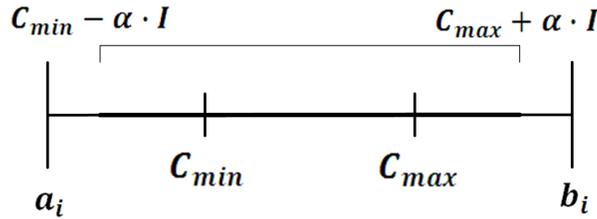
Con este método los nuevos cromosomas creados se acercarán a aquél progenitor con mejor aptitud. En la figura 6.4 se puede ver los intervalos de los genes creados mediante las dos funciones mencionadas en función de  $t$  y suponiendo que la aptitud de  $C_1$  es mayor que  $C_2$ .



**Figura 6.4.** Evolución de las posiciones de los dos genes creados mediante el cruce dinámico heurístico, en función de  $t$ , siendo  $t \in [1, g_{max}]$  y suponiendo que  $C_1$  tiene mejor aptitud que  $C_2$  y siendo  $c_i^1 \leq c_i^2$

**6.1.5. Cruce Blend Alpha (BLX- $\alpha$ )**

Mediante este método se crean dos nuevas cromosomas  $H_k = (h_1^k, h_2^k, \dots, h_i^k, \dots, h_n^k)$ , donde  $k = 1, 2$ . El valor del gen  $h_i^k$  se obtiene de forma aleatoria del intervalo  $[C_{min} - I \cdot \alpha, C_{max} + I \cdot \alpha]$ , donde  $C_{min} = \min(c_i^1, c_i^2)$ ,  $C_{max} = \max(c_i^1, c_i^2)$ ,  $I = C_{max} - C_{min}$  y  $\alpha \in [0, 1]$  (Eshelman y Schaffer, 1993).  $\alpha$  es un valor constante que debe indicar el usuario. En la figura 6.5 se puede ver el intervalo de valores posibles que se obtiene utilizando este método de cruce.

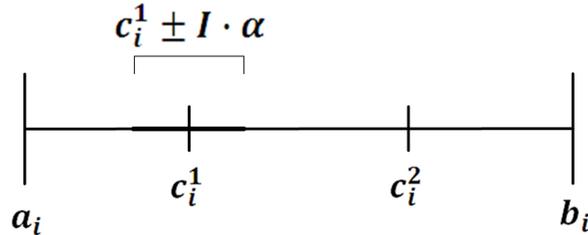


**Figura 6.5.** Intervalo de valores posibles mediante el cruce BLX- $\alpha$  con un valor de  $\alpha = 0,5$

**6.1.6. Cruce Parent-centric BLX- $\alpha$  (PBX- $\alpha$ )**

Este método es una extensión del método de cruce BLX- $\alpha$  anterior (Lozano et al., 2004). El método PBX- $\alpha$  tiene mayor probabilidad que BLX- $\alpha$  de crear cromosomas parecidos a los progenitores. Así, se crean dos nuevas cromosomas  $H_k = (h_1^k, h_2^k, \dots, h_i^k, \dots, h_n^k)$ , donde  $k = 1, 2$ . Los genes del cromosoma  $H_1$  se crean mediante la selección aleatoria en el intervalo  $[l_i^1, u_i^1]$ , siendo  $l_i^1 = \max(a_i, c_i^1 - I \cdot \alpha)$  y  $u_i^1 = \min(b_i, c_i^1 + I \cdot \alpha)$ . En el caso del cromosoma  $H_2$ , sus genes se seleccionan aleatoriamente del intervalo  $[l_i^2, u_i^2]$ , donde  $l_i^2 = \max(a_i, c_i^2 - I \cdot \alpha)$  y  $u_i^2 = \min(b_i, c_i^2 + I \cdot \alpha)$ . En ambos casos, el valor de  $I$

viene dado por la expresión  $I = |c_i^1 - c_i^2|$  y  $\alpha$  es un número constante en el intervalo  $[0, 1]$ . Cuanto mayor sea el valor de  $\alpha$ , mayor será la diversidad de los cromosomas generados. En la figura 6.6 se puede observar gráficamente el intervalo de valores posibles para un cruce.



**Figura 6.6.** Rango de valores posibles para un gen del cromosoma  $H_1$  mediante el cruce PBX- $\alpha$  con un valor de  $\alpha = 0,25$

## 6.2. Operación de mutación

En la operación de mutación, dependiendo de si la tasa o el efecto de la mutación disminuye con el paso de las generaciones o no, se pueden diferenciar las mutaciones *uniformes* y las *no-uniformes*. Mientras que las mutaciones **uniformes** tienen el mismo efecto durante todo el proceso de entrenamiento, las mutaciones **no-uniformes** atenúan su efecto según el entrenamiento va fluyendo hacia la solución óptima. De esta forma, al comienzo del entrenamiento abarcan un espacio de búsqueda mayor y hacia el final las variaciones de la mutación son muy pequeñas.

En adelante, suponiendo que el cromosoma a mutar es  $C = \{c_1, \dots, c_i, \dots, c_n\}$ , donde  $c_i \in [a_i, b_i]$ , siendo  $n$  el número de genes del cromosoma y  $a_i$  y  $b_i$  los valores mínimos y máximos posibles para el gen  $c_i$ . A continuación se explican los distintos procesos de mutación que se han implementado en la presente tesis:

### 6.2.1. Mutación aleatoria

Consiste en cambiar un gen  $c_i$  por un valor aleatorio en el rango  $[a_i, b_i]$ .

### 6.2.2. Mutación gaussiana

En lugar de utilizar todo el intervalo  $[a_i, b_i]$ , se utiliza una distribución gaussiana con media en el gen a mutar ( $\mu = c_i$ ) y una desviación estándar que puede ser una décima parte del valor máximo de ese gen ( $\sigma = 0,1 \cdot b_i$ ) (Hinterding, 1995). En la figura 6.7 se muestra visualmente el intervalo de valores posibles utilizando un mutación gaussiana.

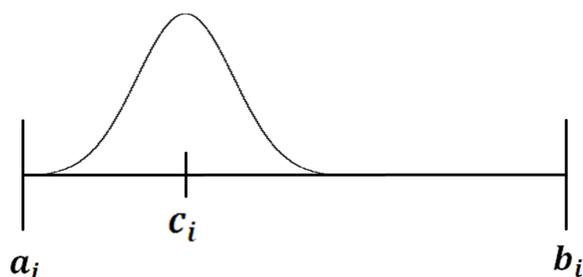


Figura 6.7. Intervalo de valores de una mutación gaussiana

### 6.2.3. Mutación de Michalewicz

Se trata de una mutación no-uniforme que consiste en modificar el gen a mutar siguiendo la siguiente expresión (Michalewicz, 1998):

$$c'_i = \begin{cases} c_i + \Delta(t, b_i - c_i) & u = 0 \\ c_i - \Delta(t, c_i - a_i) & u = 1 \end{cases} \quad (6.19)$$

En este caso,  $u$  es un número aleatorio que coge un valor de 0 ó de 1 y  $t$  es la generación actual. La función  $\Delta(t, x)$ , se define como:

$$\Delta(t, x) = x \cdot (1 - v^{(1 - \frac{t}{g_{max}})^w}) \quad (6.20)$$

En esta función,  $v$  es un valor aleatorio en el intervalo  $[0, 1]$  y  $w$  es un número constante que determina el grado de dependencia de la función en el número de iteraciones. Esta función devuelve un valor en el intervalo  $[0, x]$ , donde la probabilidad de devolver un valor cercano a 0 aumenta con el número de iteraciones. En la figura 6.8 se puede ver visualmente el intervalo de valores barajado por esta operación de mutación.

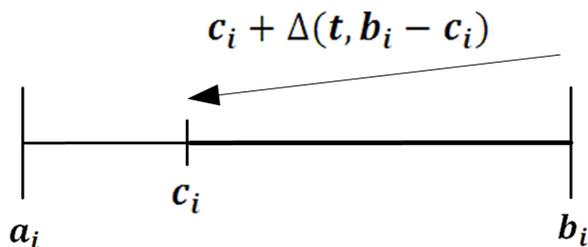


Figura 6.8. Evolución del intervalo de valores de la mutación de Michalewicz siendo  $v = 0,5$ ,  $u = 0$  y  $w = 1$



## Parte III

---

### Estado del arte



La Parte III se compone de tres capítulos, en los cuales se analiza el **estado del arte** en relación a la solución presentada en la tesis.

En el capítulo 7 se comparan distintos formalismos de representación de Guías Clínicas Informatizadas en base a los sistemas de clasificación de los que hacen uso.

En el capítulo 8, por su lado, se estudian distintas soluciones informáticas utilizadas en medicina que utilizan Sistemas de Inferencia Difusa como sistema de razonamiento.

Por último, en el capítulo 9 se presentan las alternativas existentes en relación a *frameworks* para el desarrollo de Sistemas de Inferencia Difusa.



## Sistemas de clasificación integrados en formalismos de GCI

En este capítulo se realiza una revisión de los sistemas de clasificación incluidos en formalismos de representación de GCI que siguen el paradigma MRT y que se han tenido en cuenta en las comparativas de Peleg (Peleg et al., 2003) y De Clercq (Clercq et al., 2008): *PROforma*, *GLIF3*, *GLARE*, *New-Guide*, *SAGE* y *Asbru*. Como siguen el paradigma MRT todos los formalismos estudiados representan las GCI como un grafo multinivel. De la misma forma, todos los formalismos utilizan nodos de decisión, consulta o pregunta y acción.

Los sistemas de clasificación son un elemento muy importante en los formalismos de representación de GCI ya que se encargan de procesar los cálculos complejos contenidos en las guías. En esta revisión se ha tenido en cuenta tanto el *tipo de sistema de clasificación* utilizado en el formalismo como la *relación entre el formalismo y el sistema de clasificación*.

El **tipo de sistema de clasificación** establece tanto la efectividad del sistema como la información que ofrece al usuario tras realizar la clasificación. La estructura de los sistemas de clasificación es común: todos ellos tienen unas *variables de entrada* y unas *variables de salida* y siguen un *proceso de razonamiento*.

La **relación entre el formalismo y el sistema de clasificación** establece cómo es la interacción entre el formalismo y el sistema de clasificación, esto es, *cuál es la estructura del formalismo en que se integra el sistema de clasificación, cómo se realiza la integración y si el formalismo dispone de un sistema para facilitar la definición de sistemas de clasificación*. Todas estas características son importantes ya que afectan a la efectividad de la GCI tanto en tiempo de desarrollo como de ejecución.

A continuación se explican más extensamente las características de comparación. Posteriormente se realiza la revisión propiamente dicha. Finalmente, se realiza una breve síntesis de la revisión y una comparativa de los formalismos revisados.

### 7.1. Tipo de sistema de clasificación

Esta característica especifica el sistema de clasificación que permite implementar el formalismo de representación de GCI. Dado que el número de tipos de sistemas de clasificación es muy elevado, se ha optado por realizar una breve taxonomía de los mismos en base a ciertos criterios, tales como la *comprensibilidad* por parte de los no expertos, la *metodología de construcción* seguida para crearlos y el uso de la *lógica difusa*.

La **comprensibilidad** mide la facilidad con la que personas no expertas pueden entender la estructura del sistema de clasificación y su proceso de razonamiento. Existen dos tipos de sistemas basados en este criterio: los de *caja blanca* y los de *caja negra*. Los sistemas de **caja blanca** son sistemas transparentes donde la estructura y el razonamiento están bien explicados, mientras que los sistemas de **caja negra** resultan opacos debido a que bien su estructura bien su razonamiento no está claramente explicado.

La **metodología de construcción** indica cómo se ha creado el sistema de clasificación en sí, siendo la *construcción manual* a partir de información de expertos y bibliografía una de las opciones, y la *construcción automática* a partir de datos la otra opción. En aquéllos sistemas que se **construyen manualmente** el usuario (ingeniero, diseñador...) debe establecer la estructura del sistema de clasificación, así como ciertos elementos clave del proceso de razonamiento. En los sistemas de clasificación **construidos automáticamente** se utilizan algoritmos de aprendizaje automático y los usuarios sólo deben indicar ciertas características de los algoritmos. Los sistemas de clasificación construidos automáticamente a partir de datos siguen distintos algoritmos de aprendizaje, los cuales se dividen en dos corrientes, los de *aprendizaje supervisado* y los de *aprendizaje no supervisado*.

En los algoritmos de **aprendizaje supervisado** los datos que se utilizan para su construcción tienen los valores de salida correspondientes. En el caso de los algoritmos de **aprendizaje no supervisado**, los datos utilizados no tienen la información de las variables de salida. Por lo tanto, estos sistemas requieren una intervención humana para indicar posteriormente el número y el tipo de las variables de salida.

Por último, la **lógica difusa** es una característica interesante desde el punto de vista del modelado de la incertidumbre y algunos sistemas de clasificación se basan en la misma.

A continuación se analizan en base a los criterios anteriores los tipos de sistemas de clasificación más conocidos considerados de *soft computing* (Karlík, 2013) (Bhatia et al., 2014) y que se pueden encontrar en los formalismos de representación de GCI considerados en las comparativas de Peleg (Peleg et al., 2003) y De Clercq (Clercq et al., 2008). El *soft computing* es un término utilizado para denominar todos los sistemas de clasificación con capacidades de procesamiento de información flexible o incierta en entornos ambiguos (Yardimci, 2009). Los sistemas analizados son los *sistemas basados en reglas*, *árboles de decisión*, *redes neuronales artificiales*, *sistemas de inferencia difu-*

*sa basados en redes adaptativas, máquinas de soporte vectorial, sistemas de razonamiento probabilístico, sistemas basados en votación, mapas cognitivos difusos y sistemas basados en argumentación.*

### 7.1.1. Sistemas basados en reglas

Los sistemas basados en reglas suelen ser de caja blanca y entre las diversas variantes existentes, aparte de los **SID** destacan las *reglas de asociación*.

Las **reglas de asociación** son sistemas basados en reglas en el que el cumplimiento del antecedente implica que el consecuente también se cumple (Agrawal et al., 1993). Se representan mediante reglas de tipo  $A \Rightarrow B$ , donde  $A$  y  $B$  son dos conjuntos de hechos. Esta regla indica que cuando se cumplen todos los hechos del conjunto  $A$ , también se cumplen todos los hechos del conjunto  $B$ . Son sistemas de clasificación que se crean automáticamente a partir de un conjunto de casos utilizando algoritmos de aprendizaje no supervisado.

### 7.1.2. Árboles de decisión

Los árboles de decisión son sistemas de clasificación de caja blanca que sirven para representar y categorizar condiciones que ocurren de forma sucesiva (Rokach y Maimon, 2007). En el ámbito de la medicina son uno de los sistemas de clasificación más utilizados (Esfandiari et al., 2014). Se pueden crear tanto manualmente a partir de información de expertos y de bibliografía como de forma automática a partir de datos utilizando algoritmos de aprendizaje supervisado.

### 7.1.3. Redes neuronales artificiales

Las redes neuronales artificiales (RNA) son un tipo de sistema de clasificación inspirado en la forma en que funciona el sistema nervioso de los animales (Rosenblatt, 1958). Son sistemas de clasificación de caja negra que se crean o entrenan a partir de los datos. Se pueden encontrar modalidades que siguen algoritmos supervisados y algoritmos no supervisados.

### 7.1.4. Sistemas de inferencia difusa basados en redes adaptativas

Los sistemas de inferencia difusa basados en redes adaptativas (SIDRA) combinan las RNA con los SID con la intención de conseguir sistemas de clasificación que aúnen interpretabilidad con efectividad. El primero de estos sistemas lo presentó Jang a comienzos de los 90 como un método para mejorar las prestaciones de los SID (Jang, 1993). En su momento, ya detectó que los SID a pesar de todas las aplicaciones en que se habían utilizado, tenían ciertas limitaciones en cuanto a efectividad y que era necesario incluir un sistema que adaptase algunas de sus características. Para ello, propuso un sistema basado

en una RNA que emulaba el funcionamiento de la variante Takagi-Sugeno de SID y adaptó los algoritmos de entrenamiento de las RNA a ese nuevo formalismo. Los SIDRA son sistemas de caja negra, debido a que su estructura se asemeja a la de las RNA. También se crean automáticamente a partir de casos y para ello hacen uso de algoritmos de aprendizaje supervisado. Los SIDRA normalmente obtienen unos resultados en términos de eficiencia muy altos. Por ello, se han convertido en uno de los sistemas de clasificación más utilizados (Ankishan y Yilmaz, 2013) (Efe, 2009) (Gharaviri et al., 2008).

### 7.1.5. Máquinas de soporte vectorial

Las máquinas de soporte vectorial son sistemas de clasificación que tratan de dividir en dos el conjunto de casos a clasificar mediante un hiperplano óptimo (Vapnik, 1995). Son sistemas de caja negra que se crean automáticamente a partir de los datos, para lo cual utilizan algoritmos de aprendizaje supervisado.

### 7.1.6. Sistemas basados en razonamiento probabilístico

Los sistemas basados en razonamiento probabilístico combinan la capacidad de la teoría de la probabilidad para manejar la incertidumbre con la capacidad de la lógica deductiva para explotar la estructura. Normalmente son sistemas que se describen mediante un grafo y probabilidades, y se pueden considerar de caja blanca. Existen múltiples alternativas diferentes, cada una con sus características específicas, tales como los *clasificadores bayesianos ingenuos*, las *redes bayesianas*, los *diagramas de influencia* y los *modelos ocultos de Markov* que se comentan a continuación.

Los **clasificadores bayesianos ingenuos** se fundamentan en el teorema de Bayes y en la creencia de la independencia de las variables de entrada (Maron y Kuhns, 1960). Asumen que la presencia o ausencia de una característica de una variable de entrada es totalmente independiente del resto de sus características. Es posible definir clasificadores bayesianos ingenuos a partir de información de expertos y de bibliografía, aunque normalmente se crean automáticamente a partir de datos utilizando algoritmos de aprendizaje supervisado (concretamente estimadores de máxima verosimilitud). La creación automática necesita muy pocos datos para poder estimar los parámetros del clasificador.

Las **redes bayesianas** son grafos acíclicos dirigidos cuyos nodos representan las variables de entrada y de salida y los arcos establecen una relación de dependencia entre esas variables (Pearl, 1988). Estas relaciones de dependencia de expresan en términos de probabilidad. Las redes bayesianas permiten modelar de forma intuitiva modelos inciertos y son muy utilizadas en sistemas de ayuda a la toma de decisiones en medicina (Lucas et al., 2004). Normalmente se suelen crear de forma manual a partir de información bibliográfica y de

expertos. Requieren alto conocimiento del ámbito, ya que hay que especificar explícitamente las probabilidades (Orphanou et al., 2014).

Los **diagramas de influencia** permiten describir las dependencias entre un conjunto de variables de entrada y salida (Howard y Matheson, 2005). Son una generalización de las redes bayesianas y también se representan mediante grafos acíclicos, aunque utilizan distintos tipos de nodos y arcos. Se suelen crear manualmente a partir de bibliografía y de información de expertos. Requieren de un extenso conocimiento del ámbito concreto del diagrama por parte del diseñador del sistema para poder reflejar los distintos elementos con exactitud.

Los **modelos ocultos de Markov** (MOM) son sistemas probabilísticos en los que se asume que el proceso a modelar es un proceso de Markov (esto es, un proceso donde los estados sólo dependen del valor actual y no de la historia de los valores) (Baum y Petrie, 1966). Los MOM son grafos donde los nodos representan las variables de entrada y salida y los arcos entre los nodos las probabilidades de desplazarse de un nodo a otro. Se crean en parte manualmente a partir de información de expertos o bibliografía y en parte automáticamente a partir de casos. Así, el primer modelo del MOM necesita ser definido por un usuario. Posteriormente, los valores de las probabilidades del modelo se establecen automáticamente.

#### 7.1.7. Sistemas basados en votación

Los sistemas basados en votación toman sus decisiones en base a un conjunto de casos conocidos siguiendo un proceso de votación, es decir, agregando preferencias individuales con el objetivo de llegar a una decisión colectiva (Pacuit, 2012). Históricamente se ha trabajado mucho sobre ellos en el campo de las ciencias sociales, pero también se han extendido al campo de la toma de decisiones utilizando herramientas informáticas (Saeed et al., 2012). Se consideran sistemas de caja blanca. Los más representativos son los *k vecinos más cercanos*, y su variante con lógica difusa, los *k vecinos más cercanos difusos*.

Los **k vecinos más cercanos** trata de identificar el valor de la variable de salida de un caso de entrada en base a cierto número ( $k$ ) de casos con salidas conocidas y que tienen características similares (Silverman y Jones, 1989). El criterio de decisión utilizado es la distancia que hay entre las características de los casos conocidos y el caso de entrada. Se crean de forma automática a partir de los datos mediante algoritmos de aprendizaje supervisado.

Los **k vecinos más cercanos difusos** utilizan las características de la lógica difusa para dotar de mayor flexibilidad a los  $k$  vecinos más cercanos (Derrac et al., 2014). Así, permiten tanto que las distancias calculadas entre los casos conocidos y el caso de entrada como las decisiones de cada uno de los casos conocidos tengan asociados grados de certeza.

### 7.1.8. Mapas cognitivos difusos

Los mapas cognitivos difusos (MCD) son sistemas de caja blanca que se representan mediante grafos dirigidos. Los nodos del grafo representan cualquier concepto del sistema a modelar (atributos, características, cualidades, variables de entrada y salida y estados) y los arcos las relaciones de causalidad entre los mismos (Kosko, 1986). Aunque hay algoritmos para crear y afinar los MCD de forma automática a partir de datos (Song et al., 2010), lo más común es construirlos manualmente a partir de información de expertos y de bibliografía (Papageorgiou y Salmerón, 2013).

### 7.1.9. Sistemas basados en argumentación

Los sistemas basados en argumentación se basan en la construcción y evaluación de distintos argumentos (Amgoud y Prade, 2009). Estos argumentos se utilizan para apoyar, explicar o desestimar decisiones y opiniones. No pretenden elegir una única respuesta, sino razonar sobre todas las respuestas posibles para que el usuario pueda elegir, ya que muestran explícitamente la información de su razonamiento. Por lo tanto, se consideran de caja blanca. Normalmente, estos sistemas se crean automáticamente a partir de una base de conocimiento haciendo uso de árboles como estructuras para guardar los argumentos y utilizando algoritmos que se incluyen dentro de los de aprendizaje supervisado (Prakken, 2009).

## 7.2. Relación entre el formalismo y el sistema de clasificación

Las características que tienen en cuenta la relación entre el formalismo de representación y el sistema de clasificación son la *estructura de definición* de la GCI en que se integra el sistema de clasificación, el *método de integración* del mismo y el *método de especificación* que ofrece el formalismo.

La **estructura de definición** indica en qué elemento específico de la GCI se integra el sistema de clasificación. La estructura en que se define el sistema de clasificación es importante ya que cada estructura tiene un funcionamiento diferente dentro de la GCI. Las diferentes alternativas de esta característica abarcan desde los distintos tipos de nodos de la GCI (*acción, decisión o consulta*) hasta las condiciones que definen el flujo de ejecución de la misma.

El **método de integración** muestra cómo se integra el sistema de clasificación desarrollado en la estructura indicada por la característica anterior y afecta directamente al rendimiento general de las GCI. Las alternativas que se valoran son dos: Los sistemas pueden estar *embebidos* en el propio código de la GCI o pueden ser sistemas *externos* a los que se llama durante la ejecución de la GCI. Los sistemas **embebidos** se especifican bien en el mismo lenguaje que la GCI, bien en un lenguaje de expresiones incluido en el formalismo de

representación. Los sistemas **externos**, por su lado, pueden estar descritos en cualquier lenguaje de programación para el que se disponga de soporte en tiempo de ejecución de la GCI. El método de integración también determina el *sistema de ejecución*. Así, los sistemas embebidos se ejecutan utilizando el mismo motor utilizado para ejecutar las GCI. Sin embargo, los sistemas externos pueden utilizar cualquier motor de ejecución que soporte el lenguaje en que están descritos los sistemas de clasificación y que tenga comunicación con el motor de ejecución de las GCI. El hecho de utilizar un único motor de ejecución para la GCI y el sistema de clasificación hace que los sistemas embebidos sean más rápidos. Sin embargo, son menos flexibles ya que están limitados por la expresividad del formalismo de representación de GCI.

Finalmente, el **método de especificación** indica si el formalismo de representación de GCI incluye una herramienta para evitar la programación manual del sistema de clasificación. Esta característica afecta a la etapa de desarrollo del sistema de clasificación, y por lo tanto, de la GCI. Disponer de un método para evitar la programación manual del sistema de clasificación evita o minimiza el tener que conocer todos los detalles técnicos de los sistemas de clasificación y reduce el número de los errores propios de la programación manual, de forma que la definición del sistema es más intuitiva y la depuración de errores resulta más sencilla. Entre las posibilidades valoradas en esta característica están las *tablas*, los *formularios* y los *editores específicos*.

### 7.3. Revisión de los formalismos de representación de GCI

A continuación se explica cada uno de los formalismos estudiados. Para cada formalismo primero se realiza una breve descripción y seguidamente se detallan sus especificaciones respecto a las características de tipo de sistema de clasificación, estructura de definición, método de integración y método de especificación.

#### 7.3.1. PROforma

PROforma combina la programación lógica y el modelado orientado a objetos (Sutton y Fox, 2003). Dispone de cuatro tipos de nodos: planes, decisiones, acciones y consultas. Entre otros resultados, PROforma ha dado lugar a la solución libre de soporte a la decisión Tallis (COSSAC, 2017) y a la solución comercial Arezzo (InferMed, 2017).

PROforma utiliza un **sistema de razonamiento basado en argumentación**. Cada decisión tiene una serie de candidatos y cada uno de éstos dispone de una serie de argumentos a favor y en contra. Algunos de estos argumentos son definitivos, esto es, si se cumple el argumento el candidato será elegido o excluido. El resto de argumentos aportarán un peso que decantará la decisión hacia uno u otro candidato.

En PROforma el sistema de clasificación se especifica en los **nodos de decisión**.

Los sistemas de clasificación en PROforma van **embebidos** en el propio código de la GCI, por lo que se ejecutan por el mismo motor de ejecución de la GCI.

Por último, tanto Arezzo como Tallis ofrecen herramientas basadas en el uso de **formularios** para poder especificar los sistemas de clasificación. Estas herramientas permiten definir los candidatos y las variables a tener en cuenta en la decisión y ofrecen un sistema de calculadora formal o editor de expresiones para indicar los argumentos y los pesos de los mismos.

### 7.3.2. GLIF3

GLIF3 es el acrónimo de *Guideline Interchange Format versión 3* (Boxwala et al., 2004). Se trata de un modelo para la representación de GCI compartibles. En este formalismo se da especial importancia a la posibilidad de compartir guías entre distintas instituciones y programas informáticos. Se ha especificado basándose en características de otros formalismos de representación e incorporando estándares usados en medicina como Health Level 7 (HL7, 2017).

En GLIF3 se permite el uso tanto de **árboles de decisión** como de **sistemas basados en reglas**.

Estos sistemas de clasificación se utilizan en un tipo de **nodo de decisión** llamado *data mining decision step* (Kazemzadeh y Sartipi, 2006).

Los sistemas de clasificación utilizados en GLIF3 son **externos** y se especifican en lenguaje *Predictive Model Markup Language* (DMG, 2017). Estos ficheros se integran en los nodos de decisión de la GCI y en tiempo de ejecución, cuando se llega a uno de estos nodos, se ejecuta un módulo encargado de llevar a cabo el cálculo del sistema de clasificación. En este caso, los módulos son programas en lenguaje Java, los cuales se encargan de leer el fichero del sistema de clasificación y de acceder a la información del paciente. Una vez realizado el cálculo, se devuelve el resultado para que la ejecución de la guía continúe.

En lo que al método de especificación se refiere, GLIF3 dispone de una serie de **formularios** basados en la herramienta para la definición de ontologías Protégé (Protégé, 2017).

### 7.3.3. GLARE

GLARE es un sistema independiente del dominio para la adquisición, representación y ejecución de GCI (Terenziani et al., 2001) que permite la inclusión de restricciones temporales en las acciones de las guías (Terenziani et al., 2002).

El sistema de clasificación utilizado en GLARE es un **sistema basado en votación**. Mediante este sistema primero se definen unos valores/rangos

en base a los cuales se debe tomar la decisión. Posteriormente, se relacionan esos valores/rangos y las decisiones mediante pesos. También se establece un umbral por encima del cual el sistema realizará su recomendación. El funcionamiento consiste en sumar los pesos de aquellos rangos/valores que coinciden con los datos de entrada. El sistema recomendará aquella decisión que supere el umbral.

Los sistemas de clasificación de GLARE se introducen en las **acciones de decisión** (*decision action*) de la GCI.

Son sistemas que van **embebidos** en la guía. Por lo tanto, el motor de ejecución de la GCI se encarga de ejecutar también los sistemas de clasificación.

Por último, los sistemas de clasificación se especifican utilizando **tablas**, de forma que en las columnas se establecen los valores/rangos, en las filas las decisiones y en las celdas los pesos.

#### 7.3.4. NewGuide

NewGuide es un *framework* de modelado y ejecución de GCI que se caracteriza principalmente por su énfasis en la integración de la ejecución de GCI con los flujos de trabajo organizacionales de los centros sanitarios (Cicarese et al., 2005).

Incluye la posibilidad de definir **sistemas basados en razonamiento probabilístico** (concretamente *diagramas de influencia*) y **árboles de decisión** (Quaglioni et al., 1998).

Estos sistemas de clasificación se definen en los **nodos de decisión** de tipo *non-deterministic one of*. De esta forma, se pretende suplir la posible falta de evidencia a la hora de tomar algunas decisiones.

Los sistemas de clasificación son sistemas **externos**, ya que se ejecutan mediante una serie de applets de Java. Durante la ejecución de la guía se llama a estas aplicaciones para que realicen el cálculo y devuelvan el resultado al motor de ejecución de la GCI.

Para la especificación de estos sistemas de clasificación se ofrecen **editores específicos**.

#### 7.3.5. SAGE

Este formalismo permite definir GCI que pueden interactuar con los HCE de los centros sanitarios (Tu et al., 2007). Las principales características de SAGE son: la organización de las guías en conjuntos de recomendaciones, un Registro Médico Virtual basado en el Modelo de Referencia de Información de Health Level 7 que facilita la conexión de las guías con los HCE y el uso de terminologías estándar para la representación de conceptos.

El sistema de clasificación utilizado es un **sistema basado en argumentación**.

En el formalismo SAGE los sistemas de clasificación se integran en los **nodos de decisión** (Tu y Glasgow, 2006).

En lo que a la integración con la GCI se refiere, no se ha hallado información sobre si los sistemas de clasificación van embebidos en las GCI o son sistemas externos.

Para especificar los sistemas de clasificación SAGE hace uso de **formularios** basados en Protégé. Mediante estos formularios, el usuario puede indicar las distintas alternativas y las condiciones a favor y en contra de las mismas.

### 7.3.6. Asbru

Asbru es un formalismo de representación de GCI que hace especial hincapié tanto en las recomendaciones de la guía como en las intenciones de la misma (Shahar et al., 1998). De esta forma, se captura la esencia del procedimiento y a la vez se deja suficiente flexibilidad en tiempo de ejecución para alcanzar intenciones específicas en base al estado del paciente.

Este formalismo no ofrece la posibilidad de utilizar sistemas de clasificación. Asbru utiliza **sentencias de lógica proposicional** para establecer precondiciones en los nodos de la GCI. Estas precondiciones sirven para definir patrones temporales que dirijan el flujo de ejecución de la GCI.

Como es evidente, las sentencias proposicionales se definen en los propios nodos que condicionan, por lo que **cualquier nodo** puede incluir estas sentencias.

Como se definen utilizando el mismo lenguaje utilizado para definir la propia GCI, se considera que están completamente **embebidas en la guía**.

No se ha hallado evidencia sobre si Asbru ofrece algún método de especificación que no requiera de la programación manual de las sentencias proposicionales.

## 7.4. Síntesis y comparativa

En la tabla 7.1 se muestra un resumen de las características analizadas para todos los formalismos de representación.

Todos los formalismos de representación de GCI analizados (excepto Asbru) incluyen los sistemas de clasificación en los nodos de decisión. En Asbru se incluyen en los propios nodos que condicionan, sin importar el tipo.

Entre los formalismos estudiados se identifican dos corrientes diferentes a la hora de integrar un sistema de clasificación la GCI. Así, mientras que en los casos de PROforma, GLARE y Asbru se utilizan los propios atributos del nodo para definir los sistemas de clasificación, en los casos de GLIF3 y NewGuide los sistemas de clasificación se definen como recursos de la guía que son accedidos desde los nodos de decisión. Por lo tanto, los primeros se encuentran directamente embebidos en la guía, mientras que los segundos mantienen cierta independencia respecto a la misma. En el caso de SAGE no se ha encontrado información sobre esta característica.

En cuanto a los sistemas de clasificación utilizados, el punto anterior tiene un efecto directo en las alternativas ofrecidas por los formalismos analizados. De esta forma, aquellos formalismos que llevan el sistema de clasificación directamente embebido en la guía (PROforma, GLARE y Asbru) ofrecen un único sistema de clasificación, mientras que los otros formalismos (GLIF3 y NewGuide) ofrecen más de una alternativa. Aunque la variedad de sistemas de clasificación es amplia, todos ellos tienen un funcionamiento en el que dan mayor importancia a la capacidad de justificar los resultados que a la efectividad (sistemas de caja blanca).

Es interesante señalar que todos los formalismos analizados ofrecen herramientas específicas para la definición de los sistemas de clasificación, no siendo necesaria de esta forma la programación manual de los mismos.

Formalismo	Sistema de clasificación	Estructura de definición	Método de integración	Método de especificación
<b>PROForma</b>	Sistema basado en argumentación	Nodos de decisión	Embebido	Formularios
<b>GLIF3</b>	Árboles de decisión, sistemas basados en reglas	Nodos de decisión	Externo	Formularios
<b>GLARE</b>	Sistema basado en votación	Acciones de decisión	Embebido	Tablas
<b>NewGuide</b>	Diagramas de influencia y árboles de decisión	Nodos de decisión	Externo	Editores específicos
<b>SAGE</b>	Sistema basado en argumentación	Nodos de decisión		Formularios
<b>Asbru</b>	Lógica proposicional temporal	Cualquier nodo	Embebido	

Tabla 7.1. Comparativa sobre los sistemas de clasificación integrados en formalismos de representación de GCI

## Sistemas de Inferencia Difusa utilizados en medicina

En este capítulo se revisan 37 soluciones en el ámbito médico que utilizan como sistema de clasificación un SID. Todas estas soluciones se han extraído de los repositorios ScienceDirect (Elsevier, 2017) e IEEEExplore (IEEE, 2017) entre los años 2010 y 2017.

Las distintas soluciones se comparan de acuerdo a los *tipos de SID* y *FP* que utilizan y a la *metodología de creación* que se ha seguido para construir el SID. En el caso de aquellas soluciones que sigan el MDL, también se ha tenido en cuenta si hacen uso de alguna clase de sistema de aprendizaje automático para mejorar sus prestaciones. También se mencionan otras características o detalles que hagan destacar la solución analizada frente al resto, como puede ser, por ejemplo, el uso de la LD-2.

En el apartado siguiente se describe brevemente cada una de las soluciones analizadas para seguidamente especificar las características de comparación que cumplen. Al final del capítulo se realiza una síntesis de la revisión.

### 8.1. Análisis de las soluciones

En las tablas 8.1, 8.2 y 8.3 se muestran de forma concisa las características de cada una de las soluciones revisadas. La columna *Tipo de SID* muestra el tipo de SID que se utiliza en la solución, la columna *FP* hace referencia a las FP que se utilizan en el SID de la solución y en la columna *Metod. creación (Entr.)* se indica el método que se ha seguido para crear la solución, MDL, MDP o mixto y en el caso de las que se han creado siguiendo el MDL, se indica si dispone de entrenamiento o no para mejorar las características del SID utilizado.

(Khatibi y Montazer, 2010) presentan un sistema de evaluación del riesgo de sufrir una enfermedad coronaria. Esta solución es un **sistema de inferencia híbrida difuso-evidente**. Los conjuntos difusos utilizados siguen una **distribución propia** acorde a sus necesidades, se ha construido siguiendo

Solución	Tipo de SID	FP	Metod. creación (Entr.)
(Khatibi y Montazer, 2010)	SIHDE	PR	MDL (No)
(Mahmoodabadi et al., 2010)	CDR	PR	MDL (No)
(Rabbi et al., 2010)	M	TA y TP	MDL (No)
(Esposito et al., 2011)	M	TP	MDL (Sí)
(Cosenza, 2012)	2 M, 1 TS, y 1 SIDRA	TA y TP	MDL (No)
(Muthukaruppan y Er, 2012)	M	TA	Mixto
(Mago et al., 2012)	M	TP, G, S, PL y PR	MDL (No)
(Czabanski et al., 2012, 2013)	TS	TP	MDL (No)
(Garibaldi et al., 2012)	M	TA y TP	MDL (No)
(Pal et al., 2012)	M	TP	MDL (No)
(Seki y Mizumoto, 2012, 2013)	MREU	TA y G	MDL (Sí)

**Tabla 8.1.** Comparativa sobre las soluciones en medicina que utilizan un SID. Leyenda: *Tipo de SID*: M - Mamdani, TS - Takagi-Sugeno, SIDRA - Sistema de inferencia difusa basado en redes adaptativas, CDR - Clasificador difuso relacional, SIHDE - Sistema de inferencia híbrida difuso-evidente, MREU - SID con módulos de entrada única, SDP - Sistema difuso probabilístico. *FP*: TA - Triangular, TP - Trapezoidal, G - Gaussiana, S - Sigmoidea, CP - De campana, PL - polinómica, PR - Propia, CT - Constante, R - Rectangular.

<b>Solución</b>	<b>Tipo de SID</b>	<b>FP</b>	<b>Metod. creación (Entr.)</b>
(Samuel et al., 2013)	M	TA	MDL (No)
(Ghasemi et al., 2013)	SIHDE	TA	MDP
(Beloufa y Chikh, 2013)	M	TA	MDP
(Castanho et al., 2013)	TS	TP	MDP
(de Brito et al., 2013)	M	TA y TP	MDL (No)
(Pawade et al., 2013)	M	G	MDL (No)
(Lai et al., 2013, 2014)	M	CP	MDL (Sí)
(Marateb et al., 2014)	M	G y S	Mixto
(Bárdossy et al., 2014)	TS	S	MDP
(Chourasia et al., 2014)	M	G	MDL (No)
(Sanz et al., 2014)	M	TA	MDP
(Dennis y Muthukrishnan, 2014)	M	TA	MDP
(Miranda y Felipe, 2015)	M	PL	Mixto

**Tabla 8.2.** Comparativa sobre las soluciones en medicina que utilizan un SID (continuación). Leyenda: *Tipo de SID*: M - Mamdani, TS - Takagi-Sugeno, SIDRA - Sistema de inferencia difusa basado en redes adaptativas, CDR - Clasificador difuso relacional, SIHDE - Sistema de inferencia híbrida difuso-evidente, MREU - SID con módulos de entrada única, SDP - Sistema difuso probabilístico. *FP*: TA - Triangular, TP - Trapezoidal, G - Gaussiana, S - Sigmoidea, CP - De campana, PL - polinómica, PR - Propia, CT - Constante, R - Rectangular.

<b>Solución</b>	<b>Tipo de SID</b>	<b>FP</b>	<b>Metod. creación (Entr.)</b>
(Nguyen et al., 2015)	TS	G	MDP
(Andrei et al., 2015)	TS	TP	MDL (No)
(Silveira y de Barros, 2015)	TS	TP	MDP
(Nuryani et al., 2015)	M	G	MDL (Sí)
(Doctor et al., 2016)	M	TP	MDL (No)
(Al-Ebbini et al., 2016)	M y TS	CP y CT	MDP
(de Bruin et al., 2016)	TS	TP	MDL (No)
(Salgado et al., 2016)	TS	G	MDP
(Büyükkavcu et al., 2016)	M	TA	MDL (No)
(Oluwagbemi et al., 2016)	M	TA	MDL (No)
(Fialho et al., 2016)	SDP	G	MDP
(Meza-Palacios et al., 2017)	M	TP	MDL (No)
(Gorzalczany y Rudzinski, 2017)	M	TP	MDP

**Tabla 8.3.** Comparativa sobre las soluciones en medicina que utilizan un SID (continuación). Leyenda: *Tipo de SID*: M - Mamdani, TS - Takagi-Sugeno, SIDRA - Sistema de inferencia difusa basado en redes adaptativas, CDR - Clasificador difuso relacional, SIHDE - Sistema de inferencia híbrida difuso-evidente, MREU - SID con módulos de entrada única, SDP - Sistema difuso probabilístico. *FP*: TA - Triangular, TP - Trapezoidal, G - Gaussiana, S - Sigmoidea, CP - De campana, PL - polinómica, PR - Propia, CT - Constante, R - Rectangular.

la metodología **MDL** y **no utiliza** ningún sistema de entrenamiento para mejorar su efectividad.

(Mahmoodabadi et al., 2010) presentan un sistema de diagnóstico de enfermedades metabólicas cerebrales en niños. Para ello utilizan un **clasificador difuso relacional**. Los conjuntos difusos de este clasificador siguen una **distribución propia**. Han definido el clasificador siguiendo el **MDL**. En cuanto al afinamiento del comportamiento de la solución, **no utiliza** ninguna clase de sistema de entrenamiento.

(Rabbi et al., 2010) ofrecen un sistema para detectar ataques epilépticos mediante el análisis de registros de encefalogramas. El núcleo de esta solución es un SID de **tipo Mamdani**. Sus variables clínicas están formadas por conjuntos difusos definidos mediante **funciones triangulares y trapezoidales**. El sistema analizado se ha creado siguiendo el **MDL** y **no dispone** de un sistema de aprendizaje automático para mejorar el rendimiento.

(Esposito et al., 2011) presentan un sistema para la evaluación del estado de salud en pacientes con esclerosis múltiple. El SID utilizado es de **tipo Mamdani** y sus variables clínicas están formadas por conjuntos difusos que siguen **funciones trapezoidales**. El sistema se ha creado en base a la metodología **MDL** e **incluye un sistemas de aprendizaje automático** para el afinamiento de los valores de los conjuntos difusos.

(Cosenza, 2012) expone una solución para el control de la glucemia en pacientes con diabetes de tipo 1. Para ello usa una combinación de SID formada por dos SID de **tipo Mamdani**, un SID de **tipo Takagi-Sugeno** y un **SIDRA**. Los tres primeros siguen **distribuciones triangulares y trapezoidales** para la definición de los conjuntos difusos y se han definido siguiendo la metodología **MDL**. **Ninguno** de los SID utiliza un sistema de entrenamiento para afinar su comportamiento.

(Muthukaruppan y Er, 2012) presentan una aplicación para el diagnóstico de enfermedades coronarias. El SID utilizado es de **tipo Mamdani** y utiliza **funciones triangulares** para la definición de sus conjuntos difusos. El SID se ha definido en dos fases. Primero han conseguido un sistema basado en reglas **automáticamente a partir de datos** utilizando un algoritmo de árbol de decisión (Rokach y Maimon, 2007), y posteriormente han utilizado esa solución para definir el SID cumpliendo la metodología **MDL**.

(Mago et al., 2012) ofrecen una alternativa para dar soporte a las decisiones sobre tratamientos dentales. Esta solución utiliza un SID de **tipo Mamdani** y una combinación de **funciones trapezoidales, gaussianas, asimétricas y otras propias** creadas *ad-hoc* para la representación de los conjuntos difusos. El SID se ha creado siguiendo el **MDL**. **No hacen uso** de ningún tipo de entrenamiento para mejorar el rendimiento del sistema.

(Czabanski et al., 2012) exponen un sistema para la identificación de la acidemia neonatal a partir del ritmo cardíaco del feto. Aunque la alternativa presenta distintos sistemas de clasificación, uno de ellos es un SID de **tipo Takagi-Sugeno**. Los conjuntos difusos utilizados en el sistema siguen unas **funciones trapezoidales**. El sistema se ha creado siguiendo el **MDL**

y **no utiliza** ningún sistema de entrenamiento para mejorar sus aptitudes. Posteriormente, los autores han presentado un sistema con las mismas características generales para el mismo propósito (Czabanski et al., 2013).

(Garibaldi et al., 2012) presentan un sistema para ayudar en la selección de los posibles tratamientos del cáncer de mama. Utiliza un SID de **tipo Mamdani** como núcleo, el cual utiliza **funciones trapezoidales y triangulares** para la definición de los conjuntos difusos. Se crea en base a la metodología **MDL**. Esta solución **no utiliza** ningún sistema de entrenamiento para mejorar sus elementos. Su principal característica consiste en la **generación de pequeñas perturbaciones en los conjuntos difusos para dar la sensación de variabilidad en las decisiones**, funcionalidad que mejora los resultados.

(Pal et al., 2012) ofrecen una solución para el diagnóstico de enfermedades coronarias. Esta solución utiliza un SID de **tipo Mamdani** como núcleo. Este SID utiliza conjuntos difusos con **forma trapezoidal**. El clasificador lo han construido siguiendo el MDL. En cuanto a la posibilidad de entrenamiento se refiere, **no ofrece esa opción**.

(Seki y Mizumoto, 2012) y (Seki y Mizumoto, 2013) presentan dos soluciones para el diagnóstico de la diabetes. Ambas soluciones están basadas en **sistemas de inferencia difusa con módulos de reglas de entrada única**. Aunque en las dos soluciones analizadas no se menciona el método utilizado para definir los SID, se da a entender que éstos se definen siguiendo la metodología **MDL**. En cuanto a las diferencias entre las dos soluciones, una de ellas utiliza **funciones triangulares** para la definición de los conjuntos difusos (Seki y Mizumoto, 2012), mientras que en la otra los conjuntos difusos siguen **funciones triangulares y gaussianas** (Seki y Mizumoto, 2013). En ambos casos **utilizan un algoritmo de aprendizaje** para el afinamiento de los parámetros de los SID definidos.

(Samuel et al., 2013) ofrecen una aplicación web para el diagnóstico de la fiebre tifoidea. El SID utilizado es de **tipo Mamdani** y utiliza **funciones triangulares** para representar los conjuntos difusos. Lo han creado siguiendo el **MDL**. **No utilizan** ningún tipo de entrenamiento para mejorar el rendimiento de la herramienta.

(Ghasemi et al., 2013) exponen una solución para ayudar en la segmentación de imágenes de resonancias magnéticas cerebrales. Esta solución sigue un acercamiento de **sistema de inferencia híbrida difuso-evidente**. El sistema se ha creado en base a la metodología **MDP** y sus conjuntos difusos siguen **funciones triangulares**.

(Beloufa y Chikh, 2013) ofrecen una alternativa para el diagnóstico de la diabetes. El SID utilizado es de **tipo Mamdani**, con la peculiaridad de que utiliza una **función de desfuzzificación de máximo**, lo que hace que el sistema se quede únicamente con la información de la mejor de las reglas activadas. Los conjuntos difusos siguen **funciones triangulares**. En cuanto a su creación, se genera siguiendo la metodología **MDP**.

(Castanho et al., 2013) ofrecen una alternativa para la predicción del estado patológico del cáncer de próstata. El núcleo de este sistema es un SID de **tipo Takagi-Sugeno** cuyos conjuntos difusos siguen **funciones trapezoidales**. El sistema se ha creado siguiendo el **MDP**.

(de Brito et al., 2013) presentan una solución para medir el nivel de sufrimiento o aflicción de pacientes con trastorno dismórfico corporal. Esta alternativa utiliza un SID de **tipo Mamdani** como componente principal. Las variables clínicas se han modelado utilizando conjuntos difusos que siguen **funciones trapezoidales y triangulares**. Este sistema se ha creado siguiendo el **MDL** y **no utiliza** ningún tipo de sistema de aprendizaje para mejorar su rendimiento.

(Pawade et al., 2013) presentan un sistema para la estimación de la dosis de anestesia necesaria para una persona a partir de una serie de parámetros. Esta solución utiliza un SID de **tipo Mamdani** como elemento principal que se ha creado en base a la metodología **MDL**. El SID utiliza **funciones gaussianas** para representar los conjuntos difusos de las variables clínicas. Esta solución **no utiliza** ningún sistema de aprendizaje automático para mejorar el funcionamiento.

(Lai et al., 2013) ofrecen una solución para la detección de episodios de hipoglucemia en pacientes con diabetes. El núcleo de esta solución es un SID de **tipo Mamdani** cuyas variables clínicas se han modelado utilizando **funciones con forma de campana**. El SID se ha creado siguiendo la metodología **MDL**. Para mejorar el rendimiento del SID **utilizan un sistema de aprendizaje automático**. Estos mismos autores han realizado mejoras en el sistema de entrenamiento en posteriores trabajos (Lai et al., 2014).

(Marateb et al., 2014) exponen una alternativa para identificar casos de microalbuminuria en pacientes que sufren diabetes tipo 2. El elemento principal de la solución es un SID de **tipo Mamdani** cuyos conjuntos difusos siguen **funciones gaussianas y sigmoideas**. El sistema se ha construido de forma híbrida. Por un lado, los valores lingüísticos de todas las variables clínicas, así como sus conjuntos difusos correspondientes, se han definido siguiendo la metodología **MDL**. Las reglas, sin embargo, se han creado siguiendo la metodología **MDP**.

(Bárdossy et al., 2014) presentan un sistema para la clasificación de taquiarritmias a partir del análisis de electrocardiogramas. El SID utilizado en la solución es de **tipo Takagi-Sugeno** y los conjuntos difusos del sistema siguen **funciones sigmoideas**. En cuanto a su creación, el clasificador se ha generado siguiendo el **MDP**.

(Chourasia et al., 2014) exponen un sistema para calcular el bienestar del feto analizando su ritmo cardíaco. Para ello proponen un SID de **tipo Mamdani** cuyos conjuntos difusos se han creado utilizando **funciones gaussianas**. El sistema se ha creado bajo la metodología **MDL** y **no se utiliza** ningún sistema de aprendizaje automático para mejorar las prestaciones del sistema. Su principal característica es que **utilizan LD-2** en la representación del SID.

(Sanz et al., 2014) ofrecen un sistema para el diagnóstico de enfermedades cardiovasculares. Este sistema utiliza un SID de **tipo Mamdani**, pero su principal característica es que **utiliza conjuntos difusos intervalo-valorados**, los cuales son un caso particular de la LD-2. Los conjuntos difusos utilizados siguen **funciones triangulares**. El SID se ha creado siguiendo la metodología **MDP**.

(Dennis y Muthukrishnan, 2014) exponen una alternativa que se utiliza en problemas de diagnóstico de distintas enfermedades, tales como problemas cardíacos, cáncer de mama y diabetes. El núcleo del sistema es un SID de **tipo Mamdani** que se crea siguiendo el **MDP**. Los SID creados utilizan conjuntos difusos creados utilizando **funciones triangulares**.

(Miranda y Felipe, 2015) presentan una solución para categorizar los grados de cáncer de mama. El SID utilizado es de **tipo Mamdani** y los conjuntos difusos siguen **funciones polinómicas**. Los conjuntos difusos se han creado siguiendo el **MDP**, pero las reglas del sistema se han creado bajo la metodología **MDL**.

(Nguyen et al., 2015) presentan una alternativa tanto para el diagnóstico del cáncer de mama como para el diagnóstico de enfermedades coronarias. Para ello utilizan un SID de **tipo Takagi-Sugeno** cuyos conjuntos difusos siguen **funciones gaussianas**. El sistema lo han creado siguiendo la metodología **MDP**. Una de las principales características de esta alternativa es que **utiliza LD-2**.

(Andrei et al., 2015) exponen un sistema para evaluar el estado de los pacientes con lumbosacralgia. Esta solución utiliza un SID de **tipo Takagi-Sugeno** y las variables clínicas del SID están formadas por conjuntos difusos que siguen **funciones trapezoidales**. Esta solución se ha creado siguiendo la metodología **MDL** y **no presenta** ningún sistema de aprendizaje automático para mejorar su rendimiento.

(Silveira y de Barros, 2015) ofrecen un sistema para analizar el riesgo de dengue en diversas zonas o territorios. Para ello hacen uso de un SID de **tipo Takagi-Sugeno** diseñado de forma que las variables clínicas que utiliza siguen **funciones trapezoidales**. El SID de la solución se ha creado siguiendo la metodología **MDP**.

(Nuryani et al., 2015) exponen una solución para detectar fibrilaciones en el atrio del corazón interpretando las ondas de un electrocardiograma. Esta solución es de **tipo Mamdani** y utiliza **funciones gaussianas** para la composición de sus variables. Aunque el SID se ha creado siguiendo la metodología **MDL**, **utilizan un sistema de aprendizaje automático** para mejorar ciertas características.

(Doctor et al., 2016) presentan un sistema para la regulación automática de la anestesia. El sistema utiliza un SID de **tipo Mamdani** cuyas variables clínicas están formadas por **funciones trapezoidales**. El SID se ha creado siguiendo la metodología **MDL**. Aunque **no hay evidencia** de que se utilice algún sistema para refinar el funcionamiento del SID, cabe destacar que **utilizan LD-2** y que el SID se **retroalimenta** con sus propios resultados.

(Al-Ebbini et al., 2016) presentan una solución para la selección de pacientes a los que realizar un trasplante de pulmón. Esta solución utiliza dos SID, uno de **tipo Mamdani** y otro de **tipo Takagi-Sugeno**. Las variables de ambos sistemas se han construido utilizando funciones **constantes** y **de campana**. La metodología seguida para crear los SID ha sido la **MDP**.

(de Bruin et al., 2016) exponen una solución para la detección de infecciones en casos que normalmente se confunden con normales. Para ello utiliza un SID de **tipo Takagi-Sugeno** donde las variables clínicas se han definido mediante **funciones trapezoidales**. Todo el sistema se ha creado siguiendo la metodología **MDL** y **no hay evidencia** de que se haya utilizado un sistema de aprendizaje para mejorar las características del SID.

(Salgado et al., 2016) ofrecen un sistema para la correcta administración de medicamentos vasoconstrictores. El núcleo de este sistema es un SID de **tipo Takagi-Sugeno** que utiliza **funciones gaussianas** para la representación de las variables. El SID de esta solución se ha creado siguiendo la metodología **MDP**.

(Büyükcavcu et al., 2016) presentan una compleja solución para analizar los factores de riesgo del cáncer de mama. Uno de los componentes de la solución es un SID de **tipo Mamdani**. Las variables clínicas que se han tenido en cuenta en este SID se han modelado utilizando **funciones triangulares**. El SID se ha construido siguiendo la metodología **MDL** y **no hay evidencia** de que se haya utilizado algún sistema de aprendizaje automático para mejorar su funcionamiento.

(Oluwagbemi et al., 2016) ofrecen un sistema para el diagnóstico del ébola. El componente principal es un SID de **tipo Mamdani** cuyas variables clínicas han sido diseñadas utilizando **funciones triangulares**. La metodología seguida para crear el SID se enmarca dentro del **MDL** y **no hay evidencia** de que se haya utilizado algún sistema adicional para mejorar las prestaciones del SID.

(Fialho et al., 2016) exponen un sistema para la predicción de la mortalidad por shock séptico. El componente principal de este sistema es un **sistema difuso probabilístico**. Las FP que se han utilizado en este SID son **funciones gaussianas** y la solución expuesta se ha creado siguiendo la metodología **MDP**.

(Meza-Palacios et al., 2017) presentan una solución para el control de posibles nefropatías en pacientes con diabetes mellitus de tipo 2. Esta solución se basa en el uso de un SID de **tipo Mamdani**. Las variables clínicas del SID se han definido utilizando **funciones trapezoidales** y el SID se ha desarrollado siguiendo la metodología **MDL**. **No hay constancia** de que se haya usado algún sistema para la mejora de las prestaciones del SID.

(Gorzalczany y Rudzinski, 2017) proponen un sistema para el diagnóstico de distintas enfermedades como el cáncer de mama, la diabetes y los problemas coronarios. Para ello propone un SID de **tipo Mamdani** que utiliza **funciones trapezoidales** para representar las variables clínicas. El SID se ha construido siguiendo la metodología **MDP**.

## 8.2. Síntesis y comparativa

22 de las 37 soluciones analizadas hacen uso de SID creados siguiendo una metodología MDL, esto es, se crean manualmente a partir de información de expertos y/o de bibliografía (ver tabla 8.4). Los SID de otras 12 soluciones se han creado automáticamente a partir de los datos siguiendo una metodología MDP. Por último, en 3 de las alternativas evaluadas los SID se han creado siguiendo un acercamiento mixto. El hecho de que la mayoría de los sistemas analizados sigan la metodología MDL puede indicar que la creación automática de SID no resulta tan sencilla como la de otros sistemas de clasificación, tales como las redes neuronales artificiales y las máquinas de soporte vectorial. Cabe destacar que de aquellos sistemas que se han creado siguiendo el MDL sólo 4 utilizan un sistema de aprendizaje automático para mejorar sus prestaciones.

El tipo de SID más utilizado es Mamdani o alguna de sus variantes (24 de 37) (ver tablas 8.5 y 8.6). Otras 10 soluciones hacen uso de un SID de tipo Takagi-Sugeno (o alguna de sus variantes). Cabe señalar que en una de las alternativas combinan dos SID de tipo Mamdani con uno de tipo Takagi-Sugeno y un SIDRA. El hecho de que la mayoría de los SID sean de tipo Mamdani probablemente venga dado por su cercanía al razonamiento humano y por su facilidad para la definición e interpretación.

En cuanto a las funciones de pertenencia utilizadas para representar los conjuntos difusos, las más extendidas son las funciones trapezoidales y triangulares, llegando a utilizarse en 15 de los 37 sistemas analizados las primeras y en 13 de 37 casos las segundas (ver tablas 8.7, 8.8 y 8.9). Por otro lado, cabe señalar que sólo en 3 de las soluciones utilizan un SID con conjuntos difusos de factura propia.

Por último, hay que señalar que en 4 de las alternativas estudiadas utilizan LD-2.

<b>MDL</b>	<b>MDP</b>	<b>Mixto</b>
(Khatibi y Montazer, 2010)	(Ghasemi et al., 2013)	(Muthukaruppan y Er, 2012)
(Mahmoodabadi et al., 2010)	(Beloufa y Chikh, 2013)	(Marateb et al., 2014)
(Rabbi et al., 2010)	(Castanho et al., 2013)	(Miranda y Felipe, 2015)
(Esposito et al., 2011)	(Bárdossy et al., 2014)	
(Cosenza, 2012)	(Sanz et al., 2014)	
(Mago et al., 2012)	(Dennis y Muthukrishnan, 2014)	
(Czabanski et al., 2012, 2013)	(Nguyen et al., 2015)	
(Garibaldi et al., 2012)	(Silveira y de Barros, 2015)	
(Pal et al., 2012)	(Al-Ebbini et al., 2016)	
(Seki y Mizumoto, 2012, 2013)	(Salgado et al., 2016)	
(Samuel et al., 2013)	(Fialho et al., 2016)	
(de Brito et al., 2013)	(Gorzalczany y Rudzinski, 2017)	
(Pawade et al., 2013)		
(Lai et al., 2013, 2014)		
(Chourasia et al., 2014)		
(Andrei et al., 2015)		
(Nuryani et al., 2015)		
(Doctor et al., 2016)		
(de Bruin et al., 2016)		
(Büyükcavcu et al., 2016)		
(Oluwagbemi et al., 2016)		
(Meza-Palacios et al., 2017)		
<b>22</b>	<b>12</b>	<b>3</b>

**Tabla 8.4.** Resumen de la metodología de creación seguida por las distintas soluciones analizadas

Mamdani	Takagi-Sugeno	Clasificador difuso relacional
(Rabbi et al., 2010)	(Cosenza, 2012)	(Mahmoodabadi et al., 2010)
(Esposito et al., 2011)	(Czabanski et al., 2012, 2013)	
(Cosenza, 2012)	(Castanho et al., 2013)	
(Muthukaruppan y Er, 2012)	(Bárdossy et al., 2014)	
(Mago et al., 2012)	(Nguyen et al., 2015)	
(Garibaldi et al., 2012)	(Andrei et al., 2015)	
(Pal et al., 2012)	(Silveira y de Barros, 2015)	
(Samuel et al., 2013)	(Al-Ebbini et al., 2016)	
(Beloufa y Chikh, 2013)	(de Bruin et al., 2016)	
(de Brito et al., 2013)	(Salgado et al., 2016)	
(Pawade et al., 2013)		
(Lai et al., 2013, 2014)		
(Marateb et al., 2014)		
(Chourasia et al., 2014)		
(Sanz et al., 2014)		
(Dennis y Muthukrishnan, 2014)		
(Miranda y Felipe, 2015)		
(Nuryani et al., 2015)		
(Doctor et al., 2016)		
(Al-Ebbini et al., 2016)		
(Büyükcavcu et al., 2016)		
(Oluwagbemi et al., 2016)		
(Meza-Palacios et al., 2017)		
(Gorzalczany y Rudzinski, 2017)		
<b>24</b>	<b>10</b>	<b>1</b>

**Tabla 8.5.** Resumen del tipo de SID utilizado por las distintas soluciones analizadas

Sistema de inferencia híbrida difuso-evidente	Sistema de inferencia difusa con módulos de reglas de entrada única	Sistema difuso probabilístico
(Khatibi y Montazer, 2010) (Ghasemi et al., 2013)	(Seki y Mizumoto, 2012, 2013)	(Fialho et al., 2016)
<b>2</b>	<b>1</b>	<b>1</b>

**Tabla 8.6.** Resumen del tipo de SID utilizado por las distintas soluciones analizadas (continuación)

<b>Triangular</b>	<b>Trapezoidal</b>
(Rabbi et al., 2010)	(Rabbi et al., 2010)
(Cosenza, 2012)	(Esposito et al., 2011)
(Muthukaruppan y Er, 2012)	(Cosenza, 2012)
(Garibaldi et al., 2012)	(Mago et al., 2012)
(Seki y Mizumoto, 2012, 2013)	(Czabanski et al., 2012, 2013)
(Samuel et al., 2013)	(Garibaldi et al., 2012)
(Ghasemi et al., 2013)	(Pal et al., 2012)
(Beloufa y Chikh, 2013)	(Castanho et al., 2013)
(de Brito et al., 2013)	(de Brito et al., 2013)
(Sanz et al., 2014)	(Andrei et al., 2015)
(Dennis y Muthukrishnan, 2014)	(Silveira y de Barros, 2015)
(Büyükavcu et al., 2016)	(Doctor et al., 2016)
(Oluwagbemi et al., 2016)	(de Bruin et al., 2016)
	(Meza-Palacios et al., 2017)
	(Gorzalczany y Rudzinski, 2017)
<b>13</b>	<b>15</b>

**Tabla 8.7.** Resumen del tipo de FP utilizado por las distintas soluciones analizadas

<b>Gaussiana</b>	<b>Sigmoidea</b>	<b>Polinómica</b>
(Mago et al., 2012)	(Mago et al., 2012)	(Mago et al., 2012)
(Seki y Mizumoto, 2012, 2013)	(Marateb et al., 2014)	(Miranda y Felipe, 2015)
(Pawade et al., 2013)	(Bárdossy et al., 2014)	
(Marateb et al., 2014)		
(Chourasia et al., 2014)		
(Nguyen et al., 2015)		
(Nuryani et al., 2015)		
(Salgado et al., 2016)		
(Fialho et al., 2016)		
<b>9</b>	<b>3</b>	<b>2</b>

**Tabla 8.8.** Resumen del tipo de FP utilizado por las distintas soluciones analizadas (continuación)

<b>Campana</b>	<b>Constante</b>	<b>Propia</b>
(Lai et al., 2013, 2014)	(Al-Ebbini et al., 2016)	(Khatibi y Montazer, 2010)
(Al-Ebbini et al., 2016)		(Mahmoodabadi et al., 2010)
		(Mago et al., 2012)
<b>2</b>	<b>1</b>	<b>3</b>

**Tabla 8.9.** Resumen del tipo de FP utilizado por las distintas soluciones analizadas (continuación)



## Frameworks para el desarrollo de Sistemas de Inferencia Difusa

En el siguiente capítulo se realiza una revisión de los *frameworks* más completos especificados por la *European Society for Fuzzy Logic and Technology* (EUSFLAT, 2017) y que se han podido encontrar en Internet para el desarrollo específico de SID siguiendo la metodología MDL: *Fuzzy Logic Toolbox de Matlab*, *Xfuzzy 3.0*, *FuzzyTech*, *GUAJE FUZZY* y *Linguistic Fuzzy Logic Controller*.

Las distintas soluciones se han revisado en base a *características de los SID que permiten crear los frameworks*, así como a sus propias *características para el desarrollo de SID*. Así, dentro de las **características de los SID que permiten crear** se engloban aquellas características que no pertenecen a las herramientas en sí, sino a la libertad que dan al usuario para definir SID. Dentro de estas características se encuentran el *tipo de SID* que permiten definir, los *tipos de FP* que ofrecen, así como el abanico de *funciones de implicación, agregación y desfuzzificación* que incluyen. Las **características para el desarrollo de SID**, por su lado, inciden sobre las funcionalidades que ofrecen las herramientas para llevar a cabo los pasos del diseño de un SID siguiendo la metodología MDL. Dentro de estas características destacan la *definición de las variables de entrada y salida*, la *especificación de las funciones de implicación, agregación y desfuzzificación*, la *definición de las reglas*, así como las opciones de *simulación y exportación* del sistema resultante. La oferta de un *sistema de aprendizaje automático* para la mejora de los SID diseñados también es una característica a tener en cuenta.

A continuación se realiza la revisión de los *frameworks* para el desarrollo de SID. Para cada uno de los *frameworks* analizados, primero se describe brevemente y posteriormente se analiza incidiendo especialmente en las características explicadas. Al final se ofrece una síntesis de la comparativa del estudio.

## 9.1. Fuzzy Logic Toolbox de Matlab

Matlab (Mathworks, 2017a) es un entorno interactivo para el cálculo, la visualización y la programación numéricos. Entre los distintos *frameworks* que contiene Matlab se encuentra *Fuzzy Logic Toolbox* (FLT) (Mathworks, 2017b). FLT contiene funciones y herramientas gráficas para el análisis, diseño y simulación de sistemas basados en la lógica difusa. De esta forma, el usuario puede modelar el comportamiento de sistemas complejos utilizando simples reglas lógicas, para posteriormente implementar esas reglas en un SID.

FLT permite tanto el diseño de SID de tipo **Mamdani** como **Takagi-Sugeno**.

Ofrece una amplia gama de tipos de FP, entre las que se encuentran las funciones **triangular**, **trapezoidal**, dos tipos de funciones basadas en la **distribución Gaussiana** (de una única curva y con una curva Gaussiana diferente en cada extremo), **de campana**, tres funciones basadas en la **función sigmoidea** (sigmoidea normal abierta en alguno de los extremos, diferencia entre dos funciones sigmoideas y producto entre las mismas) y otros tres tipos de funciones basadas en las **polinómicas** (Z si es abierta a la izquierda, S si es abierta a la derecha y Pi si es cerrada en ambos extremos). Aparte de todas estas opciones, este *framework* también ofrece la posibilidad de que el usuario defina sus **propias** FP.

En cuanto a las funciones de implicación, FLT ofrece la posibilidad de utilizar tanto un operador de **mínimo** como de **producto**.

En el caso de las funciones agregación, FLT permite la utilización de las funciones **máximo**, **suma probabilística** y **suma acotada**.

En lo que a las funciones de desfuzzificación se refiere, FLT dispone de un abanico de cinco alternativas: **cálculo del centroide**, **bisector**, **media de los máximos**, **mayor de los máximos** y **mínimo de los máximos**.

Para la definición de las variables de entrada y salida FLT ofrece un **soporte gráfico** compuesto por un formulario. Este formulario permite al usuario ir introduciendo los distintos parámetros de las variables y FP y ver los resultados gráficamente en tiempo real.

La selección de las funciones de implicación, agregación y desfuzzificación también se realiza de **forma gráfica** mediante una serie de menús desplegables.

El *framework* FLT utiliza un **soporte gráfico** para la definición de las reglas. Concretamente se utiliza un formulario. Una vez que están definidas las reglas, se permite visualizar la relación entre las variables de entrada y las de salida mediante unos gráficos en tres dimensiones.

FLT **ofrece la posibilidad de simular** el funcionamiento del SID diseñado dentro de la propia herramienta. La funcionalidad de simulación permite al usuario introducir distintos valores de entrada y observar todo el proceso de inferencia y el valor o los valores de salida.

Este *framework* también **permite la exportación** del SID resultante, aunque ésta sólo lo realiza en el propio lenguaje de Matlab. Por lo tanto, es

necesario disponer un módulo de ejecución de Matlab para poder utilizarlo en otras soluciones.

Por último, en lo que al sistema de aprendizaje automático se refiere, FLT **no dispone de esta opción**.

Como característica extra cabe destacar que este *framework* de Matlab permite también **definir SIDRA**. Es una herramienta diferente de la utilizada para definir los SID ya que los SIDRA se crean automáticamente a partir de un fichero de casos que el usuario especifica. El usuario debe especificar una serie de características de los SIDRA

## 9.2. Xfuzzy 3.0

Xfuzzy 3.0 es un entorno de desarrollo para sistemas basados en lógica difusa (Moreno, 2002). Las herramientas que ofrece cubren todas las etapas de desarrollo del sistema, desde su descripción o especificación, hasta la implementación final, pasando por las fases de ajuste y verificación.

Este *framework* permite la definición tanto de SID de tipo **Mamdani** como **Takagi-Sugeno**. De hecho, permite la creación de sistemas complejos que incluyen la **combinación de varios SID**.

En cuanto a las posibles FP que se pueden utilizar, aparte de permitir que el usuario especifique sus **propias** FP, también ofrece una amplia gama de FP prediseñadas, entre las que se encuentran, dos tipos de **funciones triangulares**, una **función trapezoidal**, una **función con forma de campana**, una **función sigmoidea**, una **función rectangular** (variante de la trapezoidal) y una **función constante**.

Xfuzzy 3.0 también ofrece ocho funciones de implicación diferentes: **Sienes-Rescher**, **Mizumoto**, **Lukasiewicz**, **Dubois-Prade**, **Zadeh**, **Goguen**, **Gödel** e **implicación afilada**.

En lo que a **funciones de agregación** se refiere, Xfuzzy 3.0 ofrece cuatro funciones: **máximo**, **suma probabilística**, **suma acotada** y **suma drástica**.

Xfuzzy 3.0 dispone de una gama de diez funciones de desfuzzificación compuesta por las funciones **cálculo del centroide**, **primer máximo**, **último máximo**, **media de los máximos**, **media difusa**, **media difusa ponderada**, **método de calidad**, **método de calidad gamma**, **etiqueta mayor** y **Takagi-Sugeno**.

Este *framework* dispone de una funcionalidad con **soporte gráfico** para poder definir las variables de entrada y salida, así como las FP de las que están formadas. Para ello ofrece un formulario que permite ir viendo en tiempo real los cambios que se van haciendo en los distintos parámetros.

La selección de las funciones de implicación, agregación y desfuzzificación se realiza de **forma gráfica** utilizando una serie de menús desplegados.

Respecto a la definición de las reglas, Xfuzzy 3.0 permite tanto la definición de reglas utilizando un **soporte gráfico** como un **soporte textual**. Para

definir de forma gráfica las reglas se puede utilizar una tabla (o lo que podría entenderse como texto estructurado) o una matriz (sólo para los SID con dos variables de entrada y una de salida).

Xfuzzy 3.0 **dispone de una funcionalidad para la simulación** de los SID diseñados. Con esta funcionalidad, en base a los valores de entrada, Xfuzzy 3.0 muestra tanto la evolución de las variables de salida como de todos los procesos de inferencia intermedios.

Este *framework* también **ofrece la opción de exportación**. La exportación se realiza a diferentes lenguajes de alto nivel, tales como C, C++, Java, así como a otros lenguajes específicos de control de sistemas, como VHDL y SysGen.

En lo que a la inclusión de un sistema de aprendizaje automático se refiere, Xfuzzy 3.0 **ofrece esta opción** mediante una funcionalidad que dispone de 17 algoritmos de aprendizaje diferentes clasificados en cuatro familias de algoritmos: algoritmos de descenso de gradiente, algoritmos de gradiente conjugado, algoritmos de segundo orden y algoritmos que no utilizan derivadas. Los parámetros ajustables son los correspondientes a los valores de los conjuntos difusos del modelo, de los cuales el usuario puede seleccionar los que desea afinar.

Una de las características únicas del *framework* Xfuzzy 3.0 es que **permite el diseño de sistemas difusos complejos jerárquicos**, mediante la combinación de dos o más SID simples. Además, todas las funcionalidades mencionadas hasta el momento, tales como la verificación, entrenamiento y ejecución son directamente aplicables a estos modelos complejos.

### 9.3. FuzzyTech

FuzzyTech es un entorno de desarrollo para la definición, simulación y exportación de aplicaciones difusas para el control de sistemas (INFORM, 2017). FuzzyTech se ha basado en el estándar para el desarrollo de sistemas industriales de lógica difusa IEC 1131-7 (IEC, 1997), lo que le ha otorgado un mayor alcance.

FuzzyTech permite la definición de SID de tipo **Mamdani** y de tipo **Takagi-Sugeno**.

Además de las FP **constantes**, FuzzyTech también permite el uso de FP **triangulares, trapezoidales y gaussianas** y que el usuario defina sus **propias funciones**.

En lo que a la implicación se refiere, FuzzyTech ofrece únicamente la función de **producto**.

El proceso de agregación se puede realizar mediante las funciones de **máximo y suma acotada**.

Por último, para llevar a cabo el proceso de desfuzzificación FuzzyTech ofrece las funciones de **centro del área mayor** y cuatro versiones de la **función de máximo**.

FuzzyTech incluye una funcionalidad con **soporte gráfico** para definir variables de entrada y salida. Mediante un formulario se indican los valores lingüísticos de las variables y mediante una tabla se establecen las FP y los parámetros de las mismas. Los cambios se visualizan en tiempo real de forma gráfica.

Las funciones de implicación, agregación y desfuzzificación se especifican de **forma gráfica** mediante una tabla.

FuzzyTech permite definir las reglas de los SID en **soporte textual y gráfico**. Para el soporte gráfico dispone de un editor en forma de hoja de cálculo y un editor matricial. Este último editor permite crear las reglas seleccionando los valores en varios menús desplegables.

Este *framework* **incluye una funcionalidad para la simulación** de los SID diseñados. La funcionalidad para la simulación permite no sólo simular el funcionamiento del SID, sino el de todo un sistema de control que utiliza el SID definido.

FuzzyTech también **incluye la posibilidad de exportar** el SID definido. Esta exportación se puede realizar a lenguajes de alto nivel Java y C, así como a los lenguajes específicos para el control de sistemas ST y FTR.

En lo que a sistemas de aprendizaje se refiere, **no se ha encontrado** evidencia de que FuzzyTech ofrezca alguna clase de herramienta para el afinamiento de los SID diseñados.

Como característica especial se puede mencionar que FuzzyTech incluye un **asistente para la creación dirigida** de sistemas difusos. Esto permite crear soluciones de forma más rápida, aunque su potencial no esté a la altura de las soluciones que se crean manualmente.

## 9.4. GUAJE FUZZY

GUAJE FUZZY es un *framework* para la construcción de SID interpretables y eficientes (Alonso y Magdalena, 2011). Para ello combina varias herramientas existentes para el tratamiento y la visualización de SID. Este *framework* permite construir SID siguiendo las metodologías MDL y MDP, aunque a continuación se explican sus características en base a la primera de ellas.

GUAJE FUZZY permite construir SID de tipo **Mamdani** y **Takagi-Sugeno**.

En lo que a las FP se refiere, este *framework* permite utilizar **funciones triangulares y trapezoidales**.

Para el proceso de implicación, GUAJE FUZZY permite utilizar las funciones de **mínimo, producto y Lukasiewicz**.

El proceso de agregación, por su lado, se permite realizar utilizando las funciones de **máximo y suma**.

Para llevar a cabo el proceso de desfuzzificación GUAJE FUZZY ofrece las funciones **centro del área mayor, media de los máximos** y **Takagi-Sugeno**.

Las variables de entrada y salida se definen utilizando una funcionalidad con **soporte gráfico**. Esta funcionalidad ofrece varios formularios mediante los cuales el usuario establece el número de valores lingüísticos, así como las FP correspondientes. Mientras se van haciendo cambios visualmente se muestra cómo quedan compuestas las variables.

Las funciones de implicación, agregación y desfuzzificación se definen de **forma gráfica** utilizando varios menús desplegables donde los usuarios seleccionan las funciones que quieren utilizar.

Para la definición de las reglas GUAJE FUZZY utiliza una **funcionalidad gráfica** que muestra una tabla con menús desplegables para que el usuario explicita la relación entre las variables de entrada y salida.

Este *framework* **dispone de una funcionalidad para poder simular** el funcionamiento del SID diseñado. De esta forma, mientras el usuario modifica los valores de entrada esta funcionalidad muestra cómo se activan las reglas y el resultado final de esa activación.

También **es posible la exportación** del SID construido. En concreto, GUAJE FUZZY permite exportar los SID a otros formatos como FisPro, Matlab, Xfuzzy y Espresso.

En lo que a un sistema de aprendizaje automático se refiere, **no se ha encontrado** evidencia de que GUAJE FUZZY disponga de una funcionalidad para mejorar las características de los SID creados manualmente.

Por último, de este sistema cabe destacar un **asistente para la creación dirigida** de SID, así como el hecho de poder **crear SID siguiendo la metodología MDP**.

## 9.5. Linguistic Fuzzy Logic Controller

Linguistic Fuzzy Logic Controller (LFLC) es un *framework* que permite la edición y simulación de SID utilizando tanto métodos clásicos de SID como métodos de una teoría propia (Novák y Perfilieva, 2004). La base de esta teoría consiste en usar *expresiones evaluativas*, unas expresiones lingüísticas con una semántica diferente a como se utilizan en los SID clásicos.

Este *framework* permite construir SID de tipo **Mamdani** y **Takagi-Sugeno**.

LFLC permite utilizar FP de tipo **triangular**, **trapezoidal** y **polinómico**.

El proceso de implicación puede seleccionarse de un conjunto de funciones que incluye la función de **mínimo**, **producto**, **Lukasiewicz**, así como un método propio llamado **deducción lógica basada en la percepción** (Novák, 2005).

En cuanto al proceso de agregación, **no se ha encontrado evidencia** de cómo se lleva a cabo este proceso en LFLC.

Por su lado, para el proceso de desfuzzificación LFLC ofrece **tres funciones del cálculo del centroide**, una función para la **media de los máximos**

y **tres métodos para la desfuzzificación de expresiones evaluativas**, parte de su teoría (Novák, 2005).

La definición de las variables de entrada y salida se realiza de **forma gráfica** utilizando una serie de formularios. Estos formularios permiten a los usuarios modificar varias características de las variables, tales como el número de valores lingüísticos y las FP correspondientes, y ver en tiempo real el efecto que tienen esas modificaciones.

Para la definición de las funciones de implicación, agregación y desfuzzificación LFLC utiliza un **sistema gráfico** formado por varios menús desplegables.

En lo que a las reglas se refiere, LFLC ofrece una funcionalidad a medio camino entre el **sistema gráfico** y el **textual**. Las reglas se muestran y se van añadiendo en una tabla, pero la especificación de los valores lingüísticos que unen las variables de entrada y salida se realiza de forma textual escribiendo el nombre de los valores.

Este *framework* **ofrece una funcionalidad para simular** el comportamiento de los SID diseñados. Además de poder ver el funcionamiento del sistema en base a distintos valores de entrada, también permite cambiar de forma dinámica los métodos de implicación y desfuzzificación para obtener distintos resultados.

LFLC **ofrece la posibilidad de exportar** o conectar los SID creados con Matlab, de forma que se puede llamar desde Matlab a SID construidos mediante LFLC.

Aunque este *framework* **no ofrece** un sistema de aprendizaje automático para mejorar las características de los SID diseñados, sí incluye una funcionalidad para el aprendizaje automático de expresiones lingüísticas. Estas expresiones se pueden utilizar en el SID que se está diseñando.

Como característica especial cabe destacar el hecho de que LFLC permita utilizar, además de métodos conocidos de los SID, **métodos de una teoría propia** de los desarrolladores de la herramienta.

## 9.6. Síntesis y comparativa

En las tablas 9.1, 9.2 y 9.3 se muestra un resumen sobre las características de los distintos *frameworks* analizados.

Todas las herramientas estudiadas requieren de un conocimiento bastante específico sobre los SID y sus características. Todos los *frameworks* estudiados permiten crear tanto SID de tipo Mamdani como Takagi-Sugeno.

En cuanto a las FP que ofrecen los *frameworks* FLT dispone del mayor abanico de alternativas (12). Por su lado, Xfuzzy 3.0 ofrece 8 funciones, Fuzzy-Tech 5, LFLC 3 y GUAJE FUZZY 2.

En lo que a las funciones de implicación, agregación y desfuzzificación se refiere, el catálogo más amplio lo ofrece Xfuzzy 3.0. Así, dispone de 8 funciones de implicación, 4 de agregación y 10 de desfuzzificación. FLT, por su lado,

ofrece 2 funciones de implicación, 3 de agregación y 5 de desfuzzificación. El *framework* FuzzyTech dispone de una función de implicación, 2 funciones de agregación y 5 de desfuzzificación. LFLC dispone de 4 funciones de implicación (incluyendo una función propia) y 7 funciones de desfuzzificación, 3 de las cuales son también propias. Por último, GUAJE FUZZY dispone de un abanico de 3 funciones de implicación, 2 funciones de agregación y 3 funciones de desfuzzificación.

En cuanto a las funcionalidades ofrecidas por estas herramientas para la definición de las variables de entrada y salida, todas ellas ofrecen soluciones gráficas. Mientras que FuzzyTech utiliza una tabla para especificar las funciones, todas las demás soluciones utilizan menús desplegables.

En referencia a los modos de edición de reglas que los *frameworks* ofrecen, tanto Xfuzzy 3.0 como FuzzyTech ofrecen la posibilidad de definir las reglas en forma textual. Ambos *frameworks* también ofrecen funcionalidades parecidas para definir las reglas en soporte gráfico mediante herramientas en forma de tabla y matriz. LFLC ofrece un sistema híbrido gráfico-textual, de forma que las reglas se muestran en forma de tabla, pero la especificación de los valores lingüísticos se hace de forma textual. FLT, y GUAJE FUZZY únicamente ofrecen la opción de definir las reglas de forma gráfica utilizando una herramienta con formularios y una serie de menús desplegables respectivamente. En todos los casos las reglas se definen manualmente.

De cara a la simulación, todos los *frameworks* analizados ofrecen una funcionalidad que la permite. Xfuzzy 3.0 y FuzzyTech van un paso más allá y no sólo posibilitan la simulación de los SID diseñados, sino que también permiten la simulación de los sistemas finales donde los SID van integrados.

En cuanto a la ejecución de los SID desarrollados, todos los *frameworks* incluyen esta opción. Sin embargo, FLT y LFLC únicamente permiten la exportación en el lenguaje de Matlab, mientras que Xfuzzy 3.0 y FuzzyTech permiten la exportación a lenguajes de programación de propósito general, tales como C y Java, así como a lenguajes específicos de control de sistemas. GUAJE FUZZY, por su lado, permite la exportación en lenguajes específicos de otros *frameworks* para el análisis y ejecución de SID, tales como los ya vistos Matlab y Xfuzzy.

Por último, cabe señalar que Xfuzzy 3.0 es el único de los *frameworks* que ofrece la posibilidad de mejorar las prestaciones de los SID diseñados mediante el uso de algoritmos de aprendizaje automático.

<b>Framework</b>	<b>Tipos de SID</b>	<b>FP</b>	<b>Func. implicación</b>	<b>Func. agregación</b>
Fuzzy Logic Toolbox	M y TS	TA, TP, G, S, CP, PL y PR	MN y P	MX, SP y SA
Xfuzzy 3.0	M y TS	TA, TP, CP, S, R, CT y PR	SR, MZ, LK, DP, ZD, GG, GD e IA	MX, SP, SA y SD
FuzzyTech	M y TS	CT, TA, TP, G y PR	MN y P	MX y SA
GUAJE FUZZY	M y TS	TA y TP	MN, P y LK	MX y S
LFLC	M y TS	TA, TP y PL	MN, P, LK y método propio	-

**Tabla 9.1.** Comparativa sobre los *frameworks* para la definición de SID. Legenda: *Tipos de SID*: M - Mamdani, TS - Takagi-Sugeno. *Func. pertenencia*: TA - Triangular, TP - Trapezoidal, G - Gaussiana, S - Sigmoidea, CP - De campana, PL - polinómica, PR - Propia, CT - Constante, R - Rectangular. *Func. implicación*: MN - Mínimo, P - Producto, MZ - Mizumoto, LK - Lukasiewicz, DP - Dubois-Prade, ZD - Zadeh, GG - Gogen, GD - Gödel, IA - Implicación afilada, SR - Sienes-Rescher. *Func. agregación*: MX - Máximo, S - Suma, SA - Suma acotada, SP - Suma probabilística, SD - Suma drástica.

<b>Framework</b>	<b>Func. defuzzificación</b>	<b>Def. variables</b>	<b>Def. funciones</b>	<b>Def. reglas</b>
Fuzzy Logic Toolbox	C, B, MM, YM y NM	Soporte gráf. (formularios)	Soporte gráf. (menús despl.)	Soporte gráf. (formularios)
Xfuzzy 3.0	C, PM, UM, MM, MD, MDP, MC, MCG, EM y TS	Soporte gráf. (formularios)	Soporte gráf. (menús despl.)	Soporte gráf. (tabla y editor matricial)
FuzzyTech	CAM y 4 M	Soporte gráf. (formularios y tablas)	Soporte gráf. (tabla)	Soporte gráf. (hoja de cálculo y editor matricial) y soporte text.
GUAJE FUZZY	CAM, MM y TS	Soporte gráf. (formularios)	Soporte gráf. (menús despl.)	Soporte gráf. (menús despl.)
LFLC	3 C, MM, 3 métodos propios	Soporte gráf. (formularios)	Soporte gráf. (menús despl.)	Mixto (tablas y texto)

**Tabla 9.2.** Comparativa sobre los *frameworks* para la definición de SID (continuación). Leyenda: *Func. defuzzificación*: C - Centroide, B - Bisector, CAM - Centro del área mayor, M - Máximo, PM - Primer máximo, UM - Último máximo, MM - Media de los máximos, YM - Mayor de los máximos, NM - Menor de los máximos, MD - Media difusa, MDP - Media difusa ponderada, MC - Método de calidad, MCG - Método de calidad gamma, EM - Etiqueta mayor, TS - Takagi-Sugeno.

<b>Framework</b>	<b>Simulación</b>	<b>Ejecución</b>	<b>Entrenamiento</b>
Fuzzy Logic Toolbox	Sí	Sí (Matlab)	No
Xfuzzy 3.0	Sí	Sí (C, C++, Java, VHDL y SysGen)	Sí
FuzzyTech	Sí	Sí (Java, C, ST, FTR)	No
GUAJE FUZZY	Sí	Sí (Fispro, Matlab, Xfuzzy y Espresso)	No
LFLC	Sí	Sí (Matlab)	No

**Tabla 9.3.** Comparativa sobre los *frameworks* para la definición de SID (continuación)

Solución propuesta



La Parte IV **recoge el trabajo realizado** para desarrollar Sistemas de Inferencia Difusa siguiendo métodos y técnicas recogidas dentro de los Métodos de Desarrollo Dirigido por Modelos, así como las evaluaciones realizadas para probar la validez de la solución propuesta.

El capítulo 10 presenta el proceso de modelado de Diagnósticos Diferenciales y Sistemas de Inferencia Difusa que se basa en el uso de técnicas y herramientas de Métodos de Desarrollo Dirigido por Modelos. También explica los distintos procesos de transformación automática utilizados.

El capítulo 11 presenta las herramientas desarrolladas para poder definir modelos de Diagnósticos Diferenciales y de Sistemas de Inferencia Difusa, así como para trabajar sobre los modelos de Sistemas de Inferencia Difusa definidos.

Por último, el capítulo 12 presenta los procesos de evaluación seguidos para evaluar el trabajo realizado.



## Modelado de DD y SID

En este capítulo se presenta el proceso de modelado de DD y SID, así como los procesos de transformación automática que se han definido. Todo esto se ha llevado a cabo utilizando las técnicas y herramientas de MDDM incluidas en el EMP.

Supóngase que se quiere representar mediante un SID el DD ficticio mostrado en la figura 10.1, cuya única finalidad es facilitar la comprensión de los conceptos explicados en este capítulo. En las columnas están dispuestas las variables clínicas, mientras que en las filas se pueden observar los trastornos (o los conjuntos de trastornos). Las celdas contienen los valores lingüísticos de las variables clínicas con los que se relacionan los trastornos.

Diagnóstico diferencial de un trastorno			
	Gasometría	Amonio (N < 90 µg/dl)	Insulina / Glucemia (N < 0,34)
Trastornos de la cetogénesis	< 6,9	Normal o leve	Según ayuno
Trastornos de la beta oxidación de los ácidos grasos	Bajo o < 7,2	200 - 700 µmol/l	Siempre > 0,4
Aciduria glutárica tipo 2	7,0 - 7,2	E	> 0,34

**Figura 10.1.** Muestra de un DD

Este DD, así como gran parte de los DD que aparecen en las GC, son fácilmente comprensibles por un humano (concretamente un médico) pero son difíciles de adaptar a un esquema general que permita su informatización (López-Cuadrado et al., 2015). Esto es debido a una falta de estandarización, tanto a nivel de estructura como a nivel de los conceptos que se utilizan. Así, existen DD donde las variables clínicas se especifican en las columnas y los trastornos en las filas y otros DD donde se estructuran al contrario. De la misma forma, no es difícil encontrar valores en las celdas del DD que no sean concretos. Por ejemplo, en el DD de la figura 10.1 el valor de la celda que relaciona el trastorno *Aciduria glutárica tipo 2* y la variable clínica *Gasometría* es *Normal o < 7,4*. Viendo esto no se puede saber si el valor *Normal* está

por debajo o por encima de 7,4. También puede faltar información en el DD, como ocurre con la variable *Amonio*. Indican que el valor de normalidad está por debajo de  $90 \mu g/dl$ , mientras que en la única celda que tiene un valor numérico para dicha variable el valor indicado está expresado en  $\mu mol/l$ . Con todo esto, se quiere señalar que antes de poder tan siquiera empezar a informatizar DD, normalmente se requiere realizar un trabajo previo completando y homogeneizando los valores de los DD, trabajo para lo cual es indispensable disponer de la ayuda de un experto médico en el ámbito del DD. En la figura 10.2 se muestra el DD versión corregida y homogeneizada del DD de la figura 10.1. El proceso de homogeneización de un DD está fuera del alcance de la tesis y en adelante, cuando se refiera a un DD, se supondrá que éste ya está homogeneizado y preparado para ser informatizado.

Diagnóstico diferencial de un trastorno			
	Gasometría (7,35 < pH normal < 7,45)	Amonio (Normal < 50 $\mu mol/l$ , Normal < 90 $\mu g/dl$ , Niño > 30 días)	Insulina / Glucemia (Normal < 0,34)
Trastornos de la cetogénesis	Muy bajo	Normal o discretamente elevado	Normal o discretamente elevado
Trastornos de la beta oxidación de los ácidos grasos	Bajo	Normal o discretamente elevado	Elevado
Aciduria glutárica tipo 2	Bajo	Elevado	Elevado

Figura 10.2. Muestra de otro DD

En la figura 10.3 se muestran los metamodelos que forman parte de la solución propuesta, así como las transformaciones que se han definido sobre ellos. En la parte izquierda de la figura se muestra el metamodelo de DD (*Metamodelo DD*) independiente de conceptos de la computación. Esto es, este metamodelo sirve para representar todos los conceptos que abarcan los DD, sin hacer uso de terminología de la computación. También se ha definido un metamodelo que se utiliza para representar SID (*Metamodelo SID*) independiente de la plataforma, lo que quiere decir que sirve para representar un SID genérico, ya que no hace uso de conceptos de lenguajes de programación específicos. Entre ambos, se ha definido una transformación M2M entre modelos de DD y modelos de SID. También se han definido tres transformaciones M2T para generar SID ejecutables en *CLIPS/FuzzyCLIPS*, *JESS/FuzzyJESS* y el lenguaje de *scripting de Matlab*. **CLIPS** (Savely et al., 2017) y **FuzzyCLIPS** (NRCC, 2015) son estándares *de facto* para la construcción de sistemas basados en reglas, mientras que **JESS** (Friedman-Hill, 2017) y **FuzzyJESS** (Orchard, 2001) son lenguajes para la definición de sistemas basados en reglas que se ejecutan sobre Java y el lenguaje de **scripting de Matlab** es un lenguaje para el cálculo de matrices y vectores. Por último, también se han definido otros dos metamodelos para representar los datos de entrada (*Metamodelo datos entrada*) y los resultados de los SID (*Metamodelo resultados*). De la misma forma, también se han definido dos transformaciones M2T para obtener un formulario en lenguaje java que permita introducir datos de entra-

da a partir del *Metamodelo datos entrada* y para mostrar los resultados del SID en HTML a partir del *Metamodelo resultados*.

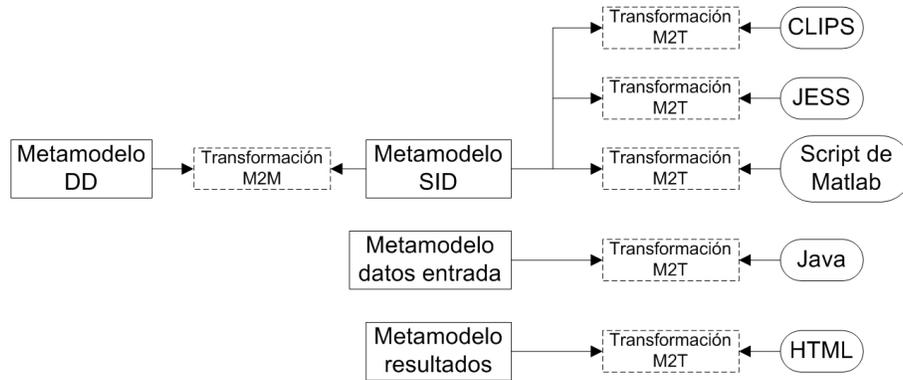


Figura 10.3. Metamodelos y transformaciones de la solución propuesta

A continuación se explican los metamodelos y las transformaciones.

### 10.1. Metamodelo de Diagnóstico Diferencial

En la figura 10.4 se muestra el metamodelo completo de DD que se ha definido y que sirve para especificar toda la información contenida en DD como el mostrado en la figura 10.2.

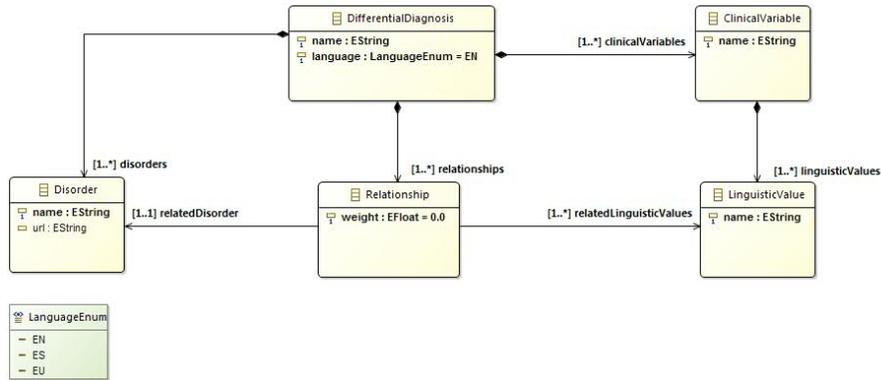


Figura 10.4. Metamodelo de DD

La metaclass principal se denomina *DifferentialDiagnosis* y está compuesta por las metaclasses *Disorder*, *ClinicalVariable* y *Relationship*. Estas metaclasses

ses se utilizan para especificar la información sobre los trastornos, las variables clínicas y las relaciones entre trastornos y variables del DD, respectivamente.

La metaclase *DifferentialDiagnosis* contiene dos atributos para guardar el nombre del DD, así como el idioma en que se ha expresado el mismo. Por el momento, se soportan tres idiomas: inglés (en), castellano (es) y euskara (eu) (ver el enumerador *LanguageEnum* en la figura 10.4). Para el DD mostrado en la figura 10.2 el nombre es *Diagnóstico de un trastorno*.

La metaclase *Disorder* guarda el nombre de un trastorno, así como la URL de una página web con información sobre el mismo. En el DD de la figura 10.2 se muestran tres conjuntos de trastornos, los valores de la primera columna: *Trastornos de la cetogénesis*, *Trastornos de la beta oxidación de los ácidos grasos* y *Aciduria glutárica tipo 2*.

Las variables clínicas del DD se definen mediante la metaclase *ClinicalVariable* y expresan todos los valores que pueden tener utilizando la metaclase *LinguisticValue*. En el DD de la figura 10.2 aparecen tres variables clínicas: *Gasometría*, *Amonio* e *Insulina / Glucemia*. La primera de ellas, por ejemplo, está formada por seis valores lingüísticos: *Muy bajo*, *Bajo*, *Discretamente bajo*, *Normal*, *Discretamente elevado* y *Elevado*.

Por último, la metaclase *Relationship* se utiliza para indicar la información de las celdas del DD. Cada relación une un trastorno (o conjunto de trastornos) con uno o más valores lingüísticos pertenecientes a una variable clínica (relaciones de asociación *relatedDisorder* y *relatedLinguisticValues* respectivamente en el metamodelo). Por ejemplo, en el DD de la figura 10.2 se muestra que la variable clínica *Gasometría* tiene una relación con el conjunto de trastornos *Trastornos de la cetogénesis* cuando la variable tiene un valor *Muy bajo*. De la misma forma, la variable *Amonio* tiene dos relaciones con ese mismo conjunto de trastornos, con los valores *Normal* y *Discretamente elevado*. Cada relación dispone de un peso para indicar cómo de importante es esa relación respecto al trastorno. Este peso se indica en el atributo *weight* y consiste en un valor en el rango  $[0, 1]$ . La suma de los pesos de todas las relaciones que afectan al mismo trastorno es la unidad. Suponiendo que las tres variables clínicas del DD mostrado en la figura 10.2 tienen el mismo peso para el conjunto de trastornos *Trastornos de la cetogénesis*, a cada una de las relaciones habrá que especificarle un peso de  $0,3333$ .

## 10.2. Metamodelo de Sistema de Inferencia Difusa

Los SID contemplados siguen la estructura de los *SID con módulos de reglas de entrada única* y los de tipo *Takagi-Sugeno de orden-cero*. En la figura 10.5 se muestra el metamodelo relativo a este tipo de SID que se explica a continuación.

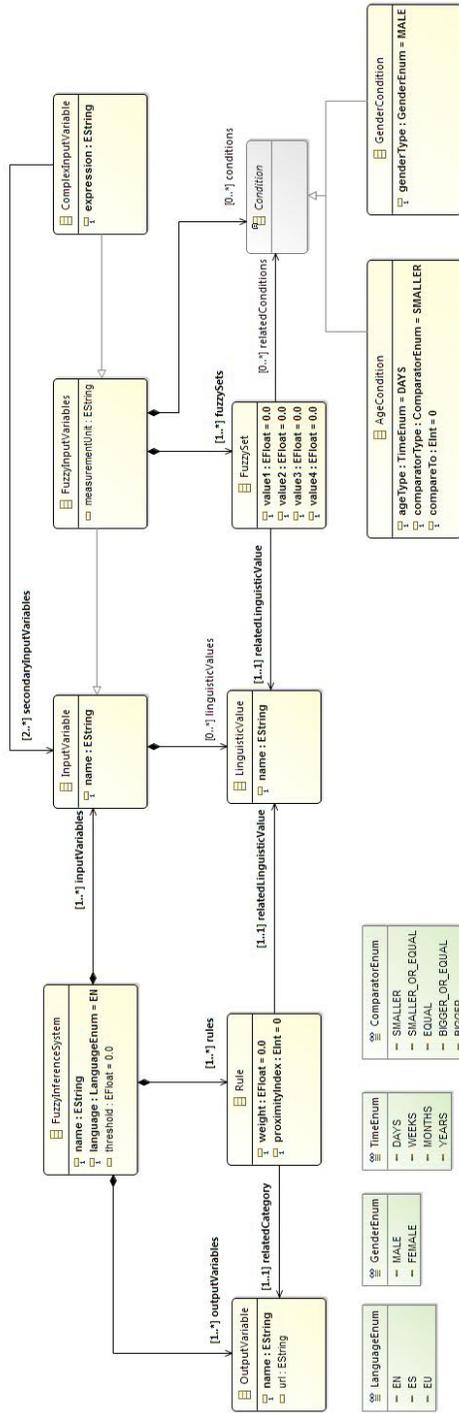


Figura 10.5. Metamodelo de SID

Un SID (*FuzzyInferenceSystem* en la figura 10.5) está compuesto por variables de salida (*OutputVariable*), variables de entrada (*InputVariable*) y una serie de reglas (*Rule*) que las relacionan. La metaclass *FuzzyInferenceSystem* se utiliza para especificar la información general del sistema, el nombre, el idioma o el umbral de probabilidad que deberá superar como mínimo una de las variables de salida para que no se considere un diagnóstico normal.

La metaclass *OutputVariable* se utiliza para describir las variables de salida, esto es, las categorías sobre las que el SID devolverá un resultado, o lo que es lo mismo, las variables de los consecuentes del SID. En el ámbito médico las variables de salida serán los trastornos y los conjuntos de trastornos. Cada variable de salida se caracteriza por un nombre y opcionalmente una URL que enlace a una página web que ofrezca más información sobre ella.

La metaclass *InputVariable*, por su lado, se utiliza para representar las variables sobre las que se introduce información en el SID, esto es, las variables de los antecedentes del SID. Dependiendo del uso que se vaya a hacer de las variables de entrada el metamodelo dispone de cuatro tipos de variables: *Simple*, *discreta*, *difusa* y *compleja*.

Las variables de entrada **simples** (*InputVariable*) sólo se utilizan como componentes de las variables de entrada complejas.

Las variables **discretas** (*InputVariable*) se utilizan para describir variables de entrada que se expresan mediante una escala de valores lingüísticos (*LinguisticValue*). Por ejemplo, una variable discreta llamada *Dolor muscular* puede estar compuesta por los valores lingüísticos {*Sin dolor*, *Soportable*, *Molestia*, *Moderado*, *Insoportable*}.

Las variables **difusas** (*FuzzyInputVariable*), por su lado, son variables de entrada compuestas por valores lingüísticos (*LinguisticValue*) descritos mediante conjuntos difusos trapezoidales (*FuzzySet*). La metaclass *FuzzySet* dispone de atributos para definir los cuatro valores numéricos necesarios para expresar una FP trapezoidal. Las variables difusas también disponen de un atributo para indicar la unidad de medida en que se expresa (atributo *measurementUnit*).

Por último, las variables **complejas** (*ComplexInputVariable*) se utilizan para representar variables de entrada que son combinación de una o más variables. Esto ocurre, por ejemplo, cuando en un DD se utiliza una proporción entre dos variables. En el DD de la figura 10.2, sin ir más lejos, se utiliza la proporción entre la *insulina* y la *glucemia*. Esta proporción en un SID se describe utilizando una variable compleja. A su vez, las variables *insulina* y *glucemia* podrán ser variables simples o difusas, ya que el metamodelo da libertad para ello en sus relaciones. El atributo *expression* indica la expresión aritmética que une las variables que forman la variable compleja en cuestión.

En el dominio médico es muy habitual que las variables de entrada difusas y complejas estén condicionadas, esto es, que los valores lingüísticos que las forman tengan asociadas distintas FP dependiendo de ciertos factores como la edad y el género. Por ejemplo, póngase la variable de entrada *Amonio* (ver tabla 10.1). Esta variable se mide en  $\mu\text{mol/l}$ , su universo de discurso está de-

finido en el rango [5, 7000] y está compuesta por cinco valores lingüísticos: *Normal*, *Discretamente elevado*, *Elevado*, *Muy elevado* y *Crítico*. A estos valores lingüísticos les corresponden FP trapezoidales diferentes dependiendo de si el paciente tiene 3 o menos meses de edad o si es mayor de 3 meses. Así, al valor lingüístico *Discretamente elevado* le corresponden los valores de FP {90, 110, 162, 198} cuando el paciente tiene 3 o menos meses de edad y los valores {45, 55, 135, 165} cuando es mayor que dicha edad.

<b>Amonio</b> ( $\mu\text{mol/l}$ )	<b><math>\leq 3</math> meses</b>	<b><math>&gt; 3</math> meses</b>
<b>Normal</b>	5, 5, 90, 110	5, 5, 45, 55
<b>Discretamente elevado</b>	90, 110, 162, 198	45, 55, 135, 165
<b>Elevado</b>	162, 198, 315, 385	135, 165, 315, 385
<b>Muy elevado</b>	315, 385, 1800, 2200	315, 385, 1800, 2200
<b>Crítico</b>	1800, 2200, 7000, 7000	1800, 2200, 7000, 7000

**Tabla 10.1.** PF trapezoidales correspondientes a los valores lingüísticos de la variable *Amonio* para pacientes menores y mayores de 3 meses (extraído y adaptado de (Couce et al., 2009))

La metaclass *GenderCondition* se utiliza para definir condiciones de género. El género viene dado por uno de los valores establecidos mediante el enumerador *GenderEnum*, *hombre* o *mujer*. La condición de edad, por su lado, se establece en la metaclass *AgeCondition* y requiere tres parámetros, una unidad de edad entre las ofrecidas en el enumerador *AgeEnum*, un comparador, seleccionado del enumerador *ComparatorEnum*, y un número para indicar la cantidad. De esta forma se pueden crear condiciones como  $\leq 3$  meses o  $> 4$  semanas.

Las reglas (*Rule*) se utilizan para relacionar variables de entrada con variables de salida. Cada regla guarda un peso (atributo *weight*) que representa la importancia de la variable de entrada para dicha variable de salida. La suma de los pesos de todas las reglas para una misma variable de salida es la unidad. También guarda el valor de la FP constante que forma el conjunto difuso de la variable de salida en el atributo *proximityIndex*.

### 10.3. Metamodelos de datos de entrada y resultados

El metamodelo que especifica la estructura de los datos de entrada se puede observar en la figura 10.6 y se utiliza para formalizar los datos de entrada del SID.

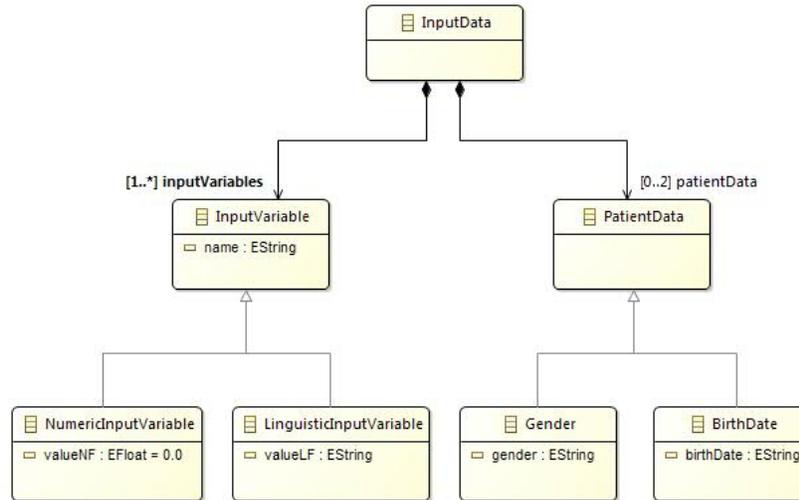


Figura 10.6. Metamodelo de los datos de entrada

Por un lado, se describen los valores de entrada que se corresponden con las variables de entrada del SID (*InputVariable*). Dependiendo del tipo de variable de entrada el valor será de tipo numérico (*NumericInputVariable*) o lingüístico (*LinguisticInputVariable*). Así, como a las variables simples y difusas del metamodelo de SID les corresponde un valor numérico, se utilizará la primera de las metaclases para especificar sus valores de entrada, mientras para indicar los valores de entrada de las variables discretas del metamodelo de SID se utilizará la segunda metaclase.

Por otro lado, este metamodelo describe los datos de entrada relativos a datos del paciente que son objetivos e invariables (*PatientData*), tales como la fecha de nacimiento (*BirthDate*) o el género (*Gender*). Estos datos se corresponden o se utilizan para resolver las condiciones establecidas en el SID.

En la figura 10.7, por su lado, se puede observar el metamodelo que especifica la estructura de los resultados ofrecidos por el SID.

Supóngase que los resultados obtenidos por un SID son los mostrados en la figura 10.8. Los resultados corresponden a un SID con 7 variables de entrada y 12 variables de salida.

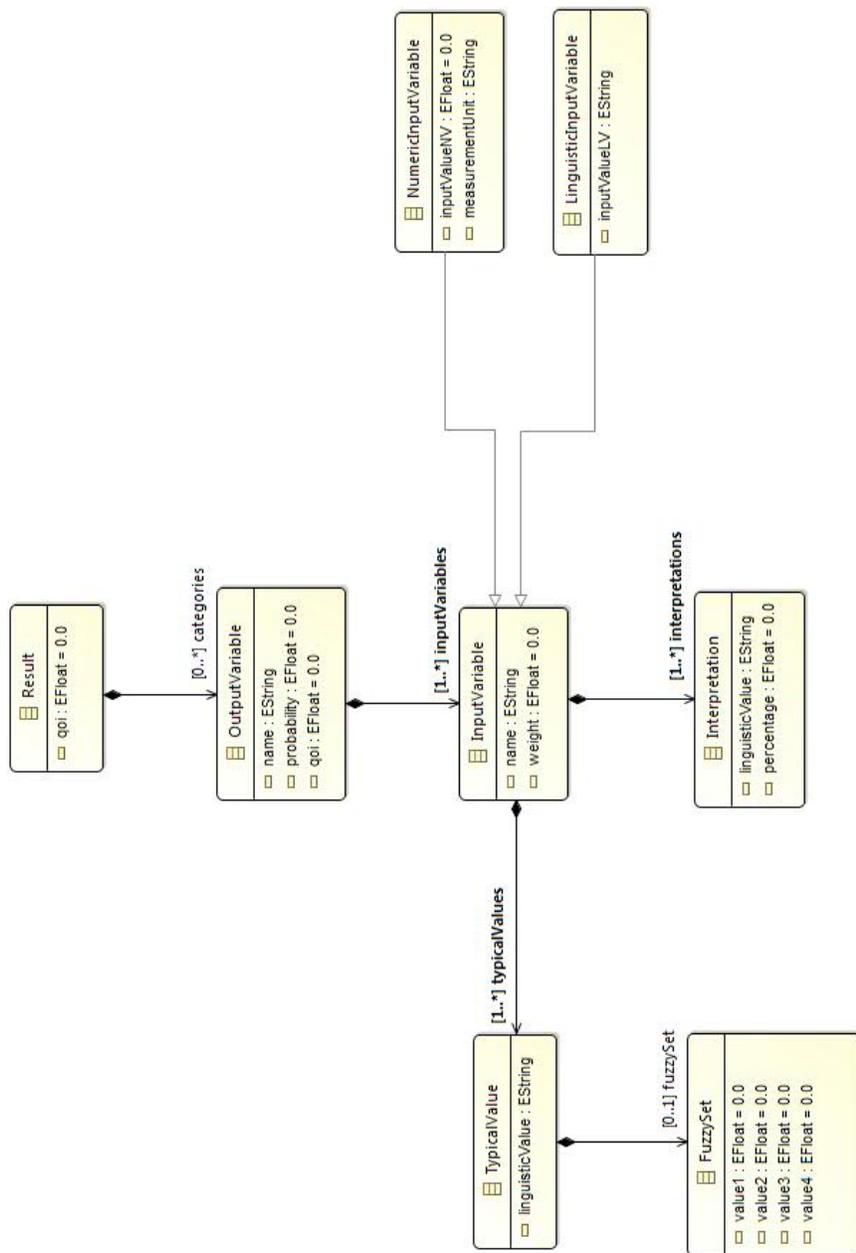


Figura 10.7. Metamodelo de los resultados

Esta recomendación tiene un coeficiente de calidad de información (Qoi, Quality of Information) del 85.71%.

#### (77.99%) Trastornos de la cetogénesis

- Índice de calidad de información (Qoi, Quality of Information) del 90.00%

Valores introducidos	Interpretación	Valores típicos de la patología	Pesos de los factores
- pH: 6.8	- <b>Muy bajo (50.0%)</b> - <b>Bajo (50.0%)</b>	- Muy bajo [5.5, 5.5, 6.7, 6.9]	0.2
- <b>Amonio</b> (mmol/L): 137.0	- <b>Discretamente elevado (93.33%)</b> - <b>Elevado (6.67%)</b>	- Normal [5.0, 5.0, 40.0, 60.0] - Discretamente elevado [40.0, 60.0, 135.0, 165.0]	0.3
- <b>Lactato</b> : Normal	- <b>Normal (100.0%)</b>	- Normal - Discretamente elevado	0.1
- <b>GOT-GPT</b> (UI/L): 200.0	- <b>Elevado (100.0%)</b>	- Elevado [70.0, 90.0, 2000.0, 2000.0]	0.1
- <b>CPK</b> (UI/L): -		- Elevado [350.0, 410.0, 20000.0, 20000.0]	0.1
- <b>Ácido úrico</b> (mg/dl): 3.0	- <b>Normal (100.0%)</b>	- Normal [0.0, 0.0, 6.0, 8.0]	0.1
- <b>3-Hidroxiacetil-CoA</b> (mmol/L): 0.25	- <b>Bajo (100.0%)</b>	- Bajo [0.0, 0.0, 0.25, 0.35]	0.1

#### (57.99%) Trastornos de la beta oxidación de los ácidos grasos

- Índice de calidad de información (Qoi, Quality of Information) del 90.00%

Valores introducidos	Interpretación	Valores típicos de la patología	Pesos de los factores
- pH: 6.8	- <b>Muy bajo (50.0%)</b> - <b>Bajo (50.0%)</b>	- Bajo [6.7, 6.9, 7.28, 7.32]	0.2
- <b>Amonio</b> (mmol/L): 137.0	- <b>Discretamente elevado (93.33%)</b> - <b>Elevado (6.67%)</b>	- Normal [5.0, 5.0, 40.0, 60.0] - Discretamente elevado [40.0, 60.0, 135.0, 165.0]	0.3

**Figura 10.8.** Fragmento de los resultados correspondientes al SID sobre la identificación de la etiología más probable causa de hiperamonemia

Todos los resultados se engloban dentro de la metaclase *Result*. En la primera línea de los resultados de la figura 10.8 se muestra el Índice de Calidad de la Información (ICI) de los mismos (Lima et al., 2010), el cual se representa mediante el atributo *qoi*. Este índice indica la fiabilidad de una recomendación dependiendo del número de valores de entrada. Así, cuantos más valores se tengan en cuenta, más fiable será la recomendación que da el sistema. Por el contrario, a valores muy bajos del ICI, la recomendación ofrecida no será muy fiable, en cuyo caso será aconsejable volver a realizar el cálculo cuando se tengan más valores de entrada. El ICI indicado en la metaclase *Result* tiene en cuenta el total de los valores de entrada del SID. Como el ICI de la recomendación se ha establecido en un 85.71% y teniendo en cuenta que el SID tiene 7 variables de entrada, se puede sobrentender que falta un valor de entrada.

Los resultados están formados por un conjunto de variables de salida (*OutputVariable*) para cada una de las cuales se especifica el nombre, la probabi-

lidad y su ICI particular. Las variables de salida están ordenadas en orden descendente de probabilidad. El ICI de las variables de salida sólo tiene en cuenta el número de valores de entrada de las variables de entrada que afectan a esta variable de salida, así como los pesos de las mismas. En los resultados de la figura 10.8 se muestran dos variables de salida (*Trastornos de la cetogénesis* y *Trastornos de la beta oxidación de los ácidos grasos*) con unas probabilidades del 77.99 % y 57.99 % respectivamente. Ambas variables han obtenido el mismo ICI (90.00 %), lo que quiere decir que el valor de entrada que falta en el SID se corresponde a una variable de entrada que tiene el mismo peso en ambas variables de salida.

Con la intención de poder ofrecer una justificación de los resultados, para cada una de las variables de salida se ofrece información de cada una de las variables de entrada que le afectan (*InputVariable*). Para cada una de las variables de entrada se indica el nombre (atributo *name* de la metaclass *InputVariable* del metamodelo) que se muestra en la primera columna de las tablas de la figura 10.8), el *valor de entrada* (primera columna), la *interpretación* (segunda columna), los *valores típicos* (tercera columna) y el *peso* (cuarta columna).

El **valor de entrada** indica el valor que se ha introducido en el SID para la variable de entrada, valor que puede ser numérico (metaclass *NumericInputVariable*) o lingüístico (metaclass *LinguisticInputVariable*) dependiendo del tipo de variable de entrada. En la figura 10.8 la variable de entrada *pH* ha tenido un valor de entrada de *6.8*, mientras que la variable de entrada *Lactato* ha tenido un valor *Normal*. Cuando no se introduce un valor de entrada para una variable, éste se representa mediante un guión (ver valor de la variable de entrada *CPK* en la figura 10.8).

Cada variable de entrada tiene una o más **interpretaciones** (metaclass *Interpretation*). Estas interpretaciones indican el valor o los valores lingüísticos de las variables de entrada que les corresponden a los valores de entrada. En el caso de las variables discretas, su interpretación será igual al valor de entrada, pero en el caso de las variables de entrada difusas y complejas su interpretación dependerá de la pertenencia de su valor de entrada en los distintos valores lingüísticos de la variable. El grado de pertenencia se representa mediante el atributo *percentage*. Por ejemplo, la variable de entrada *pH* tiene una pertenencia del 50 % en los valores lingüísticos *Muy bajo* y *Bajo* tal y como se puede observar en la segunda columna de la figura 10.8. Por su lado, la variable de entrada *Lactato* es discreta y por lo tanto, en la segunda columna se muestra el mismo valor que el de entrada, *Normal*, con una pertenencia del 100 %. Cuando la interpretación de un valor de entrada coincide con alguno de los valores típicos de la variable de entrada se muestra en color verde y en negrita. Sin embargo, cuando no coincide se muestra en color rojo y grosor normal. En el ejemplo anterior, en la variable *pH* la interpretación *Muy bajo* se muestra en verde, mientras que la interpretación *Bajo* se muestra en rojo.

Los **valores típicos** representan los valores que suele tener la variable de entrada respecto a la variable de salida a la que afecta y están formados por

valores lingüísticos (metaclase *TypicalValue*). Para los casos de las variables de entrada difusas y complejas, los valores lingüísticos se describen mediante los cuatro valores numéricos necesarios para representar la FP trapezoidal correspondiente que siguen (metaclase *FuzzySet*). En la tabla de la figura 10.8 se puede observar que la variable de entrada discreta *Lactato* tiene dos valores típicos, *Normal* y *Discretamente elevado*. La variable *pH*, al ser una variable difusa, aparte del valor lingüístico típico *Muy bajo*, también indica los cuatro valores que describen el conjunto difuso de ese valor lingüístico,  $\{5.5, 5.5, 6.7, 6.9\}$ .

Finalmente, el **peso** (atributo *weight*) representa la importancia de la variable de entrada respecto a la variable de salida a la que afecta. Por ejemplo, en la figura 10.8 el peso de la variable de entrada *Amonio* respecto a la variable de salida *Trastornos de la cetogénesis* es *0.1*.

#### 10.4. Transformaciones de modelo a modelo

Se ha considerado que los SID son un medio adecuado para representar DD, ya que la información dispuesta en tablas es directamente transformable a reglas. Por ejemplo, tómesese el DD mostrado en la figura 10.2. En este caso concreto, la regla correspondiente a la información que muestra la primera fila (relativa a la variable de salida *Trastornos de la cetogénesis*) es la siguiente: “**IF** Gasometría **IS** Muy bajo **AND** Amonio **IS** Normal o Discretamente elevado **AND** Insulina/Glucemia **IS** Normal o Discretamente elevado **THEN** Trastornos de la cetogénesis”.

Las reglas de transformación M2M entre los metamodelos de DD y SID se han creado en base a correspondencias entre ambos metamodelos. Las transformaciones M2M se han definido en lenguaje ATL. En la tabla 10.2 se resumen esas correspondencias, las cuales se explican en los siguientes párrafos.

Metamodelo de origen (DD)	Metamodelo de destino (SID)
DifferentialDiagnosis	FuzzyInferenceSystem
Disorder	Output Variable
ClinicalVariable	Input Variable
LinguisticValue	LinguisticValue
Relationship	Rule

**Tabla 10.2.** Correspondencias entre los metamodelos de DD y SID

En un DD (metaclase *DifferentialDiagnosis*) que se convierte a SID (metaclase *FuzzyInferenceSystem*) cada trastorno del DD (*Disorder*) se transforma

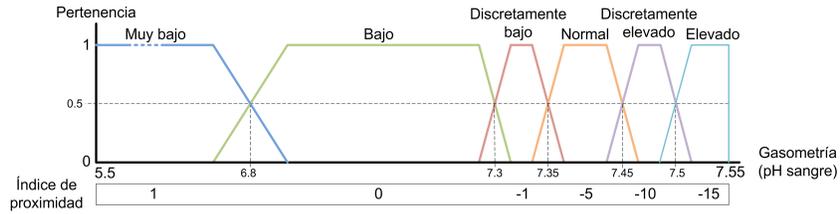
en una variable de salida del SID (*OutputVariable*) y cada variable clínica del DD (*ClinicalVariable*) se transforma en una variable de entrada del SID *InputVariable*.

De la misma forma, los valores lingüísticos de las variables clínicas (*LinguisticValue*) del DD se convierten en valores lingüísticos de las variables de entrada del SID (también llamadas *LinguisticValue*). En este punto, todas las variables de entrada del SID se consideran discretas, ya que sólo se tiene de ellas la información sobre los valores lingüísticos.

Al contrario de los casos anteriores, donde las equivalencias han sido de uno-a-uno, a cada relación en el DD (metaclase *Relationship*) le corresponde un número de reglas en el SID (*Rule*) igual al número de valores lingüísticos de la variable clínica. Cada una de las reglas tendrá un valor constante de FP diferente representado en el atributo índice de proximidad (*proximity index*). El índice de proximidad se establece automáticamente a partir de la distancia entre el valor lingüístico típico que une una variable clínica y un trastorno (o conjunto de trastornos) y el resto de los valores lingüísticos de esa variable clínica. Por ejemplo, en el DD de la figura 10.2 se puede ver que la variable clínica *Gasometría* tiene una relación con el conjunto de trastornos *Trastornos de la cetogénesis* con un valor lingüístico *Muy bajo*. Esto indica que usualmente los pacientes aquejados por este conjunto de trastornos tienen la *Gasometría* en el valor indicado. Por lo tanto, cuando un paciente tiene la *Gasometría* en ese valor, la probabilidad de que el paciente tenga *Trastornos de la cetogénesis* es mayor. Por el contrario, cuanto más se alejen los valores lingüísticos de ese valor típico menos probable será que el paciente tenga esa enfermedad. Esto es, partiendo de que la variable clínica tiene seis valores lingüísticos (*Muy bajo*, *Bajo*, *Discretamente bajo*, *Normal*, *Discretamente elevado* y *Elevado*), si el paciente tiene la *Gasometría* en nivel *Discretamente elevado* (que difiere bastante del valor típico), la probabilidad de tener *Trastornos de la cetogénesis* será baja.

En la figura 10.9 se muestran los índices de proximidad para todos los valores lingüísticos de la variable clínica *Gasometría* en referencia al conjunto de trastornos *Trastornos de la cetogénesis*. De esta forma, mientras que a la regla que une el valor lingüístico (*Muy bajo*) de la variable de entrada *Gasometría* se le asigna un índice de proximidad de *1*, a las reglas correspondientes a los siguientes valores lingüísticos de esa variable de entrada se les asigna índices más bajos. Así, a la regla correspondiente al valor lingüístico *Bajo* se le indica un índice de *0* (lo que equivale a no tener influencia en la variable de salida), a la regla del valor *Discretamente bajo* se le asigna un índice de *-1*, etc. La regla correspondiente al último de los valores lingüísticos (*Alto*) se le asigna un índice de *-15*, por lo que la activación de esta regla tendrá una influencia muy negativa en la variable de salida *Trastornos de la cetogénesis*.

Con todo lo dicho, a partir de la relación que une el conjunto de trastornos *Trastornos de la cetogénesis* con la variable clínica *Gasometría* y su valor lingüístico *Muy bajo* en el metamodelo de DD, se construirán las reglas de la



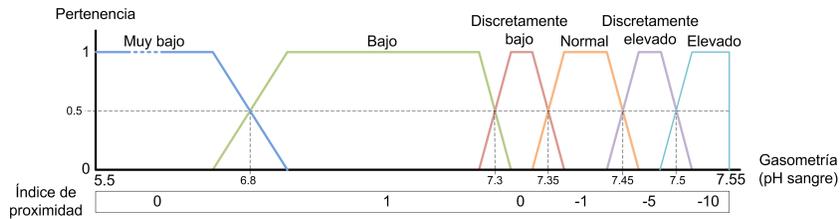
**Figura 10.9.** Índices de proximidad para los distintos valores lingüísticos de la variable clínica *Gasometría* en relación al conjunto de trastornos *Trastornos de la cetogénesis*

figura 10.10. Como la segunda regla tiene un valor de  $\theta$  en el consecuente, no tendrá ningún efecto en el cálculo, y por lo tanto, será suprimible.



**Figura 10.10.** Conjunto de reglas que describen la relación entre la variable de entrada *Gasometría* y la variable de salida *Trastornos de la cetogénesis*

También hay que tener en cuenta que el índice de proximidad varía dependiendo del trastorno. Así, los índices de proximidad de los valores lingüísticos de la variable clínica *Gasometría* para el trastorno *Aciduria glutárica tipo 2* son los que se ven en la figura 10.11. El valor típico de esta variable clínica para ese trastorno es *Bajo*. Por lo tanto, ese valor obtiene un índice de 1, mientras que los índices de los valores que se alejen de ese valor típico serán cada vez menores.



**Figura 10.11.** Índices de proximidad para los distintos valores lingüísticos de la variable clínica *Gasometría* en relación al trastorno *Aciduria glutárica tipo 2*

## 10.5. Transformaciones de modelo a texto

Las transformaciones M2T definidas para poder obtener SID ejecutables a partir del metamodelo de SID se han realizado utilizando plantillas en lenguaje Xpand. En concreto se han definido tres plantillas, para el lenguaje *CLIPS/FuzzyCLIPS*, para el lenguaje *JESS/FuzzyJESS* y para el lenguaje de *scripting de Matlab*. Esto es así porque cada uno de los lenguajes incluye características y estructuras propias. En el extracto 10.1 se muestra el fragmento de una plantilla relativa al lenguaje *CLIPS/FuzzyCLIPS* para la generación de código de una variable difusa y en el extracto 10.2 se muestra el resultado de la generación del extracto anterior.

```

1: «DEFINE createDeftemplates FOR fuzzyInferenceSystem
   ::FuzzyFactor -»
2: (deftemplate «getFactorCodeFromName(name)»
3: «getFactorMinValue(null)» «getFactorMaxValue(null)»
   «measurementUnit»
4: «FOREACH linguisticValues AS lv -»
5:   («getLinguisticValueCodeFromName(lv.name) -»
     («fuzzySets.get(0).value1» 0)
     («fuzzySets.get(0).value2» 1)
     («fuzzySets.get(0).value3» 1)
     («fuzzySets.get(0).value4» 0))
6: «ENDFOREACH -»
7: ))
8: «ENDDEFINE»

```

**Extracto 10.1.** Extracto de plantilla Xpand para generar SID en lenguaje *CLIPS/FuzzyCLIPS*

También se ha definido una plantilla para crear un formulario en lenguaje Java que permita introducir los datos en una aplicación de ejecución, así como una plantilla para poder representar los resultados en una página HTML. Sobre esta aplicación de ejecución se habla en profundidad en el capítulo 11.

```

(deftemplate factor2
0.0 80.0 mmol/L
((linguistic3 (0.0 0)(0.0 1)(2.3 1)(2.7 0))
(linguistic4 (2.3 0)(2.7 1)(2.8 1)(3.2 0))
(linguistic5 (2.8 0)(3.2 1)(80.0 1)(80.0 0))
))

```

**Extracto 10.2.** Extracto del resultado de la generación automática utilizando el extracto de la plantilla 10.1



## Herramientas para la definición de DD y SID

En este capítulo se presentan *Aide Differential Diagnosis Editor (Aide-DDE)*, un editor de modelos de DD para expertos médicos y *Aide Fuzzy Inference System Framework (Aide-FISF)*, un editor de modelos de SID y generador de SID ejecutables diseñado para ingenieros del conocimiento. Ambas herramientas se integran en Aide-GTP.

### 11.1. Aide Differential Diagnosis Editor

Aide-DDE es un editor sencillo que permite crear modelos de DD. Su objetivo principal es trasladar la información de un DD de forma fidedigna sin que el usuario deba introducir conceptos no médicos. Aide-DDE se puede integrar en el editor Aide-MAT de la plataforma Aide-DS.

En la figura 11.1 se muestra la interfaz principal de Aide-DDE. El idioma de la interfaz se establece cuando se abre el editor o utilizando la opción correspondiente en el menú *Opciones*. La interfaz se divide en cuatro secciones. En la sección superior se introduce el nombre del DD, en el caso de la interfaz mostrada el nombre del DD que se está creando es *Diagnóstico diferencial de sospecha*. En la sección izquierda se añaden los trastornos y los conjuntos de trastornos, que en la interfaz de la imagen son tres: *Trastornos de la cetogénesis*, *Trastornos de la betaoxidación de los ácidos grasos* y *Aciduria glutárica tipo 2*. De la misma forma, en la sección derecha se introducen las variables clínicas, las cuales también son tres en el DD de la figura 11.1: *Gasometría (pH en sangre)*, *Amonio* e *Insulina / Glucemia*. Por último, en la sección inferior se reflejan las relaciones entre las variables clínicas y los trastornos (o conjuntos de trastornos). Por ejemplo, para el conjunto de trastornos *Trastornos de la betaoxidación de los ácidos grasos* y la variable clínica *Amonio* el valor de la relación es *Normal - Discretamente elevado*.

Los trastornos y los conjuntos de trastornos se introducen mediante el editor de la figura 11.2, añadiendo el nombre y, opcionalmente, una URL con el

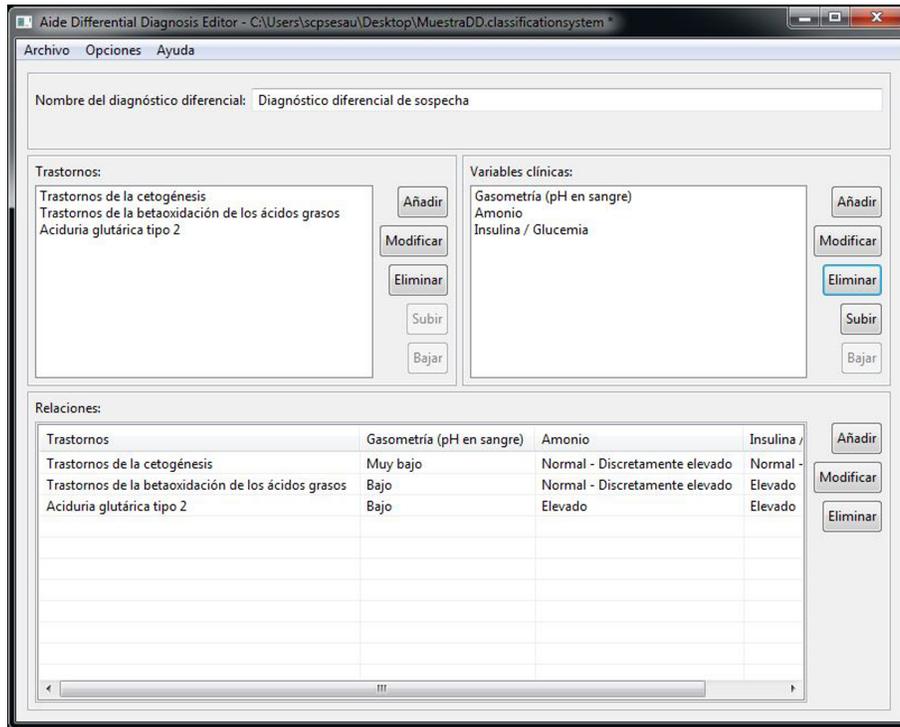


Figura 11.1. Interfaz principal de Aide-DDE

vínculo a una página web con información sobre el mismo. Una vez introducido, su nombre queda reflejado en la sección izquierda de la interfaz principal de Aide-DDE. Según se van introduciendo nuevos trastornos, automáticamente se crean nuevas filas en la tabla de relaciones de la sección inferior de la interfaz.

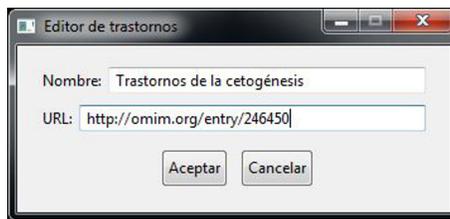
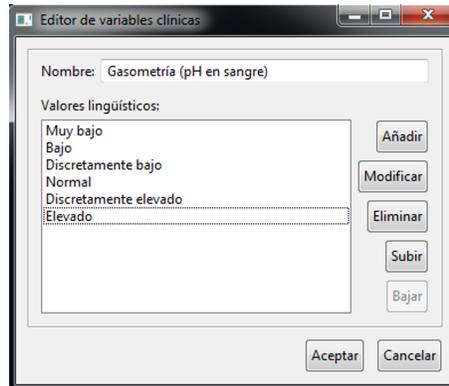


Figura 11.2. Interfaz del editor de trastornos

Las variables clínicas se introducen utilizando el editor de la figura 11.3. Mediante este editor se crean las variables, así como los valores lingüísticos

que las forman. En la figura se puede observar que para la variable clínica *Gasometría (pH en sangre)* se han definido seis valores lingüísticos diferentes. Todas las variables definidas utilizando Aide-DDE son discretas. Según se van introduciendo variables clínicas sus nombres quedan reflejados en la sección derecha de la interfaz principal. De la misma forma, también se crean columnas en la tabla de relaciones de la sección inferior.



**Figura 11.3.** Interfaz del editor de variables clínicas

La definición de las relaciones se lleva a cabo mediante el editor mostrado en la figura 11.4. Una relación se crea seleccionando de una lista el trastorno (o el conjunto de trastornos), la variable clínica y los valores lingüísticos de esta última que la unan al trastorno. En el editor de la figura se está creando una relación entre la variables clínica *Gasometría (pH en sangre)* y el conjunto de trastornos *Trastornos de la cetogénesis*. Esta variable sólo se relaciona con ese conjunto de trastornos mediante el valor lingüístico *Muy bajo*. Al crear una relación también hay que introducir el peso que tiene la relación respecto al trastorno. Dado que es un muy difícil que un usuario pueda introducir números en el rango  $[0, 1]$  y que se cumpla la cualidad de que la suma de los pesos de todas las relaciones para un mismo trastorno sea la unidad, Aide-DDE permite introducir los pesos de forma relativa al resto de relaciones del mismo trastorno especificando números en el rango  $[1, 100]$ . De esta forma, si una relación tiene un peso de 4 y otra tiene un peso de 1, la primera será cuatro veces más importante. Esto mismo se consigue si se especifica un peso de 24 en la primera y un peso de 6 en la segunda. Aide-DDE se encarga de calcular las proporciones en base a los datos introducidos y de asignar a las relaciones valores en el rango  $[0, 1]$ . A la relación en proceso de edición de la figura 11.4 se le ha asignado un peso de 1. Una vez introducida la relación, ésta queda reflejada en la tabla de la sección inferior de la interfaz principal. En la tabla de la figura 11.1 se puede ver el resultado de la adición de esta relación en la primera fila y la segunda columna.

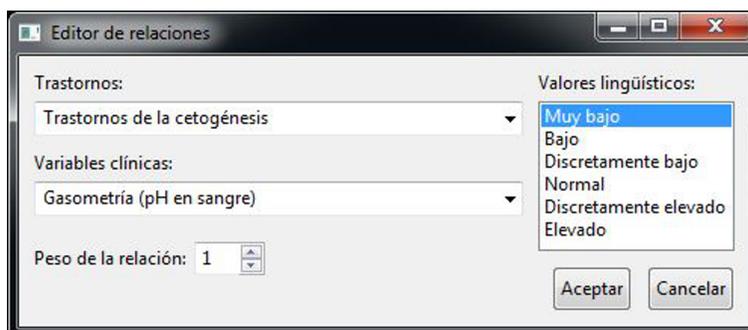


Figura 11.4. Interfaz del editor de relaciones

Los modelos resultantes de DD creados en Aide-DDE se guardan en lenguaje XMI y se pueden editar tanto con este editor como con el *framework* Aide-FISF.

## 11.2. Aide Fuzzy Inference System Framework

Aide-FISF es un *framework* para la edición de SID. Este *framework* permite definir modelos de SID desde cero y abrir modelos de DD que sean conformes al metamodelo de DD. En este último caso, cuando se abre el modelo de DD, éste se transforma automáticamente en un modelo de SID siguiendo las transformaciones M2M explicadas en el capítulo 10. Aparte de definir SID, Aide-FISF también permite aplicar técnicas de aprendizaje automático sobre los modelos de SID, ejecutarlos y generar SID ejecutables. Aide-FISF se puede integrar en el editor de GCI Aide-KAT de la plataforma Aide-DS.

A continuación, primero se explica la interfaz de la herramienta, para posteriormente explicar las opciones principales del menú opciones, que entre otras características ofrece funcionalidades para el entrenamiento y la simulación de los SID modelados.

### 11.2.1. Interfaz de Aide-FISF

En la figura 11.5 se muestra la interfaz principal de Aide-FISF. La interfaz se divide en cuatro secciones. Así, en la sección superior se muestra el nombre del SID (*Diagnóstico diferencial de sospecha* en la figura) y el umbral de los resultados. En el caso del SID de la figura, el umbral se ha establecido en  $0.0$ , lo que quiere decir que se mostrarán los resultados de todas las variables de salida que tengan una probabilidad mayor de 0. En la sección izquierda se muestran las variables de salida: *Trastornos de la cetogénesis*, *Trastornos de la betaoxidación de los ácidos grasos* y *Aciduria glutárica tipo 2*. En la sección derecha se muestran las variables de entrada, cinco en este caso: *Gasometría (pH en sangre)*, *Amonio*, *Insulina / Glucemia*, *Insulina* y *Glucemia*.

Por último, en la sección inferior se muestran las relaciones entre las variables de entrada y salida. No se muestran las reglas del SID porque éstas se crean automáticamente y las relaciones ofrecen la misma información de forma más compacta. En esta sección se puede ver que la variable de salida *Aciduria glutárica tipo 2* tiene una relación con la variable de entrada *Amonio* con un valor lingüístico *Elevado*.

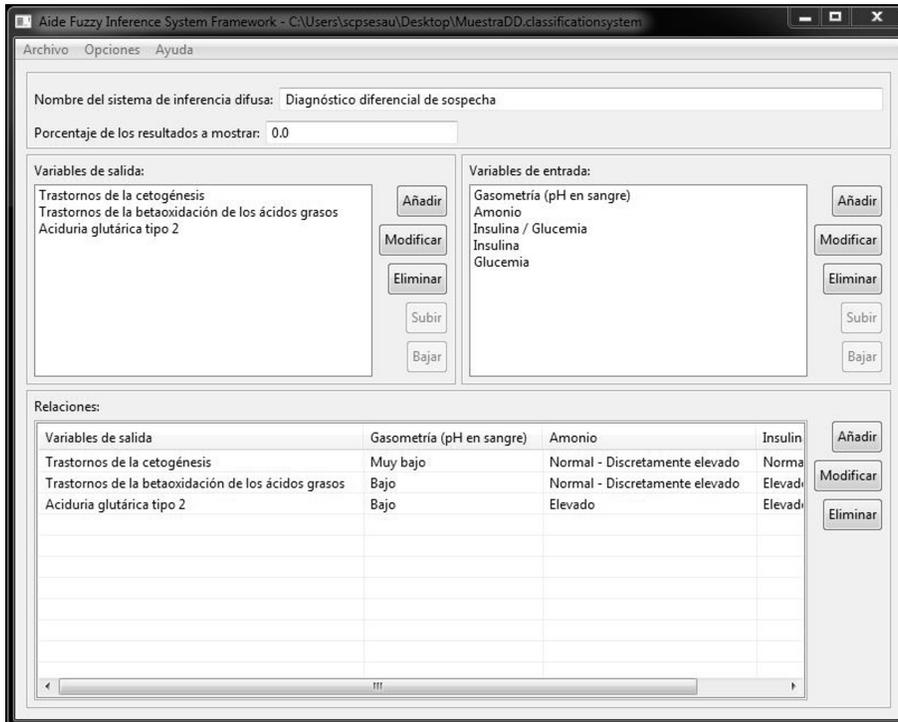


Figura 11.5. Interfaz principal de Aide-FISF

La definición y modificación de las variables de salida se realiza utilizando un editor muy similar al que se muestra en la figura 11.2. Según se van introduciendo variables de salida, se crean nuevas filas en la tabla de la sección de relaciones.

En cuanto a las variables de entrada, Aide-FISF contempla los cuatro tipos de variables de entrada del metamodelo de SID: simples, discretas, difusas y complejas. A la hora de introducir una variables de entrada hay que elegir el tipo de la misma y dependiendo de éste el editor que se abre para su edición es diferente. De las variables de entrada mostradas en la sección derecha de la figura 11.5, la variable *Gasometría (pH en sangre)* es discreta, la variable

*Amonio* es difusa, la variable *Insulina / Glucemia* es compleja y las variables *Insulina* y *Glucemia* son simples.

Las variables de entrada simples únicamente se definen mediante su nombre y una vez añadidas, aunque se muestran en la sección derecha de la interfaz, no se muestran en la tabla de relaciones de la sección inferior.

Las variables discretas se definen utilizando el editor mostrado en la figura 11.6 donde, además de su nombre, hay que introducir el conjunto de valores lingüísticos que lo forman. Cuando se crea una variable de entrada discreta, ésta se muestra en la sección derecha de la interfaz y se crea una nueva columna en la tabla de relaciones de la sección inferior.

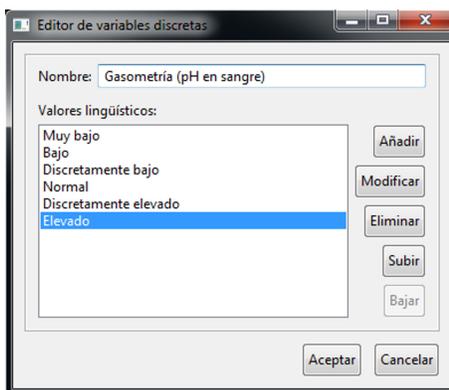


Figura 11.6. Interfaz del editor de variables discretas

Para crear variables difusas se utiliza el editor de la figura 11.7. Este editor, además de permitir definir los conjuntos difusos correspondientes a los valores lingüísticos (en la sección izquierda), permite añadir condiciones en la sección derecha del editor. En el caso de la variable de la figura (*Amonio*), en la figura se puede ver que tiene cinco valores lingüísticos (*Normal*, *Discretamente elevado*, *Elevado*, *Muy elevado* y *Crítico*) y dos condiciones (*Edad*  $\leq$  3.0 *Meses* y *Edad*  $>$  3.0 *Meses*). Para definir las condiciones de edad se utiliza un editor en el que se introducen el símbolo comparador ( $\leq$  y  $>$ ), el valor a comparar (3.0) y la unidad de tiempo (*Meses*). En cuanto a los valores lingüísticos, éstos se añaden utilizando el editor de la figura 11.8. Además del nombre del valor lingüístico (parte trasera de la figura) hay que añadir el conjunto difuso correspondiente a ese valor introduciendo los cuatro valores numéricos necesarios (parte delantera de la figura). En este caso concreto, al ser el *Amonio* una variable condicionada, hay que añadir dos conjuntos difusos para cada valor lingüístico, uno por cada condición.

El botón *Previsualizar* del editor de variables difusas permite visualizar gráficamente todos los conjuntos difusos que forman la variable de entrada (ver figura 11.9). En el caso de que la variable esté condicionada, se muestra

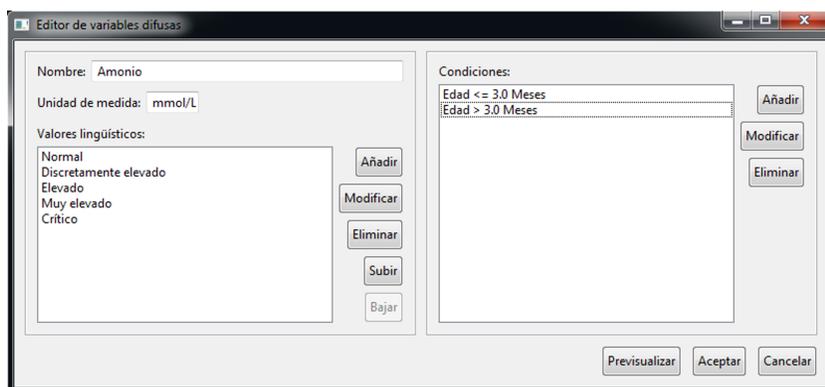


Figura 11.7. Interfaz del editor de variables difusas

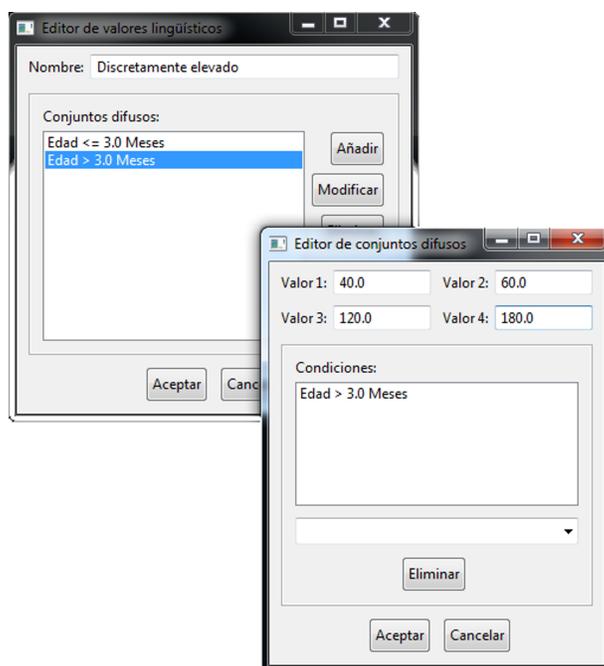


Figura 11.8. Interfaz del editor de conjuntos difusos

un gráfico por cada una de las alternativas. Cabe mencionar que las variables definidas mediante el editor deben ser variables *completas*. En cualquier otro caso el editor mostrará un mensaje de error.

Por último, para introducir variables complejas se utiliza el editor de la figura 11.10. Este editor, además de permitir la definición de valores lingüísticos (junto con los conjuntos difusos correspondientes) y condiciones, permite



Figura 11.9. Previsualización de la variable de entrada *Amonio*

introducir la expresión aritmética que une las variables simples y/o difusas que forman la variable compleja en la parte inferior del editor. En el caso de la variable *Insulina / Glucemia*, ésta está formada por las dos variables simples mencionadas anteriormente (*Insulina* y *Glucemia*). En este caso, las variables simples están unidas mediante el símbolo “/”, indicando de esta forma que se debe hacer una división entre los valores de entrada de las variables simples para obtener el valor de entrada de la variable compleja.

Finalmente, para la definición de las relaciones entre las variables de entrada y salida se utiliza un editor muy parecido al mostrado en la figura 11.4.

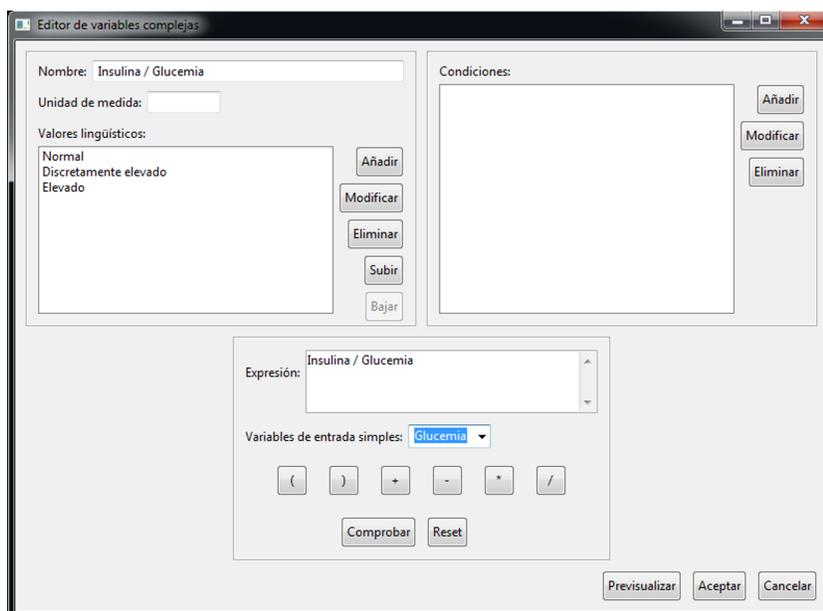
Con todo lo realizado hasta el momento, el resultado de la edición es un modelo de SID en lenguaje XMI con el cual se puede seguir trabajando. Por ejemplo, se pueden generar SID ejecutables a partir de él, se le pueden aplicar técnicas de aprendizaje automático o pueden realizarse simulaciones para ver su funcionamiento.

### 11.2.2. Funcionalidades del menú *opciones*

Este menú ofrece las funcionalidades principales del *framework* Aide-FISF, entre las que destacan el *cambio de tipo de variable de entrada*, la *generación automática de SID ejecutables*, el *entrenamiento de modelos de SID* y la *ejecución de modelos de SID*.

#### 11.2.2.1. Cambio de tipo de variables de entrada

Como ya se ha comentado, todas las variables de entrada creadas mediante el editor Aide-DDE son discretas, mientras que el *framework* Aide-FISF



**Figura 11.10.** Interfaz del editor de variables complejas

utiliza cuatro tipos de variables de entrada. Por lo tanto, cuando se abre con el *framework* un modelo de DD creado anteriormente usando Aide-DDE es probable que haya que modificar las variables para que se adecúen al modelo de SID. Mediante esta funcionalidad es posible realizar esta acción.

Cuando hay que realizar un cambio a una variable de entrada más compleja (por ejemplo, de una variable simple a una variable difusa o de una variable difusa a una compleja) automáticamente se abre el editor correspondiente para poder rellenar la información que falta en la nueva variable.

Si por el contrario, el cambio es hacia una variable más simple (por ejemplo, de variable difusa a discreta), se avisa al usuario de una posible pérdida de información y si se sigue adelante la información extra de la variable más compleja (la información sobre los conjuntos difusos y las condiciones en el ejemplo expuesto) se elimina.

#### 11.2.2.2. Generación automática de Sistemas de Inferencia Difusa ejecutables

Esta opción permite generar automáticamente a partir del modelo de SID un SID ejecutable en uno de los tres lenguajes disponibles: CLIPS/Fuzzy-CLIPS, JESS/FuzzyJESS y script de Matlab. Para llevar a cabo esta tarea es suficiente con seleccionar el directorio de destino donde se quiere guardar el fichero de SID ejecutable (ver figura 11.11).

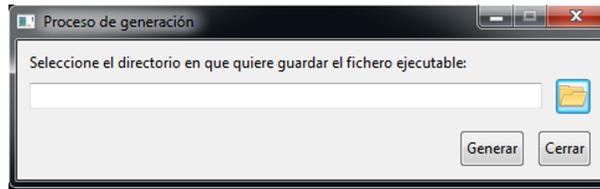


Figura 11.11. Funcionalidad de generación automática de Aide-FISF

### 11.2.2.3. Entrenamiento de modelos de Sistemas de Inferencia Difusa

Esta funcionalidad posibilita la mejora de las aptitudes de los modelos de SID diseñados mediante un sistema de aprendizaje automático. Concretamente, este sistema de aprendizaje automático es un AG de números reales que afina los valores de los conjuntos difusos de las variables de entrada difusas y complejas.

Ya que el proceso de entrenamiento trata de afinar los valores de los conjuntos difusos, los individuos de la población están formados por cadenas de números reales en lugar de valores binarios. En concreto, cada individuo está formado por todos los valores numéricos de los conjuntos difusos de todas las variables de entrada del SID. Así, por ejemplo, para un SID con 4 variables de entrada y 3 conjuntos difusos trapezoidales (4 valores) para cada una, los individuos tendrán una longitud de 48 números reales.

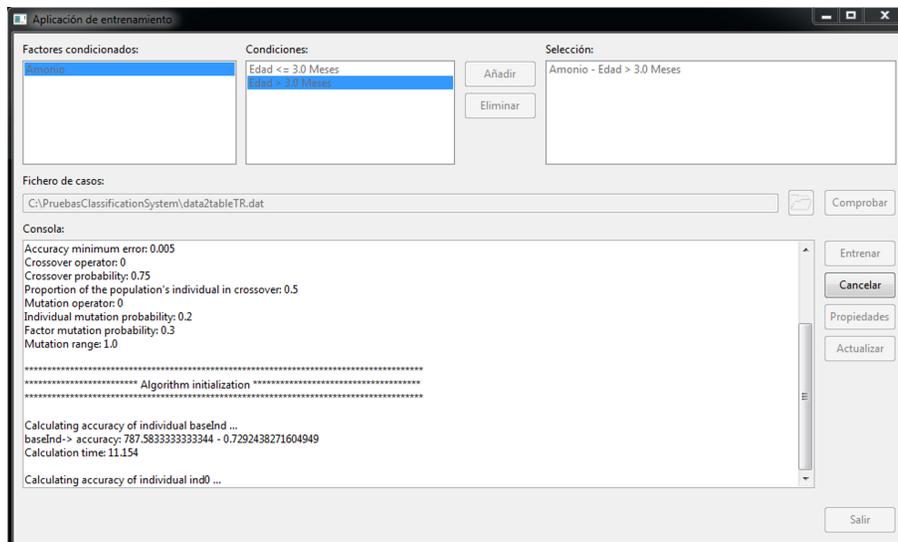


Figura 11.12. Funcionalidad de entrenamiento de Aide-FISF

En el caso de las variables de entrada condicionadas sólo se puede afinar una distribución de conjuntos difusos en cada entrenamiento. Por ello, antes de comenzar el entrenamiento el usuario debe elegir la distribución de conjuntos difusos que quiere afinar. En la lista superior izquierda de la aplicación de entrenamiento se muestran todas las variables clínicas que están condicionadas (*Amonio* en la figura 11.12). Cuando se selecciona una de ellas en la lista inmediatamente a la derecha de la anterior se muestran las distintas condiciones definidas para dicha variable. El usuario añade la condición correspondiente a los conjuntos difusos que quiera entrenar. En la lista superior derecha quedan reflejadas las condiciones seleccionadas para todas las variables clínicas condicionadas. En el caso de que el modelo que se quiera entrenar no disponga de ninguna variable clínica condicionada, entonces todas estas listas superiores en la figura 11.12 se mantendrán vacías.

A continuación hay que especificar un fichero de casos con el que entrenar el modelo. Los ficheros de casos soportados tienen un caso por línea y los valores de cada línea están separados por comas (*comma-separated-values, CSV*). Los primeros valores se corresponden con los valores de entrada de cada una de las variables de entrada y los siguientes con los valores correspondientes a las variables de salida. Éstos últimos tienen un valor de 0 si no se corresponden con los valores de entrada del caso y 1 si se corresponden. En el extracto 11.1 se puede ver un fragmento de un fichero de casos con 4 variables de entrada y 3 variables de salida. El valor correspondiente a la primera variable de entrada es *6.83*, el segundo *17.82*, el tercero *0.50* y el último valor es *472.60*. Seguido van los valores correspondientes a las tres variables de salida, los cuales son *1*, *0* y *0* respectivamente. Esto significa que el primer caso se corresponde con la primera variable de salida.

```
6.83,17.82,0.50,472.60,1,0,0
5.82,59.75,1.33,588.44,1,0,0
6.34,160.18,0.30,274.51,1,0,0
6.18,192.24,1.43,251.73,1,0,0
7.48,3075.6,2.64,147.46,0,1,0
7.44,3290.8,1.12,311.48,0,1,0
7.45,1965.6,2.38,497.01,0,1,0
7.33,6073.8,0.28,325.48,0,1,0
6.16,4538.8,14.88,30.60,0,0,1
6.58,2974.7,15.59,12.28,0,0,1
5.68,5407.4,78.53,51.64,0,0,1
5.93,2629.7,38.43,48.76,0,0,1
```

**Extracto 11.1.** Fragmento de un fichero de casos

Una vez indicada la ubicación del fichero de casos se comprueban las siguientes características:

- Los valores de los casos deben estar separados por comas.
- Los valores de los casos son números.

- La suma del número de elementos de un caso debe ser igual a la suma del número de variables de salida y variables de entrada simples y difusas del SID.
- Los valores relativos a las variables de salida sólo pueden ser 0 y 1, y sólo puede haber un valor de 1 para todas ellas.
- Los valores de entrada de los casos están dentro del rango definido para las variables de entrada del SID. Esto también se evalúa para las variables complejas, para lo cual primero se infiere su valor de entrada a partir de los valores de entrada de las variables que la forman.

Se indica al usuario si el fichero de casos es correcto o no, y si no lo es se ofrece al usuario información para localizar el error.

El AG que se utiliza para el entrenamiento sigue una *aproximación elitista* y dispone de cuatro condiciones de finalización: llegada a un número máximo de generaciones, consecución del individuo óptimo (con un margen de error), no mejoría de la aptitud de la población durante tres generaciones y no mejoría del mejor individuo de la población durante tres generaciones. En cuanto a los métodos de selección se utiliza el *muestreo estocástico universal*, junto con una *selección por truncamiento*.

Aide-FISF permite establecer el valor de ciertos parámetros del AG seleccionando el botón *Propiedades* (ver figura 11.13). Entre los parámetros que se pueden cambiar están el *número de individuos de la generación*, el *número máximo de epochs o generaciones*, el *threshold de acierto*, el *error mínimo del acierto*, la *operación de cruce*, la *probabilidad de cruce*, el *porcentaje de los mejores individuos a cruzar*, la *operación de mutación*, la *probabilidad de mutación de los individuos*, la *probabilidad de mutación de las variables* y el *rango de la mutación*.

- **Número de individuos de la población:** Este número de individuos se mantendrá estable durante todo el entrenamiento.
- **Número máximo de epochs o generaciones:** Sirve para limitar el tiempo de entrenamiento. Por defecto, si después de 200 generaciones no se ha encontrado la solución óptima, se terminará el entrenamiento.
- **Threshold del acierto:** Sólo se considerarán aciertos los casos en los que los resultados obtenidos coinciden con los resultados deseados con una probabilidad superior a este umbral.
- **Error mínimo del acierto:** Se trata del parámetro de dos de las condiciones de salida. Si durante tres generaciones no se mejora el acierto medio de la población o el acierto del mejor individuo como mínimo esta cantidad, se termina el proceso de entrenamiento.
- **Operación de cruce:** Mediante esta característica se indica qué método de cruce se quiere utilizar durante el entrenamiento. Se ofrecen los métodos explicados en el capítulo 6: *simple*, *Max-Min-Aritmético*, *dinámico*, *dinámico heurístico*, *Blend Alpha* y *Parent-centric Blend Alpha*.
- **Probabilidad de cruce:** Se trata de la probabilidad de que dos de los individuos seleccionados se crucen.

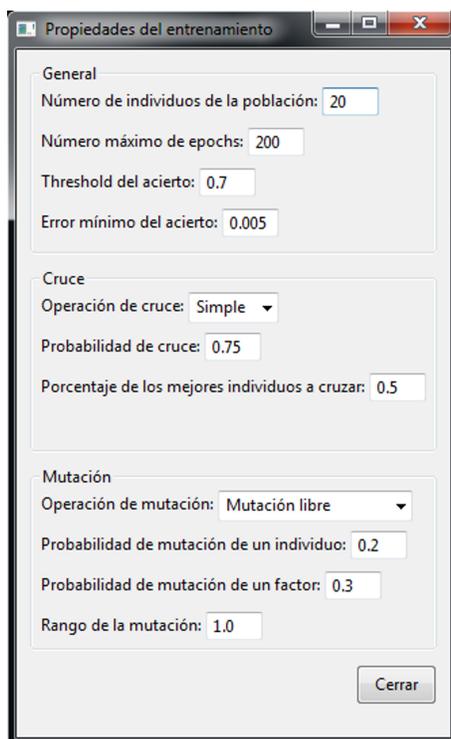


Figura 11.13. Propiedades del algoritmo genético de Aide-FISF

- **Porcentaje de los mejores individuos a cruzar:** Este es el porcentaje utilizado para realizar la *selección por truncamiento*.
- **Operación de mutación:** Sirve para indicar el método de mutación que se quiere utilizar durante el entrenamiento. Se ofrecen los tres métodos explicados en el capítulo 6: *aleatoria*, *gaussiana* y *de Michalewicz*.
- **Probabilidad de mutación de un individuo:** Especifica la tasa de mutación de la población. Por defecto, uno de cada cinco individuos de la población podrán sufrir mutaciones. Esto puede considerarse excesivo, pero con el siguiente parámetro se atenúa el efecto.
- **Probabilidad de mutación de una variable:** Especifica la probabilidad de que los valores de una variable de entrada sufran una mutación dentro de un individuo elegido para mutar. Por defecto, esta probabilidad está establecida en un tercio, por lo que se modifica una de cada tres variables de entrada de un individuo elegido para mutar. Con esta doble probabilidad de mutación (probabilidad de individuo y de variable) se consigue que sean bastantes los individuos que puedan cambiar, aunque cada uno de ellos no se modifique excesivamente. Mediante ensayo y error se ha comprobado

que este método obtiene mejores resultados que una mutación que realice cambios muy importantes en pocos individuos.

- **Rango de la mutación:** Esta propiedad especifica cuánto puede cambiar un valor elegido para mutar, dentro de su intervalo de valores permitidos, lo que en el capítulo 6 se ha denominado  $[a_i, b_i]$ . Un valor de 1 significa que podrá coger cualquier valor de su intervalo de valores permitidos, mientras que un valor de 0 significa que el valor no sufrirá ningún cambio.

El entrenamiento se puede iniciar pulsando el botón *Entrenar*. Una vez haya terminado el entrenamiento se pueden trasladar los valores del mejor individuo encontrado por el algoritmo al modelo de SID creado. Este proceso se realiza mediante el botón *Actualizar*.

#### 11.2.2.4. Ejecución de modelos de Sistemas de Inferencia Difusa

Esta funcionalidad permite probar el modelo de SID definido para comprobar si su comportamiento es correcto. El modelo se puede ejecutar de dos formas diferentes. Por un lado, se permite la ejecución de casos introducidos mediante un *formulario* en la sección superior de la aplicación de ejecución (ver figura 11.14). Por otro lado, los casos a ejecutar se pueden introducir como *expresiones* de valores separados por comas en la sección inferior de la aplicación de ejecución.

The screenshot shows a window titled 'Aplicación de ejecución'. It contains several input fields and a list of radio buttons. The fields are: 'Fecha de nacimiento (formato dd-mm-aaaa):', 'Amonio:', 'Insulina:', and 'Glucemia:'. The radio buttons are under the heading 'Gasometría (pH en sangre):' and include options: 'Muy bajo', 'Bajo', 'Discretamente bajo', 'Normal', 'Discretamente elevado', and 'Elevado'. At the bottom, there is a text area labeled 'Valores separados por comas (CSV):' and three buttons: 'Ejecutar expresión CSV', 'Ejecutar datos de formulario', and 'Cerrar'.

Figura 11.14. Funcionalidad de ejecución de Aide-FISF

En la opción del **formulario** el usuario introduce los valores correspondientes a cada una de las variables de entrada. El formulario se crea de forma

dinámica a partir del metamodelo de datos de entrada presentado en el capítulo 10. Por lo tanto, sus componentes dependen del número y tipo de las variables de entrada. Así, para las variables simples y difusas (*Amonio*, *Insulina* y *Glucemia* en la figura 11.14) el usuario debe introducir un valor numérico, mientras que para las variables de entrada discretas (caso de la *Gasometría (pH en sangre)*) debe elegir una de las opciones que se le ofrecen. Las variables complejas no se introducen ya que su valor de entrada se infiere a partir de los valores de las variables de entrada simples que las componen. Si alguna de las variables de entrada está condicionada, el usuario también debe introducir los datos del paciente correspondientes a esas condiciones. En la parte superior izquierda de la figura 11.14 se muestra la opción para introducir la fecha de nacimiento del paciente, ya que la variable *Amonio* está condicionada por la edad del paciente. El botón *Ejecutar datos del formulario* comienza la ejecución, antes de la cual se comprueba la validez de los datos introducidos.

En cuanto a la opción de la **expresión**, el usuario introduce los datos de entrada separados por comas en el mismo orden en que se han definido las variables de entrada. Si alguna variable de entrada está condicionada el valor correspondiente a su condición se introducirá al comienzo de la expresión. En el extracto 11.2 se muestra una expresión válida correspondiente a las variables de entrada mostradas en la figura 11.14. El botón *Ejecutar expresión CSV* primero comprueba la validez de la expresión introducida y, si no hay ningún error, realiza la ejecución de la misma.

2014-03-25 , Bajo , 203 , 179 , 130

**Extracto 11.2.** Expresión de datos de entrada

Independientemente del sistema de ejecución seleccionado los resultados se muestran en una página HTML, con la intención de que sean totalmente interpretables y estén plenamente justificados.



## Evaluación de la solución propuesta

En este capítulo se presentan los experimentos que se han realizado para evaluar tanto los metamodelos, como las transformaciones y las distintas herramientas desarrolladas y explicadas en la solución propuesta. Primero se ha llevado a cabo un experimento para medir la *eficacia de Aide-DDE y Aide-FISF para desarrollar SID en términos de tiempo*. Posteriormente se ha realizado otro experimento para evaluar la *eficacia de los SID generados en términos de tiempo y acierto*.

El experimento para medir la **eficacia de Aide-DDE y Aide-FISF para desarrollar SID en términos de tiempo** ha servido para comparar el tiempo que lleva desarrollar un SID ejecutable manualmente (programarlo) y el tiempo que se necesita para desarrollar ese mismo SID mediante las herramientas Aide-DDE y Aide-FISF. De esta forma se ha podido concluir si la tecnología desarrollada en esta tesis es adecuada o no para usarse con el objetivo para el que se creó.

El experimento para evaluar la **eficacia de los SID generados en términos de tiempo y acierto** ha medido la efectividad de un SID desarrollado mediante Aide-FISF utilizando casos reales. Por un lado, se ha medido la tasa de acierto del SID, y por otro lado, se ha estimado el tiempo necesario por profesionales de laboratorio para evaluar los casos y se ha comparado con el tiempo que ha necesitado el SID para evaluar los casos. Así, se ha podido comprobar si los SID creados mediante Aide-FISF son adecuadas o no para dar solución a problemas de clasificación en el ámbito en que se han probado.

A continuación se explica de forma más detallada cada uno de los experimentos. Posteriormente se realiza una evaluación de los resultados obtenidos.

### 12.1. Evaluación de la eficacia de Aide-DDE y Aide-FISF

Este experimento se ha llevado a cabo en el ámbito de la *hiperamonemia*. La **hiperamonemia** es un trastorno metabólico resultante de un alto nivel

de amonio en sangre. Se trata de una urgencia médica que debe reconocerse de manera precoz, ya que de lo contrario puede derivar en graves secuelas neurológicas e incluso la muerte. La hiperamonemia es un síntoma y suele ser consecuencia de algún trastorno. El interés médico consiste en identificar el trastorno causante para que, tratándolo, remita la hiperamonemia.

El grupo Erabaki ha desarrollado una GCI basada en la segunda edición del *Protocolo Hispano-Luso de diagnóstico y tratamiento de las hiperamonemias en pacientes neonatos y de más de 30 días de vida* (Couce et al., 2009). Esta GCI ha sido revisada y avalada por expertos independientes de la Asociación Española para el Estudio de los Errores Congénitos del Metabolismo (AECOM, 2017) y actualmente está disponible en el portal web Aide-Portal (Segundo et al., 2009).

Este experimento ha utilizado dos DD contenidos en la mencionada GC: El primero de los DD se utiliza para obtener una *sospecha* sobre la posible causa de la hiperamonemia a partir de un análisis de sangre básico, mientras que el segundo de los DD se realiza en base a los datos de un análisis específico y sirve para identificar de forma *definitiva* el trastorno causante de la hiperamonemia.

El objetivo del **DD de sospecha** de hiperamonemia es ofrecer un diagnóstico aproximado al usuario, ya que se trata de una prueba que hay que realizar de forma inmediata y que se basa en un análisis básico de sangre. Debido a la sencillez del diagnóstico, algunos trastornos están agrupados en base a procedimientos de actuación comunes. Así, por ejemplo, se dispone de un conjunto de trastornos denominado *Trastornos de la beta oxidación de los ácidos grasos* que agrupa a seis trastornos diferentes. En la figura 12.1 se muestra el DD completo. En total, el DD de sospecha utiliza 7 variables clínicas (dispuestas en la primera fila de la tabla, una en cada columna) del análisis básico para discriminar entre 12 trastornos o conjuntos de trastornos (dispuestos en la primera columna de la tabla, cada uno en una fila).

En cuanto al **DD definitivo**, su objetivo es identificar de forma inequívoca el trastorno que causa la hiperamonemia en el paciente, de forma que se pueda tratar específicamente. Para realizar este DD se necesita un análisis de sangre específico que mida los niveles de aminoácidos en sangre y orina, así como de ácidos orgánicos, acilcarnitinas, glucemia e insulina. En total, el DD tiene en cuenta 73 variables clínicas (11 aminoácidos en sangre, 9 aminoácidos en orina, 24 ácidos orgánicos, 28 acilcarnitinas y la proporción entre la glucemia y la insulina) para discriminar entre 23 trastornos (páginas 34 y 35 de (Couce et al., 2009)). Algunos conjuntos de trastornos del DD de sospecha se han separado en trastornos específicos. Así, por ejemplo, el conjunto *Trastornos de la beta oxidación de los ácidos grasos* se ha dividido en los trastornos *Deficiencia de acil-CoA dehidrogenasa de cadena larga (LCAD)*, *Deficiencia de 3-hidroxiacil-CoA dehidrogenasa de cadena larga (LCHAD)*, *Deficiencia de acil-CoA dehidrogenasa de cadena media (MCAD)*, *Deficiencia de carnitina-acilcarnitina translocasa (CACT)*, *Deficiencia de carnitina palmitoiltransferasa I (CPT I)* y *Deficiencia de carnitina palmitoiltransferasa II (CPT II)*.

Diagnóstico diferencial de la etiología más probable						
	Gasometría (7.35 < pH normal < 7.45)	Anionio (Normal < 50 mmol/L, Normal < 90 mg/dl, Niño > 30 días)	Lactato (Normal < 2.5 mM, Normal < 20 mg/dl)	GOT-GPT (Normal < 40 UI/L)	CPK (Normal < 190 UI/L)	3-Oxobutirato (Hay síntesis de cetónicos (Normal > 0.5 mM)
Trastornos de la cetogénesis	Muy bajo	Normal o discretamente elevado	Normal o discretamente elevado	Elevado	Elevado	Bajo
Trastornos de la beta oxidación de los ácidos grasos	Bajo	Normal o discretamente elevado	Elevado	Elevado	Elevado	Bajo
Aciduria glutárica tipo 2	Bajo	Elevado	Elevado	Elevado	Elevado	Bajo
Hiperinsulinemia + Hiperamonemia	Normal o discretamente elevado	Normal o discretamente elevado	Normal	Normal	Normal	Bajo
Trastornos del ciclo de la urea y/o Intoxicación por administración de ácido valproico	Normal o discretamente elevado	Muy elevado o Crítico	Normal o discretamente elevado	Elevado	Normal	Normal o discretamente bajo
Síndrome de Járabe de Arce	Normal	Normal o discretamente elevado	Normal o discretamente elevado	Normal	Normal	Elevado
Acidemia isovalérica y/o 3-metilcrotónica	Bajo	Elevado o Muy elevado	Elevado	Normal o discretamente elevado	Normal	Muy elevado
Acidemia propiónica y/o metilmalónica	Muy bajo	Muy elevado	Elevado	Normal o discretamente elevado	Normal o discretamente elevado	Elevado
Deficiencia múltiple de carboxilasas	Muy bajo	Elevado	Elevado	Normal o discretamente elevado	Normal	Elevado
Atrofia gýrata	Normal o discretamente elevado	Elevado o Muy elevado	Normal	Normal o discretamente elevado	Normal	Normal o discretamente bajo
Hiperamonemia-Hiperornitinemia-Homocitrulinuria	Normal o discretamente bajo	Elevado	Normal	Normal	Normal	Normal o discretamente bajo
Defectos de transporte de aminoácidos dibásicos, cistinuria e intolerancia a proteínas con lisinuria	Normal	Elevado	Normal	Normal	Normal	Normal o discretamente bajo

Reproducción autorizada por EFPA04/2019

Figura 12.1. DD de sospecha de la hiperamonemia (extraído y adaptado de la página 33 de (Couce et al., 2009))

En la tabla 12.1 se pueden ver las características principales de estos dos DD: número de trastornos, número de variables clínicas y número de relaciones. En esta tabla se puede observar que el diagnóstico definitivo es más complejo que el de sospecha, ya que el número de elementos es superior en todos los casos. Por otro lado, se puede concluir que en el DD de sospecha todos los trastornos están relacionados con todas las variables clínicas ( $12 \cdot 7 = 84$ ), cosa que no ocurre en el DD definitivo ( $28 \cdot 73 \neq 285$ ).

DD de hiperamonemia	Nº Trastornos	Nº Variables clínicas	Nº Relaciones
Sospecha	12	7	84
Definitivo	28	73	285

**Tabla 12.1.** Resumen de los DD de la hiperamonemia

El proceso del experimento ha sido el siguiente: un experto médico ha utilizado Aide-DDE para crear los modelos de los DD comentados y posteriormente un ingeniero del conocimiento ha creado los modelos de SID correspondientes utilizando Aide-FISF. Una vez modelados los SID, se han generado las versiones ejecutables correspondientes en los lenguajes CLIPS/FuzzyCLIPS y JESS/FuzzyJESS. Los SID en lenguaje CLIPS/FuzzyCLIPS han sido los que finalmente se han integrado en la GCI de la hiperamonemia. Por otro lado, otro ingeniero del conocimiento con avanzado conocimiento sobre SID ha desarrollado los mismos SID ejecutables de forma manual. En ambos procesos de producción el mismo experto médico ha asesorado a los dos ingenieros del conocimiento.

En la tabla 12.2 se muestran los tiempos que se han medido durante el desarrollo de cada uno de los pasos anteriores para el DD de sospecha. El experto médico ha necesitado alrededor de **1 hora** para llevar a cabo el modelado del DD. Como consideraciones específicas de este DD se pueden mencionar que a los trastornos se les ha añadido un vínculo con información sobre los mismos en el sitio web Orphanet (Inserm, 2017), las variables clínicas tienen distintas cantidades de valores lingüísticos (por ejemplo, la variable clínica *Gasometría* tiene seis valores y la variable *Lactato* tiene tres) y a todas las relaciones se les ha establecido el mismo peso. Estas consideraciones se han incluido en base a la información contenida en la GC y en base a la experiencia del experto médico.

DD sospecha	Edición con Aide-DDE y Aide-FISF	Edición manual
Creación del modelo de DD	1 h	
Creación del modelo de SID	5 h	
Generación en lenguaje CLIPS/FuzzyCLIPS	1.5 seg	
Generación en lenguaje JESS/FuzzyJESS	1.3 seg	
Creación de SID en lenguaje CLIPS/FuzzyCLIPS		14 h
Creación de SID en lenguaje JESS/FuzzyJESS		8.5 h
Total de tiempo	6 h	22.5 h

**Tabla 12.2.** Tiempo necesario para crear el SID relativo al DD de sospecha de hiperamonemia

El ingeniero del conocimiento, por su lado, ha necesitado alrededor de **5 horas**, incluyendo el testeo y la corrección de errores del modelo. En base a la información de la GC y la experiencia del experto médico también se han tomado una serie de consideraciones en el modelo de SID, tales como que todas las variables de entrada sean difusas, lo que ha conllevado que se hayan tenido que definir los conjuntos difusos de todos los valores lingüísticos, y que la variable de entrada *Amonio* esté condicionada por la edad.

Finalmente, la generación de los SID ejecutables se ha llevado a cabo en **1.5 segundos** para el lenguaje CLIPS/FuzzyCLIPS y **1.3 segundos** para el lenguaje JESS/FuzzyJESS.

Paralelamente, el ingeniero del conocimiento encargado de definir el SID ejecutable de forma manual ha necesitado alrededor de **14 horas** para definir un SID desde cero en lenguaje CLIPS/FuzzyCLIPS. Posteriormente ha reutilizado parte del trabajo previo para redefinir el SID en lenguaje JESS/-FuzzyJESS, para lo que ha necesitado **8.5 horas**.

En la tabla 12.3 se muestran los tiempos de los pasos de desarrollo del SID relativo al DD definitivo de hiperamonemia. El experto médico ha necesitado **2**

**horas** para crear el modelo de DD. Al igual que en el modelo de DD, también se han tomado ciertas consideraciones en base a la experiencia del médico y de la información de la GC. Por ejemplo, los trastornos incluyen un vínculo a información sobre los mismos en el sitio web *Online Mendelian Inheritance in Man* (OMIM, 2017). Otra consideración ha sido que todas las variables clínicas tienen entre dos y tres valores lingüísticos: *Normal* y *Elevado* para las variables de dos valores y *Bajo*, *Normal* y *Elevado* para las de tres valores. Por último, se ha decidido que las relaciones tengan distintos pesos. Sin ir más lejos, para diagnosticar los distintos trastornos de la betaoxidación de los ácidos grasos las variables clínicas agrupadas dentro de las acilcarnitinas son las más importantes, y por lo tanto, las relaciones que unen estas variables clínicas con los trastornos de la betaoxidación tienen un mayor peso. Sin embargo, para los trastornos del ciclo de la urea las variables más significativas son las que se agrupan dentro de los aminoácidos, por lo que en este caso son las relaciones que unen los aminoácidos a los trastornos del ciclo de la urea las que tienen mayor peso.

DD definitivo	Edición con Aide-DDE y Aide-FISF	Edición manual
Creación del modelo de DD	2 h	
Creación del modelo de SID	9 h	
Generación en lenguaje CLIPS/FuzzyCLIPS	3 seg	
Generación en lenguaje JESS/FuzzyJESS	2.6 seg	
Creación de SID en lenguaje CLIPS/FuzzyCLIPS		22 h
Creación de SID en lenguaje JESS/FuzzyJESS		17.5 h
Total de tiempo	11 h	39.5 h

**Tabla 12.3.** Tiempo necesario para crear el SID relativo al DD definitivo de hiperamonemia

El ingeniero de conocimiento ha necesitado **9 horas** para crear el modelo de SID correspondiente al DD, incluyendo las pruebas y la corrección de errores inherentes del proceso de edición. Para este modelo únicamente se ha tenido en cuenta una consideración, en concreto, que 72 de las 73 variables de entrada sean discretas y una, la *proporción entre la insulina y la glucemia*, compleja. Esto ha conllevado que se hayan tenido que definir dos variables simples para la *Insulina* y la *Glucemia*.

Por último, la generación automática a partir del modelo de SID en los lenguajes CLIPS/FuzzyCLIPS y JESS/FuzzyJESS se ha llevado a cabo en **3** y **2.6 segundos** respectivamente.

En paralelo, el otro ingeniero del conocimiento asesorado por el mismo experto médico ha necesitado **22 horas** para crear el SID ejecutable correspondiente al DD definitivo de hiperamonemia en lenguaje CLIPS/FuzzyCLIPS, mientras que ha necesitado unas **17.5 horas** para redefinirlo en lenguaje JESS/FuzzyJESS.

Para obtener una imagen lo más objetiva posible, también hay que tener en cuenta el tiempo que se ha necesitado para desarrollar la solución propuesta. Por lo tanto, en la tabla 12.4 se muestra el tiempo que se ha necesitado para llevar a cabo cada una de las tareas de la solución propuesta. La definición del metamodelo de DD se ha realizado en 11.5 horas, lo que incluye la identificación de las clases que lo forman con ayuda de expertos médicos, así como su implementación. El metamodelo de SID, por su lado, se ha definido en 19 horas y los metamodelos sobre los datos de entrada y de resultados en 3 horas. En cuanto a las transformaciones, las transformaciones M2M entre los metamodelos de DD y SID se han definido en 4.5 horas. Para definir las distintas plantillas de las transformaciones M2T se ha necesitado un total de 34 horas. Por último, el diseño e implementación de Aide-DDE y Aide-FISF, así como la integración de los procesos de transformación en las herramientas ha llevado un total de 123 horas. Todo esto hace un total de 195 horas.

## 12.2. Evaluación de la eficacia de los SID generados

Este experimento se ha llevado a cabo dentro del ámbito del *cribado neonatal*. El **cribado neonatal** (también conocido como prueba del talón) es una prueba para la detección precoz de trastornos provenientes de un error innato del metabolismo (EIM) en recién nacidos. El método consiste en recoger una muestra de sangre en las primeras 48 horas de vida del recién nacido mediante una punción en el talón. Esta muestra se procesa posteriormente mediante una técnica denominada espectrometría de masas en tándem, que permite la separación, identificación y cuantificación de los metabolitos de la muestra. En base a estos metabolitos se puede saber si el neonato puede sufrir una determinada enfermedad metabólica o estar en un grupo de riesgo de sufrirla. Los laboratorios encargados de realizar los programas de cribado

Tarea de desarrollo	Tiempo (h)
Definición del metamodelo de DD	11.5
Definición del metamodelo de SID	19
Definición de los metamodelos de datos de entrada y resultados	3
Transformaciones M2M	4.5
Transformaciones M2T	34
Diseño e implementación de Aide-DDE y Aide-FISF	123
Total de tiempo	195

**Tabla 12.4.** Tiempo necesario para desarrollar la tecnología de la tesis

neonatal deben analizar miles de muestras al año, por lo que el proceso de cribado es muy costoso en términos de tiempo.

El cribado neonatal no es un diagnóstico. Esta prueba sirve para la detección precoz de neonatos con mayor probabilidad de sufrir alguno de los trastornos. Sin embargo, un resultado positivo no implica que vaya a sufrir dicho trastorno. Una vez detectado hay que realizar diagnósticos clínicos, bioquímicos e incluso genéticos para confirmar el resultado del cribado.

El grupo Erabaki también ha creado una GCI para realizar el proceso de cribado neonatal y actualmente está disponible en Aide-Portal. El SID lo ha desarrollado un ingeniero del conocimiento utilizando Aide-FISF con la colaboración de un experto médico. A continuación primero se explica el diseño del modelo de SID para seguidamente especificar las pruebas realizadas y los resultados obtenidos.

### 12.2.1. Diseño de un modelo de SID para el cribado neonatal

El DD de cribado neonatal utiliza un total de 75 variables clínicas para discriminar entre 48 trastornos diferentes. Los trastornos se dividen en tres grandes conjuntos: trastornos de aminoácidos, trastornos de ácidos orgánicos y trastornos de ácidos grasos. En el anexo A se muestra la lista completa de los trastornos que se valoran en el DD, así como los códigos OMIM por los

que se identifican y sus abreviaturas. En adelante se utilizarán estas abreviaturas para hacer referencia a los trastornos. Cabe destacar que la acidemia metilmalónica es un trastorno especial, ya que es un EIM genéticamente heterogéneo que puede tener distintos orígenes. Así, la acidemia metilmalónica asume distintos nombres dependiendo de su origen (ver *MUT*, *Cbl A*, *B* y *Cbl C*, *D* en el anexo A). También hay que indicar que el trastorno *B12 Def (Mat)* se refiere a una falta maternal de vitamina B12 que puede causar acidemia metilmalónica en recién nacidos. Por otro lado, la acidemia propiónica es un EIM difícil de diferenciar de la acidemia metilmalónica utilizando un análisis bioquímico (Seashore, 2009).

En cuanto a las variables clínicas, se tienen en cuenta los niveles en sangre de 15 aminoácidos, 27 acilcarnitinas y 33 proporciones entre esas variables (ver anexo B). Todas las variables clínicas no están relacionadas con todos los trastornos. De hecho, cada trastorno está relacionado de media con unas cuatro o cinco variables clínicas. Toda la información relativa a los trastornos, las variables clínicas y las relaciones que las unen se ha obtenido de distintas fuentes bibliográficas: (Soria et al., 2009), (Vilarinho et al., 2010), (McHugh et al., 2011), (Lund et al., 2012) y (Trefz et al., 2017). En la tabla 12.5 se muestra un resumen del DD de cribado neonatal, mientras que en el anexo C se muestra el DD completo.

DD	Nº Trastornos	Nº Variables clínicas	Nº Relaciones
Cribado neonatal	48	75	203

**Tabla 12.5.** Resumen del DD de cribado neonatal

Durante la edición del modelo de SID se han tenido en cuenta ciertas consideraciones en base a la experiencia del experto médico y a la bibliografía. Así, las variables de salida disponen de un vínculo a su página correspondiente en el sitio web OMIM. En cuanto a las variables de entrada, se ha decidido que tanto los 15 aminoácidos como las 27 acilcarnitinas sean variables de entrada difusas, mientras que las 33 proporciones sean variables complejas. En todos los casos, a las variables de entrada se les han asignado tres valores lingüísticos: *Bajo*, *Normal* y *Alto*. A cada uno de los valores lingüísticos se le ha añadido su conjunto difuso correspondiente en base a los datos de la literatura.

Una vez terminada la edición del SID y tras comprobar su funcionamiento, se ha creado el SID ejecutable correspondiente en lenguaje CLIPS/Fuzzy-CLIPS y se ha integrado dentro de la GCI para el cribado neonatal. El proceso

de definición del SID ha costado en términos de tiempo un total de **24.5 horas**.

### 12.2.2. Pruebas realizadas

El SID se ha probado con un total de 152 muestras reales. 112 muestras pertenecen a sujetos sanos o sujetos sintomáticos pero no sufriendo un trastorno proveniente de un EIM, mientras que 40 pertenecen a sujetos con un trastorno proveniente de un EIM. Las muestras provienen de tres fuentes distintas y se han dividido en dos conjuntos: *muestras de cribado* y *muestras clínicas*. En la tabla 12.6 se muestra un resumen de las muestras utilizadas para evaluar el SID.

Conjunto de muestra	Valor de muestra	Número de muestras	Total de muestras
Muestras de cribado	TYR I y TYR II	10	105
	Sanos	95	
Muestras clínicas	PROP	6	47
	MCAD	5	
	GA I	4	
	Cbl A, B y MUT	4	
	CUD	2	
	GA II	2	
	CPT II	1	
	HMG	1	
	IVA	1	
	LCHAD	1	
	B12 Def (mat)	1	
	MCD	1	
	VLCAD	1	
	No EIM	17	

**Tabla 12.6.** Resumen de las muestras utilizadas para evaluar el SID

Las *muestras de cribado* corresponden a un conjunto de sujetos neonatos. De un total de 105 muestras, 10 pertenecen a sujetos sufriendo algún trastorno proveniente de un EIM y el resto pertenecen a sujetos sanos.

Las *muestras clínicas*, por su lado, pertenecen a sujetos sintomáticos. De las 47 muestras que forman este conjunto, 30 pertenecen a pacientes sufriendo

un trastorno proveniente de un EIM y 17 pertenecen a sujetos que no sufren un trastorno proveniente de un EIM, pero que tienen una presentación sintomática parecida.

Los resultados de las pruebas realizadas se muestran resumidos en la tabla 12.7. La versión extendida de los resultados se puede consultar en el anexo D. De las 105 muestras de cribado el SID ha identificado correctamente las 10 muestras de tirosinemias, así como las 95 muestras restantes pertenecientes a sujetos sanos. En lo que a las muestras clínicas se refiere, el SID de cribado ha identificado correctamente las 17 muestras de sujetos sin trastornos provenientes de un EIM. En otras 13 muestras ha asignado la probabilidad más alta al trastorno correcto. En 8 de las muestras clínicas el trastorno correcto no ha obtenido la probabilidad más alta, pero está dentro de un conjunto de trastornos muy parecidos, donde alguno de estos trastornos sí tiene la probabilidad más alta. Esto ha ocurrido con los trastornos relativos a las acidemias propiónicas y metilmalónicas. Para el caso de 4 muestras, el trastorno correcto sí aparece en la lista de resultados, pero otro trastorno ha obtenido la probabilidad más alta. En otras 4 muestras el SID no ha sido capaz de indicar en la lista de resultados el trastorno correcto. Por último, se ha producido un falso-negativo, ya que una muestra se ha dado por normal y realmente pertenece a un sujeto con un trastorno proveniente de un EIM.

En cuanto al tiempo necesario por el SID para analizar las muestras, las muestras de cribado se han analizado en **103 segundos** y las muestras clínicas se han analizado en **23 segundos**. En total el SID ha necesitado **126 segundos** para analizar las 152 muestras.

En base a una estimación a partir de información proveniente de profesionales de laboratorio, un profesional necesita alrededor de **30 segundos** por muestra para aquellas muestras que sean evidentemente pertenecientes a sujetos sanos, mientras que con algunas muestras patológicas pueden necesitar unos **5 minutos**. Extrapolando esta información a las muestras con las que se han realizado las pruebas se obtiene que un profesional habría necesitado **202 minutos** para evaluar las 152 muestras.

### 12.3. Evaluación de los resultados

En este apartado se exponen y se valoran los resultados obtenidos en las pruebas realizadas. Primero se comparan los resultados sobre la eficiencia de las herramientas desarrolladas para definir SID ejecutables frente a la implementación manual. Posteriormente se evalúan los resultados de la eficiencia de los SID creados en términos de acierto y tiempo.

#### 12.3.1. Eficacia de Aide-DDE y Aide-FISF para desarrollar SID en términos de tiempo

Para desarrollar el modelo de SID correspondiente al DD de sospecha utilizando las herramientas Aide-DDE y Aide-FISF el experto médico y el

Resultado	Nº de muestras de cribado	Nº de muestras clínicas	Total
Sujetos sin trastornos provenientes de EIM	95	17	112
Trastorno correcto con probabilidad más alta	10	13	23
Trastorno correcto incluido en conjunto de trastornos muy parecidos y con la probabilidad más alta		8	8
Trastorno correcto en la lista de resultados pero sin la probabilidad más alta		4	4
Trastorno correcto no en la lista de resultados		4	4
Falso-positivo			0
Falso-negativo		1	1

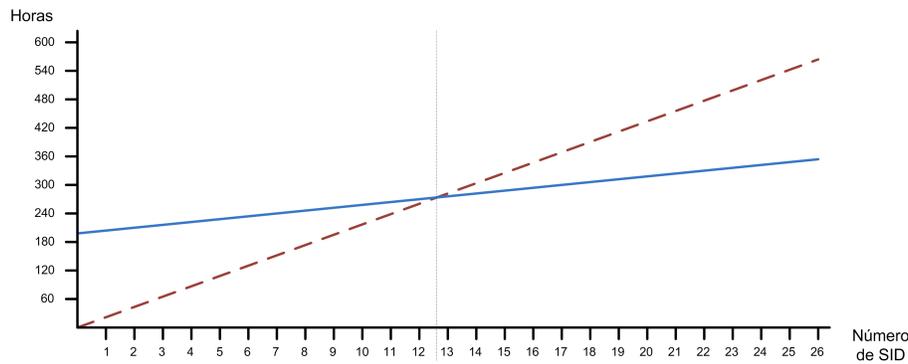
**Tabla 12.7.** Resumen de los resultados obtenidos

ingeniero del conocimiento han necesitado un total de 6 horas. Teniendo en cuenta que la generación automática del SID ejecutable en lenguaje CLIPS/-FuzzyCLIPS ha sucedido en segundos (por lo tanto, un tiempo despreciable), se puede concluir que el SID se ha creado en esas 6 horas. El ingeniero del conocimiento que ha creado el SID equivalente de forma manual ha necesitado 14 horas. La reducción de tiempo de desarrollo por utilizar las herramientas frente a la programación manual es del **57 %**. Teniendo en cuenta que la generación de SID ejecutables en otros lenguajes no lleva tiempo, si se suma a estos resultados el tiempo que ha necesitado el ingeniero del conocimiento para desarrollar el mismo SID en lenguaje JESS/FuzzyJESS (8.5 horas), la reducción del tiempo necesario para crear ambos SID es del **73 %**.

Algo parecido ocurre con el desarrollo del SID ejecutable correspondiente al DD definitivo. Mediante el uso de Aide-DDE y Aide-FISF el experto médico y el ingeniero del conocimiento han necesitado 11 horas para crear el modelo y posteriormente generar el SID ejecutable en lenguaje CLIPS/FuzzyCLIPS. Al igual que en el DD anterior el tiempo de generación es despreciable. El otro ingeniero de conocimiento ha necesitado 22 horas para crear este mismo SID. Con todo ello, la reducción de tiempo es del **50 %**. Si se tiene en cuenta el tiempo necesario para crear el SID ejecutable en lenguaje JESS/FuzzyJESS (17.5 horas) la mejoría del tiempo necesario para el desarrollo es del **72 %**.

Sin embargo, no hay que olvidar el tiempo que ha llevado desarrollar las propias herramientas, así como los metamodelos y las transformaciones de las que hacen uso. Este tiempo es necesario para poder comparar de forma fidedigna el desarrollo utilizando herramientas y el desarrollo manual.

Visto el tiempo que se ha necesitado para desarrollar toda la tecnología (195 horas) y teniendo en cuenta que una vez que se dispone de la tecnología la creación de SID se agiliza mucho, se puede concluir que el uso de las herramientas, o más bien, la creación de las herramientas es rentable siempre que el número de SID que se desarrollen sea alto. Lógicamente, cuanto más complejo o grande sea el DD original (mayor número de trastornos y de variables clínicas), mayor será la ventaja de utilizar las herramientas. En la figura 12.2 se muestra la relación entre el número de SID desarrollados y el número de horas necesarias para desarrollarlos teniendo como medida el tiempo necesario para crear el DD definitivo de la hiperamonemia (11 horas utilizando las herramientas y 22 horas manualmente). Con una línea discontinua se muestra la relación de horas del desarrollo manual y con una línea continua la relación de horas del desarrollo con las herramientas. Viendo el gráfico se puede concluir que a partir de 13 DD sale rentable en términos de tiempo crear primero una infraestructura para el desarrollo de SID.



**Figura 12.2.** Relación entre el número de SID desarrollados y el tiempo necesario para desarrollarlos, utilizando Aide-DDE y Aide-FISF (línea continua) y manualmente (línea discontinua)

Con todo esto se puede decir que queda probado que las herramientas desarrolladas, por lo menos en términos de tiempo, son adecuadas para la definición de SID. Por último, cabe añadir que las herramientas no sólo facilitan y optimizan el proceso de desarrollo de SID, sino que además facilitan las modificaciones y la corrección de errores, ya que éstas se realizan a nivel de modelo y no a nivel de código como pasa con los SID definidos manualmente.

### **12.3.2. Eficacia de los SID generados en términos de acierto y tiempo**

En cuanto a los resultados obtenidos por el SID de cribado neonatal, el SID ha sido capaz de identificar correctamente al 100 % de los individuos sanos, así como a todos los individuos sintomáticos pero sin trastornos provenientes de un EIM. Esto significa que no ha habido ningún falso-positivo. En lo que a sujetos con trastornos provenientes de un EIM, el SID ha identificado correctamente el trastorno correcto con la probabilidad más alta en un **57.5 %** de los casos (23 de 40 casos). En otros 8 casos (**20 %**) el trastorno correcto se ha clasificado con una probabilidad muy alta y entre un conjunto de trastornos muy parecidos. Esto ha ocurrido en el caso de aquellas muestras pertenecientes a sujetos afectados por acidemias propiónicas y metilmalónicas. En 4 casos el SID ha añadido el trastorno correcto en la lista de resultados (**10 %**), pero ha identificado otro trastorno con la probabilidad más alta. Finalmente, el SID no ha sido capaz de identificar el trastorno correcto en 4 de las muestras (**10 %**) y se ha producido un falso-negativo (**2.5 %**). Hay que mencionar que aquellas muestras que el SID no ha sido capaz de identificar correctamente pertenecían bien a sujetos en edad adulta (por lo tanto los intervalos de referencia de neonatos no son precisos), bien a sujetos bajo medicación, bien eran presentaciones muy débiles del trastorno o bien faltaba información sobre alguno de los valores de entrada con mayor importancia.

En lo que al tiempo necesario para procesar la muestra se refiere, el SID ha analizado las 152 muestras en 126 segundos, mientras que un profesional de laboratorio habría necesitado 202 minutos. Esto significa que el uso del SID reduce el tiempo necesario en un **98 %**.

Aunque el número de muestras resulta bastante limitado, los resultados son prometedores en términos de acierto. En cuanto a la ganancia de tiempo, la mejoría es evidente. Incluso en el caso de que el SID sólo se utilizase para discriminar entre aquellos sujetos sanos y los que padecen un trastorno, el beneficio es incuestionable.

## Conclusiones



La parte V **concluye** la exposición del trabajo de investigación realizado.

El capítulo 13 resume las aportaciones realizadas.

El capítulo 14 describe las vías de investigación y desarrollo que quedan abiertas.

Por último, el capítulo 15 enumera los artículos de investigación que han visto la luz durante el transcurso del trabajo realizado.



## Principales aportaciones

En este capítulo se resumen las principales aportaciones a las que ha dado lugar el presente trabajo de tesis doctoral.

Respecto al primer objetivo de la tesis, la definición de recursos basados en MDDM para crear SID a partir de DD, esta tesis ha supuesto las siguientes cinco aportaciones. La primera aportación es la **representación de los DD mediante un metamodelo**. El metamodelo de DD presentado es independiente de la computación, ya que sólo hace uso de conceptos del dominio médico. Esto hace que los modelos de DD sean más comprensibles por parte de los profesionales médicos, de forma que se sientan más cómodos y puedan participar activamente en el proceso de definición de los DD.

La segunda aportación consiste en la **representación de los SID mediante un metamodelo**. El metamodelo de SID presentado es independiente de la plataforma, ya que sólo hace uso de conceptos que son comunes para todos los SID. De esta forma, los diseñadores encargados de definir los SID pueden centrarse en la correcta definición del modelo sin preocuparse por las características concretas de un sistema operativo o lenguaje de programación específico. Cabe destacar también los **modelos de SID para el diagnóstico de la hiperamonemia y el cribado neonatal** desarrollados. A partir de estos modelos se han generado varios SID ejecutables que se han integrado en GCI que siguen la representación Aide. Además, se ha creado una **aplicación de escritorio que permite realizar cribado neonatal a partir de los ficheros de datos obtenidos de un espectrómetro de masas**. El SID utilizado por esta aplicación y por una de las GCI provienen del mismo modelo de SID, lo que da por probado que los modelos de SID son totalmente reutilizables, aunque las soluciones finales en que se van a utilizar tengan funcionamientos diferentes.

La tercera aportación son las **transformaciones M2M entre los modelos de DD y SID**. Estas transformaciones permiten que a los profesionales médicos se sientan más implicados en el proceso al comprobar que el conocimiento plasmado en los modelos de DD se traslada de forma correcta a los

modelos de SID. Por otro lado, utilizando estas transformaciones también se evitan los errores inherentes de la programación manual.

La cuarta aportación son las **transformaciones M2T entre el modelo de SID y los distintos lenguajes de ejecución de SID**. Estas transformaciones permiten generar SID ejecutables en distintos lenguajes en cuestión de segundos y evitando los errores innatos de la programación manual. Además, con la definición de nuevas plantillas se pueden agregar otros lenguajes objetivos de forma relativamente sencilla. La independencia de las transformaciones respecto a los modelos puede incluso permitir la utilización de otros sistemas de clasificación diferentes de los SID. En este caso, sólo habría que definir nuevos metamodelos para esos sistemas de clasificación, así como las transformaciones pertinentes.

La quinta aportación son las **representaciones de los datos de entrada y de los resultados**. El metamodelo de los datos de entrada permite crear distintas interfaces para la introducción de datos en los SID, de forma que los modelos de SID son totalmente independientes de las interfaces que se utilizan para introducir los datos. De la misma forma, el metamodelo de resultados permite mostrar los resultados en distintos formatos dependiendo de la solución objetivo. Así, para el caso de las GCI ejecutables en el navegador web los resultados se pueden mostrar en páginas de HTML, mientras que si los SID se quieren utilizar en aplicaciones de escritorio los resultados se pueden mostrar en documentos estructurados o en ficheros XML que puedan ser posteriormente utilizados por otras aplicaciones.

En cuanto al segundo objetivo de la tesis, el diseño de herramientas para la construcción de DD y SID, esta tesis ha supuesto las siguientes dos aportaciones. La primera aportación es el **editor de DD**, el cual sirve para definir modelos de DD. Este editor está dirigido a profesionales médicos y facilita el trabajo de los mismos, ya que obvia cualquier concepto computacional. Este editor está diseñado para integrarse con el editor Aide-MAT, el editor para la definición de GCI dirigido a los profesionales médicos.

La segunda aportación es el **framework para SID** que permite editar modelos de SID y trabajar sobre ellos. Este editor incluye la transformación entre modelos de DD y SID. También incluye funcionalidades para generar SID ejecutables a partir de los modelos de SID y para simular ejecuciones de los modelos de SID. Este editor se ha diseñado para ser integrado en el editor Aide-KAT, el editor para la definición de GCI dirigido a ingenieros del conocimiento. Las dos herramientas creadas mantienen el método de desarrollo de GCI de Aide-GTP (dos niveles de editores) así como su experiencia de usuario. Todo ello permite crear SID en base a los DD contenidos en las GC mientras éstas están siendo transformadas a sus equivalentes informatizadas. De esta forma se facilita y agiliza el proceso de desarrollo de GCI.

Por último, respecto al tercer objetivo de la tesis, el uso de técnicas de aprendizaje automático para la mejora de los SID creados, esta tesis aporta el **uso de un AG para el entrenamiento de los modelos de SID**. Está contrastado que los AG son un método de entrenamiento adecuado para la

mejora de las prestaciones de los SID. Aunque el uso de sistemas de aprendizaje automático no es una novedad en lo que a *frameworks* para el desarrollo de SID siguiendo la metodología MDL se refiere, en el ámbito de la medicina las soluciones creadas *ad hoc* que utilizan SID previamente entrenados siguen siendo una minoría. Además, el AG implementado no está cerrado y es totalmente parametrizable por los usuarios. Éstos pueden modificar casi todos los parámetros del AG, desde el número de individuos de la población hasta los porcentajes de individuos a cruzarse y mutar pasando por los propios métodos de cruce y mutación.



## Líneas futuras

En este capítulo se explican los trabajos futuros que podrían completar la investigación presentada en esta tesis, los cuales se resumen en:

- Ofrecer más opciones de confección de SID.
- Añadir nuevos sistemas de entrenamiento de SID.
- Añadir nuevos lenguajes de ejecución de SID.
- Añadir nuevos modelos de sistemas de clasificación.
- Ofrecer una representación gráfica del conocimiento.
- Ofrecer soporte para la gestión del conocimiento médico.

**Aumentar las opciones de confección de SID** consiste en permitir utilizar más tipos de funciones de implicación y de FP. En este momento, los SID utilizados en la tesis hacen uso de una única función de implicación (mínimo) y una única FP (trapezoidal). Aunque el formalismo de SID utilizado no permite añadir funciones de agregación y desfuzzificación, la inclusión de otras funciones de implicación y FP permitirá que los usuarios tengan mayor libertad a la hora de diseñar los SID.

La **adición de otros sistemas de entrenamiento** consiste en poder utilizar otros algoritmos de entrenamiento diferentes para mejorar las características de los SID. Aunque los AG han demostrado ser una opción que ofrece buenos resultados, puede que haya otros sistemas de entrenamiento que sean más efectivos en términos de eficiencia y tiempo con algunos tipos de SID concretos. Además, la mejora no tiene por qué limitarse a mejorar las FP del SID, se pueden incluir sistemas de entrenamiento que también mejoren la base de conocimiento del SID.

La **adición de otros lenguajes de ejecución** consiste en añadir nuevos lenguajes objetivos a los que transformar los modelos de SID. Actualmente los SID modelados se pueden transformar a tres lenguajes de ejecución de SID diferentes. Aunque estos lenguajes han resultado una buena solución en términos de eficiencia, requieren de motores de ejecución específicos. Los nuevos lenguajes incluidos pueden ser lenguajes de programación generalistas como

Java y C que no requieran de un motor de ejecución específico. De esta manera, se puede mejorar la integración de los SID ejecutables en otras soluciones. También se pueden incluir lenguajes específicos para la representación de SID, tales como los utilizados en FuzzyTech, Xfuzzy 3.0 y FisPro. Esto facilitaría la interacción entre distintas herramientas con objetivos parecidos.

La **adición de otros modelos de sistemas de clasificación** consiste en permitir utilizar modelos de sistemas de clasificación que no sean SID. Dado que los modelos de DD son independientes de los conceptos de computación se podría añadir cualquier sistema de clasificación que permitiese la transformación desde los modelos de DD.

El **ofrecer una representación gráfica del conocimiento** consiste en mejorar la forma en la que se muestra el conocimiento contenido en las herramientas creadas. Otras representaciones, tales como mapas conceptuales o mapas mentales, pueden facilitar la asimilación y el procesamiento de la información contenida en los DD.

El **ofrecimiento de soporte para la gestión del conocimiento médico** consiste en ofrecer no sólo herramientas para el procesamiento del conocimiento de los DD, sino ir más allá y dar soporte al contexto del propio DD. Por ejemplo, se podría pensar en permitir generar pequeños protocolos de actuación cuyo elemento principal sea un DD. De esta forma, se permitiría la generación de protocolos ejecutables dirigidos por el resultado del DD.

## Publicaciones generadas

El trabajo realizado durante la presente tesis doctoral ha generado 19 publicaciones. Concretamente se han publicado 4 artículos en revistas indexadas en el *Journal Citation Reports*, se han realizado 5 participaciones en congresos internacionales y 6 participaciones en congresos nacionales y se han publicado 2 libros y 2 informes técnicos. 13 de las publicaciones han surgido a raíz del diseño y construcción de la plataforma Aide-GTP, de la que también son parte las herramientas para la definición de DD y SID presentadas en esta tesis.

En el *VIII Congreso Nacional de Errores Congénitos del Metabolismo* se presentó una versión inicial tanto del formalismo de representación de GCI utilizado en Aide-GTP como de la propia plataforma, en el ámbito de la GCI para el diagnóstico y tratamiento de la Hiperamonemia (Segundo et al., 2009). En esta publicación ya se describían unas versiones iniciales de los diagnósticos sobre esta enfermedad explicados en el capítulo 12. Cabe destacar que esta publicación recibió el premio a la mejor comunicación del congreso. Posteriormente, en el *III Simposio sobre Lógica Fuzzy y Soft Computing*, el cual se llevó a cabo en el marco del *III Congreso Español de Informática*, se presentó una versión preliminar de la representación de SID para ser integrados en GCI (Segundo et al., 2010). La versión final de la representación de SID, así como del proceso de creación automática utilizando MDDM se publicó en la revista *Applied Soft Computing* (índice JCR de 2,857) (Segundo et al., 2015b).

En el *12th Annual Meeting Health Technology Assessment international* se presentó la aplicación de escritorio DSScreening que utiliza un SID para ayudar en el proceso de cribado neonatal (Segundo et al., 2015a). En ese mismo congreso también se presentó el estudio realizado sobre la problemática de la informatización de los DD (López-Cuadrado et al., 2015). Una explicación más detallada de DSScreening se ha publicado en la revista *Expert Systems with Applications* (índice JCR de 2,981) (Segundo et al., 2017).

En las *XV Jornadas de Ingeniería del Software y Bases de Datos* dentro del *III Congreso Español de Informática* se presentó una primera versión de la representación de GCI Aide formalizada mediante un metamodelo, donde se utilizan los SID presentados en esta tesis (Buenestado et al., 2010). Una

versión muy similar al formalismo de representación actual se presentó en el marco del *XXIII International Conference of the European Federation for Medical Informatics* (Buenestado et al., 2011).

En las *VIII Jornadas de Aplicaciones y Transferencia Tecnológica de la Inteligencia Artificial*, dentro del *III Congreso Español de Informática*, se presentó una primera versión de la plataforma de ejecución de GCI Aide-GEP y del portal web Aide-Portal (Iruetaguena et al., 2010). Estas primeras versiones se comunicaban con un motor de ejecución que llevaba a cabo los cálculos de unos SID bastante primarios. En el *XXIII International Conference of the European Federation for Medical Informatics* se presentó la evolución de la plataforma de ejecución y más concretamente del portal (Barrena et al., 2011). En esta nueva versión la plataforma Aide-GEP ya se conectaba a un motor de ejecución de SID que utilizaba una versión casi definitiva de los SID. Posteriormente se realizó un estudio clínico sobre el nivel aceptación y de experiencia de usuario de Aide-Portal en el contexto de la GCI para el diagnóstico y tratamiento del asma pediátrica en la revista *Journal of Medical Systems* (índice JCR de 2,164) (Buenestado et al., 2013c). Esta guía incluía cálculos complejos desarrollados utilizando parte de las herramientas presentadas en esta tesis. Asimismo, también se realizó un estudio piloto para evaluar la mejoría clínica asistencial de pacientes pediátricos con asma (Korta et al., 2014).

En el contexto de la utilización de la plataforma Aide-GTP, se publicaron dos libros describiendo las funcionalidades de la plataforma desde el punto de vista del médico-usuario y del administrador y cómo utilizar dichas funcionalidades (Buenestado et al., 2013a) (Iruetaguena et al., 2013b). El contenido de estos libros, bajo un punto de vista más técnico, se publicó en dos informes técnicos del departamento de *Lenguajes y Sistemas Informáticos* de la UPV/EHU (Buenestado et al., 2013b) (Iruetaguena et al., 2013a).

En el *VII International Symposium on Distributed Computing and Artificial Intelligence* que tuvo lugar en el marco del *III Congreso Español de Informática* se presentó una primera versión del proceso de elaboración, implementación y actualización de GCI (donde también se incluye el proceso de informatización de DD) (Pikatzta et al., 2010). En el *III Congreso Nacional de Atención Sanitaria al Paciente Crónico* se presentó una versión mejorada y formalizada de ese proceso (Pikatzta et al., 2011). Finalmente, en la revista *Expert Systems with Applications* (índice JCR de 2,981) se publicó un método para la actualización automática de bibliografía (Iruetaguena et al., 2012). Este método era utilizable para facilitar la tarea de actualización tanto de GC como de GCI, así como para buscar nuevas versiones de DD.

Finalmente, el desarrollo de la tecnología de Aide-GTP ha dado como resultado 9 modelos de utilidad inscritos en el registro de la propiedad intelectual. Estos modelos son los referentes a distintas versiones de los componentes de Aide-GTP, así como a las GCI para el diagnóstico y tratamiento del asma y de la hiperamonemia. Durante la presente tesis también se ha participado en 16 proyectos de investigación con financiación pública y se ha firmado un

contrato con un empresa y un acuerdo de colaboración con el Ministerio de Sanidad, Servicios Sociales e Igualdad.



---

## Referencias

- AECOM (2017). *AECOM*. Asociación Española para el Estudio de los Errores Congénitos del Metabolismo. <http://ae3com.eu>.
- Agrawal, R., Imielinski, T., y Swami, A. (1993). Mining association rules between sets of items in large databases. En *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, páginas 207–216.
- AHRQ (2017). *National Guideline Clearinghouse*. Agency for Healthcare Research and Quality. <http://www.guideline.gov>.
- Al-Ebbini, L., Oztekin, A., y Chen, Y. (2016). Flas: Fuzzy lung allocation system for us-based transplantations. *European Journal of Operational Research*, 248:1051–1065.
- Alayón, S., Robertson, R., Warfield, S. K., y Ruiz-Alzola, J. (2007). A fuzzy system for helping medical diagnosis of malformations of cortical development. *Journal of Biomedical Informatics*, 40:221–235.
- Alonso, J. M. y Magdalena, L. (2011). Generating understandable and accurate fuzzy rule-based systems in a java environment. En *Lecture Notes in Artificial Intelligence - 9th International Workshop on Fuzzy Logic and Applications*.
- Amgoud, L. y Prade, H. (2009). Using arguments for making and explaining decisions. *Artificial Intelligence*, 173:413–436.
- Andrei, D., Poenaru, D. V., Nemes, D., Milicin, C., Vida, M., Gal, N., y Nadasan, E. (2015). Computer aided patient evaluation in the low back pain pathology. En *2015 IEEE 10th Jubilee International Symposium on Applied Computational Intelligence and Informatics (SACI)*, páginas 27–30.
- Ankishan, H. y Yilmaz, D. (2013). Comparison of svm and anfis for snore related sounds classification by using the largest lyapunov exponent and entropy. *Computational and Mathematical Methods in Medicine*, páginas 1–13.
- Arrieta, J., Zubizarreta, J. R., Pikatza, J. M., Larrañaga, P., Arrieta, A., y Campo, L. (1995). Gestión de un sistema de producción utilizando técnicas

- avanzadas de inteligencia artificial. En *Actas de las Jornadas de Aplicaciones y Transferencia Tecnológica de la Inteligencia Artificial*.
- ATL (2017). *ATL Transformation Language*. Eclipse Foundation. <https://eclipse.org/atl>.
- Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. En *Proceedings of the Second International Conference on Genetic Algorithms and their application*.
- Bárdossy, A., Blinowska, A., Kuzmich, W., Ollitrault, J., Lewandowski, M., Przybylski, A., y Jaworski, Z. (2014). Fuzzy logic-based diagnostic algorithm for implantable cardioverter defibrillators. *Artificial Intelligence in Medicine*, 60:113–121.
- Barrena, R., Pikatza, J. M., Iruetaguena, A., Segundo, U., Buenestado, D., Elorz, J., y Aldámiz-Echevarría, L. (2011). e-guidesmed: Support for translation of clinical guidelines. En *Proceedings of the XXIII International Conference of the European Federation for Medical Informatics*.
- Baturone, I., Ángel Barriga Barros, Jiménez-Fernández, C., López, D. R., y Sánchez-Solano, S. (2000). *Microelectronic Design of Fuzzy Logic-Based Systems*. CRC Press.
- Baum, L. y Petrie, T. (1966). Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, páginas 1554–1563.
- Beale, T. (2002). Archetypes: Constraint-based domain models for future information systems. En H. Kilov, K. B., editor, *Eleventh OOPS-LA Workshop on Behavioral Semantics: Serving the Customer*, páginas 16–32.
- Beale, T. y Heard, S. (2008). Openehr architecture review. Informe técnico, OpenEHR.
- Beloufa, F. y Chikh, M. A. (2013). Design of fuzzy classifier for diabetes disease using modified artificial bee colony algorithm. *Computer Methods and Programs in Biomedicine*, 112:92–103.
- Bezdek, J., Dubois, D., y Prade, H. (1999). *Fuzzy Sets in Approximate Reasoning and Information Systems*. Springer US.
- Bhatia, A., Mago, V. K., y Singh, R. (2014). Use of soft computing techniques in medical decision making: A survey. En *International Conference on Advances in Computing, Communications and Informatics (ICACCI, 2014)*, páginas 1131–1137.
- Blickle, T. y Thiele, L. (1995). A comparison of selection schemes used in genetic algorithms. Informe técnico, Zurich: Swiss Federal Institute of Technology (ETH) Zurich, Computer Engineering and Communications Networks Lab (TIK).
- Boxwala, A. A., Peleg, M., Tu, S., Ogunyemi, O., Zeng, Q. T., Wang, D., Patel, V. L., Greenes, R. A., y Shortliffe, E. H. (2004). Glif3: a representation format for sharable computer-interpretable clinical practice guidelines. *Journal of Biomedical Informatics*, 37 (3):147–161.
- Buenestado, D. (2014). *Representación y ejecución de guías clínicas informatizadas independientes de plataforma utilizando métodos de desarrollo*

- dirigido por modelos*. Tesis doctoral, Universidad del País Vasco / Euskal Herriko Unibertsitatea.
- Buenestado, D., Barrena, R., y Pérez, T. A. (2013a). *Aide Portal paso a paso. El portal de consulta de guías clínicas informatizadas para médicos*. Editorial Académica Española.
- Buenestado, D., Barrena, R., Segundo, U., y Iruetaguena, A. (2013b). Manual de usuario del portal de guías clínicas informatizadas aide portal. Informe técnico, Facultad de Informática, Universidad del País Vasco / Euskal Herriko Unibertsitatea.
- Buenestado, D., Elorz, J., Pérez-Yarza, E. G., Iruetaguena, A., Segundo, U., Barrena, R., y Pikatza, J. M. (2013c). Evaluating acceptance and user experience of a guideline-based clinical decision support system execution platform. *Journal of Medical Systems*, 37 (2):1–9.
- Buenestado, D., Pikatza, J. M., Segundo, U., Iruetaguena, A., Barrena, R., y Aldámiz-Echevarría, L. (2011). Two-level authoring of computer-interpretable guidelines. En *Proceedings of the XXIII International Conference of the European Federation for Medical Informatics*.
- Buenestado, D., Pikatza, J. M., Segundo, U., Iruetaguena, A., Barrena, R., García, J. J., Aldámiz-Echevarría, L., y Sanjurjo, P. (2010). Representación mediante arquetipos y generación dirigida por modelo de guías clínicas ejecutables. En *Actas de las XV Jornadas de Ingeniería del Software y Bases de Datos*.
- Büyükcavcu, A., Albayrak, Y. E., y Göker, N. (2016). A fuzzy information-based approach for breast cancer risk factors assessment. *Applied Soft Computing*, 38:437–452.
- Cabrera, I. P., Cordero, P., y Ojeda-Aciego, M. (2009). Fuzzy logic, soft computing, and applications. En *IWANN '09 Proceedings of the 10th International Work-Conference on Artificial Neural Networks: Part I: Bio-Inspired Systems: Computational and Ambient Intelligence*, páginas 236–244.
- Castanho, M. J. P., Fernandes, F., Ré, A. M. D., Rautenberg, S., y Billis, A. (2013). Fuzzy expert system for predicting pathological stage of prostate cancer. *Expert Systems with Applications*, 40:466–470.
- Castillo, O. y Melin, P. (2008). *Type-2 Fuzzy Logic: Theory and Applications*, capítulo Type-2 Fuzzy Logic, páginas 29–43. Springer Berlin Heidelberg.
- Chourasia, V. S., Tiwari, A. K., y Gangopadhyay, R. (2014). Interval type-2 fuzzy logic based antenatal care system using phonocardiography. *Applied Soft Computing*, 14:489–497.
- Ciccarese, P., Caffi, E., Quaglini, S., y Stefanelli, M. (2005). Architectures and tools for innovative health information systems: The guide project. *International Journal of Medical Informatics*, 74:553–562.
- Clark, T., Sammut, P., y Willans, J. (2008). *Applied metamodelling: A foundation for language driven development (Second edition)*. Ceteva.
- Clercq, P. D., Kaiser, K., y Hasman, A. (2008). *Computer-based Medical Guidelines and Protocols: A Primer and Current Trends*, capítulo Computer-interpretable Guideline Formalisms, páginas 22–43. IOS Press.

- Cosenza, B. (2012). Off-line control of the postprandial glycemia in type 1 diabetes patients by a fuzzy logic decision support. *Expert Systems with Applications*, 39 (12):10693–10699.
- COSSAC (2017). *Tallis*. COSSAC. <https://cossac.org/tallis.html>.
- Couce, M. L., Bustos, G., Lázaro, M., Martínez-Pardo, M., Molina, A., de Pipaón, M. S., Sanjurjo, P., Serrano, M. L., García-Alix, A., Morais, A., Martín, E., Lama, R., García-Silva, M. T., Pedrón, C., Canals, M. J., y Quintana, A. B. (2009). *Protocolo Hispano-Luso de diagnóstico y tratamiento de las hiperamonemias en pacientes neonatos y de más de 30 días de vida*. Ergon.
- Crockett, K. A., Bandar, Z., Fowdar, J., y O’Shea, J. (2006). Genetic tuning of fuzzy inference within fuzzy classifier systems. *Expert Systems*, 23 (2):63–82.
- Czabanski, R., Jezewski, J., Horoba, K., y Jezewski, M. (2013). Fetal state assessment using fuzzy analysis of fetal heart rate signals - agreement with the neonatal outcome. *Biocybernetics and biomedical engineering*, 33:145–155.
- Czabanski, R., Jezewski, J., Matonia, A., y Jezewski, M. (2012). Computerized analysis of fetal heart rate signals as the predictor of neonatal acidemia. *Expert Systems with Applications*, 39:11846–11860.
- de Brito, M. J. A., Nahas, F. X., Ortega, N. R. S., Cordás, T. A., Dini, G. M., Nieto, M. S., y Ferreira, L. M. (2013). Support system for decision making in the identification of risk for body dysmorphic disorder: A fuzzy model. *International Journal of Medical Informatics*, 82:844–853.
- de Bruin, J. S., Adlassnig, K.-P., Blacky, A., y Koller, W. (2016). Detecting borderline infection in an automated monitoring system for healthcare-associated infection using fuzzy logic. *Artificial Intelligence in Medicine*, 69:33–41.
- Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38:325–339.
- Denekamp, Y. y Peleg, M. (2010). Timeddx - a multi-phase anchor-based diagnostic decision-support model. *Journal of Biomedical Informatics*, 43 (1):111–124.
- Dennis, B. y Muthukrishnan, S. (2014). Agfs: Adaptive genetic fuzzy system for medical data classification. *Applied Soft Computing*, 25:242–252.
- Derrac, J., García, S., y Herrera, F. (2014). Fuzzy nearest neighbor algorithms: Taxonomy, experimental analysis and prospects. *Information Sciences*, páginas 98–119.
- DMG (2017). *Predictive Model Markup Language (PMML)*. Data Mining Group. <http://www.dmg.org/>.
- Doctor, F., Syue, C.-H., Liu, Y.-X., Shieh, J.-S., y Iqbal, R. (2016). Type-2 fuzzy sets applied to multivariable self-organizing fuzzy logic controllers for regulating anesthesia. *Applied Soft Computing*, 38:872–889.
- Dubois, D. y Prade, H. (1980). *Fuzzy Sets: Theory and Applications to Policy Analysis and Information Systems*, capítulo New Results about Properties

- and Semantics of Fuzzy Set-Theoretic Operators, páginas 59–75. Springer US.
- Efe, M. O. (2009). A comparison of anfis, mlp and svm in identification of chemical processes. En *IEEE Control Applications & Intelligent Control*, páginas 689–694.
- Elsevier (2017). *ScienceDirect*. Elsevier. <http://www.sciencedirect.com/>.
- Esfandiari, N., Babavalian, M., Moghadam, A., y Tabar, V. (2014). Knowledge discovery in medicine: Current issue and future trend. *Expert Systems with Applications*, páginas 4434–4463.
- Eshelman, L. J. y Schaffer, J. D. (1993). Real-coded genetic algorithms and interval-schemata. En *Proceedings of the Second Workshop on Foundations of Genetic Algorithms*.
- Espósito, M., Falco, I. D., y Pietro, G. D. (2011). An evolutionary-fuzzy dss for assessing health status in multiple sclerosis disease. *International Journal of Medical Informatics*, 80:e245–e254.
- EUSFLAT (2017). *European Society for Fuzzy Logic and Technology*. European Society for Fuzzy Logic and Technology. <http://www.eusflat.org/index.php>.
- Fialho, A. S., Vieira, S. M., Kaymak, U., Almeida, R. J., Cismondi, F., Reti, S. R., Finkelstein, S., y Sousa, J. M. C. (2016). Mortality prediction of septic shock patients using probabilistic fuzzy systems. *Applied Soft Computing*, 42:194–203.
- Field, M. J. y Lohr, K. N. (1990). *Guidelines for Clinical Practice: From Development to Use*. National Academy Press.
- Flórez, S., Pikatza, J. M., Larburu, I., y Sobrado, F. (2005). Solución basada en un lenguaje de representación y un motor de ejecución de guías de práctica clínica para la ayuda a la decisión en atención primaria y urgencias. En *Actas de la XI Conferencia de la Asociación Española para la Inteligencia Artificial*, páginas 39–48.
- Fox, J., Alabassi, A., Patkar, V., Rose, T., y Black, E. (2006). An ontological approach to modelling tasks and goals. *Computers in Biology and Medicine*, 36:837–856.
- Friedman-Hill, E. (2017). *JESS, the Rule Engine for Java Platform*. Sandia National Laboratories. <http://www.jessrules.com/jess/index.shtml>.
- Frye, D., Zelazo, P., y Palfai, T. (1995). Theory of mind and rule-based reasoning. *Cognitive Development*, páginas 483–527.
- Gacto, M. J., Alcalá, R., y Herrera, F. (2011). Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. *Information Sciences*, 181:4340–4360.
- García, J. J. y Calvo, R. (2006). Mining text with pimientó. *IEEE Internet Computing*, 10 (4):27–35.
- García, J. J. y Pikatza, J. M. (2006). Web misuse detection through text categorisation of application. *International Journal on Artificial Intelligence Tools*, 15 (5):849–854.

- Garibaldi, J. M., Zhou, S.-M., Wang, X.-Y., John, R. I., y Ellis, I. O. (2012). Incorporation of expert variability into breast cancer treatment recommendation in designing clinical protocol guided fuzzy rule system models. *Journal of Biomedical Informatics*, 45 (3):447–459.
- Gharaviri, A., Dehghan, F., Teshnelab, M., y Moghaddam, H. (2008). Comparison of neural network, anfis, and svm classifiers for pvc arrhythmia detection. En *International Conference on Machine Learning and Cybernetics*, páginas 750–755.
- Ghasemi, J., Ghaderi, R., Mollaei, M. R. K., y Hojjatoleslami, S. A. (2013). A novel fuzzy dempster-shafer inference system for brain mri segmentation. *Information Sciences*, 223:205–220.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley.
- Goldberg, D. E. y Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. En *Foundations of Genetic Algorithms*.
- Gooch, P. y Roudsari, A. (2011). Computerization of workflows, guidelines, and care pathways: a review of implementation challenges for process-oriented health information systems. *Journal of the American Medical Informatics Association*, 18 (6):738–748.
- Gozalczany, M. B. y Rudzinski, F. (2017). Interpretable and accurate medical data classification - a multi-objective genetic-fuzzy optimization approach. *Expert Systems with Applications*, 71:26–39.
- Grabisch, M., Marichal, J.-L., Mesiar, R., y Pap, E. (2009). *Aggregation Functions (Encyclopedia of Mathematics and its Applications) 1st Edition*. Cambridge University Press.
- Graham, R., Mancher, M., Wolman, D. M., Greenfield, S., y Steinberg, E., editores (2011). *Clinical Practice Guidelines we can trust*. National Academy Press.
- Greenhalgh, T. (2010). *How to Read a Paper: The Basics of Evidence-Based Medicine (4th revised edition)*. John Wiley & Sons.
- Gronback, R. C. (2009). *Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit*. Addison Wesley.
- GuiaSalud (2017). *GuiaSalud*. Biblioteca de Guías de Práctica Clínica del Sistema Nacional de Salud. <http://www.guiasalud.es>.
- Guillaume, S. y Charnomordic, B. (2011). Learning interpretable fuzzy inference systems with fispro. *Information Sciences*, 181:4409–4427.
- Herrera, F. (2008). Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evolutionary Intelligence*, 1 (1):27–46.
- Herrera, F., Lozano, M., y Sánchez, A. M. (2003). A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *International Journal of Intelligent Systems*, 18:309–338.
- Herrera, F., Lozano, M., y Verdegay, J. L. (1995). Tuning fuzzy logic controllers by genetic algorithms. *International Journal of Approximate Reasoning*, 12 (3-4):299–315.

- Herrera, F., Lozano, M., y Verdegay, J. L. (1996). Dynamic and heuristic fuzzy connectives based crossover operators for controlling the diversity and convergence of real-coded genetic algorithms. *International Journal of Intelligent Systems*, 11:21.
- Herrera, F., Lozano, M., y Verdegay, J. L. (1997). Fuzzy connectives based crossover operators to model genetic algorithms population diversity. *Fuzzy Sets and Systems*, 92 (1):21–30.
- Hinterding, R. (1995). Gaussian mutation and self-adaption for numeric genetic algorithms. En *IEEE International Conference on Evolutionary Computation*.
- HL7 (2017). *Health Level 7 International*. Health Level 7 International. <http://www.hl7.org/>.
- Holland, J. H. (1992). *Adaptation In Natural And Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence (New Edition)*. MIT Press.
- Howard, R. A. y Matheson, J. E. (2005). Influence diagrams. *Decision Analysis*, 2(3):127–143.
- Hutchinson, J., Rouncefield, M., y Whittle, J. (2011). Model-driven engineering practices in industry. En *Proceedings of the 33rd International Conference on Software Engineering*.
- IBM (2017). *IBM Rational Software*. IBM. <http://www-01.ibm.com/software/es/rational/>.
- IEC (1997). *IEC 1131 - Programmable Controllers Part 7 - Fuzzy Control Programming*. International Electrotechnical Commission (Technical committee No. 65: Industrial process measurement and control).
- IEEE (2017). *IEEE Xplore*. Institute of Electrical and Electronics Engineers. <http://ieeexplore.ieee.org>.
- InferMed (2017). *Arezzo*. InferMed Ltd. <http://www.infermed.com/en/Clinical-Decision-Support/Overview.aspx>.
- INFORM (2017). *FuzzyTech*. INFORM GmbH. <http://www.fuzzytech.com/>.
- Inserm (2017). *Orphanet*. <http://www.orpha.net/>.
- Iruetaguena, A., García, J. J., Pikatza, J. M., Segundo, U., Buenestado, D., y Barrena, R. (2012). Automatic retrieval of current evidence to support update of bibliography in clinical guidelines. *Expert Systems with Applications*, 40:2081–2091.
- Iruetaguena, A., Pikatza, J. M., Segundo, U., Buenestado, D., Barrena, R., García, J. J., Aldámiz-Echevarría, L., Sanjurjo, P., y Elorz, J. (2010). e-guidesmed: Guías médicas ejecutables e información complementaria vía web para profesionales sanitarios. En *Actas de las VIII Jornadas de Aplicaciones y Transferencia Tecnológica de la Inteligencia Artificial*.
- Iruetaguena, A., Segundo, U., Barrena, R., y Buenestado, D. (2013a). Administración y gestión del portal de guías clínicas informatizadas aide portal. Informe técnico, Facultad de Informática, Universidad del País Vasco / Euskal Herriko Unibertsitatea.

- Iruetaguena, A., Segundo, U., y Pikatza, J. M. (2013b). *Administración del portal de Guías Clínicas Informatizadas Aide Portal: Guía para su implantación y soporte técnico*. Editorial Académica Española.
- Jang, J.-S. R. (1993). Anfis: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3):665–685.
- Jang, J.-S. R., Sun, C.-T., y Mizutani, E. (1997). *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*. Prentice Hall.
- Karlik, B. (2013). Soft computing methods in bioinformatics: a comprehensive review. *Mathematical and Computational Applications*, páginas 176–197.
- Karr, C. (1991). Genetic algorithms for fuzzy controllers. *AI Expert*, 6 (2):26–33.
- Kazemzadeh, R. S. y Sartipi, K. (2006). Incorporating data mining applications into clinical guidelines. En *IEEE International Symposium on Computer-Based Medical Systems*, páginas 321–328.
- Kelly, S. y Tolvanen, J.-P. (2008). *Domain-Specific Modeling: Enabling Full Code Generation*. John Wiley & Sons.
- Khatibi, V. y Montazer, G. A. (2010). A fuzzy-evidential hybrid inference engine for coronary heart disease risk assessment. *Expert Systems with Applications*, 37:8536–8542.
- Kim, D., Choi, Y.-S., y Lee, S.-Y. (2002). An accurate cog defuzzifier design using lamarkian co-adaptation of learning and evolution. *Fuzzy Sets and Systems*, 130 (2):207–225.
- Klement, E. P., Mesiar, R., y Pap, E. (2000). *Triangular Norms*. Trends in Logic. Springer Netherlands.
- Korta, J., Iruetaguena, A., Pérez-Yarza, E. G., Elorz, J., Segundo, U., y Pikatza, J. M. (2014). Evaluación de la mejoría clínica asistencial del niño asmático usando guías clínicas informatizadas. estudio piloto. En *XXXVI Reunión de la Sociedad Española de Neumología Pediátrica*, Málaga.
- Kosko, B. (1986). Fuzzy cognitive maps. *International Journal of Man-Machine Studies*, 24:65–75.
- Kuncheva, L. I. (2000). *Fuzzy Classifier Design*. Physica-Verlag.
- Lai, J. C. Y., Leung, F. H. F., y Ling, S. H. (2014). Hypoglycaemia detection using fuzzy inference system with intelligent optimiser. *Applied Soft Computing*, 20:54–65.
- Lai, J. C. Y., Leung, F. H. F., Ling, S. H., y Nguyen, H. T. (2013). Hypoglycaemia detection using fuzzy inference system with multi-objective double wavelet mutation differential evolution. *Applied Soft Computing*, 13:2803–2811.
- Larburu, I., Pikatza, J. M., Sobrado, F., García, J. J., y de Ipiña, D. L. (2003). Ayuda a la decisión en el proceso de desarrollo de software: Fase de elaboración de proyectos. En *Actas de la X Conferencia de la Asociación Española para la Inteligencia Artificial*.

- Latoszek-Berendsen, A., Tange, H., van den Herik, H. J., y Hasman, A. (2010). From clinical practice guidelines to computer-interpretable guidelines: A literature overview. *Methods of Information in Medicine*, 49 (6):550–570.
- Lima, L., Novais, P., Costa, R., Bulas-Cruz, J., y Neves, J. (2010). Decision making based on quality-of-information a clinical guideline for chronic obstructive pulmonary disease scenario. En *Improceedings of the 7th International Symposium in Distributed Computing and Artificial Intelligence*.
- López-Cuadrado, J., Armendáriz, A. J., Presedo, C., Segundo, U., Barrena, R., Korta, J., y Pérez, T. A. (2015). Why tables on clinical practice guidelines are not easily computerizable. En *Proceedings of the 12th Annual Meeting Health Technology Assessment international (HTAi 2015)*.
- Lozano, M., Herrera, F., Krasnogor, N., y Molina, D. (2004). Real-coded memetic algorithms with crossover hill-climbing. *Evolutionary Computation*, 12 (3):273–302.
- Lucas, P. J. F., van der Gaag, L., y Abu-Hanna, A. (2004). Bayesian networks in biomedicine and health-care. *Artificial Intelligence in Medicine*, páginas 201–204.
- Lund, A. M., Hougaard, D. M., Simonsen, H., Andresen, B. S., Christensen, M., Duno, M., Skogstrand, K., Olsen, R. K. J., Jensen, U. G., Cohen, A., Larsen, N., Saugmann-Jensen, P., Gregersen, N., Brandt, N. J., Christensen, E., Skovby, F., y Norgaard-Pedersen, B. (2012). Biochemical screening of 504,049 newborns in denmark, the faroe islands and greenland - experience and development of a routine program for expanded newborn screening. *Molecular Genetics and Metabolism*, 107 (3):281–293.
- Mago, V. K., Bhatia, N., Bhatia, A., y Mago, A. (2012). Clinical decision support system for dental treatment. *Journal of Computational Science*, 3:254–261.
- Mahmoodabadi, S. Z., Alirezaie, J., Babyn, P., Kassner, A., y Widjaja, E. (2010). Wavelets and fuzzy relational classifiers: A novel spectroscopy analysis system for pediatric metabolic brain diseases. *Fuzzy Sets and Systems*, 161 (1):75–95.
- Mamdani, E. H. y Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7 (1):1–13.
- Marateb, H. R., Mansourian, M., Faghihimani, E., Amini, M., y Farina, D. (2014). A hybrid intelligent system for diagnosing microalbuminuria in type 2 diabetes patients without having to measure urinary albumin. *Computers in Biology and Medicine*, 45:34–42.
- Maron, M. y Kuhns, J. (1960). On relevance, probabilistic indexing, and information retrieval. *Journal of the Association for Computing Machinery*.
- Mathworks (2017a). *Matlab*. Mathworks. <http://es.mathworks.com/products/matlab/>.
- Mathworks (2017b). *Matlab Fuzzy Logic Toolbox*. Mathworks. <http://es.mathworks.com/products/fuzzy-logic/>.

McHugh, D. M. S., Cameron, C. A., Abdenur, J. E., Abdulrahman, M., Adair, O., Al Nuaimi, S. A., Ahlman, H., Allen, J. J., Antonozzi, I., Archer, S., Au, S., Auray-Blais, C., Baker, M., Bamforth, F., Beckmann, K., Pino, G. B., Berberich, S. L., Binard, R., Boemer, F., Bonham, J., Breen, N. N., Bryant, S. C., Caggana, M., Caldwell, S. G., Camilot, M., Campbell, C., Carducci, C., Cariappa, R., Carlisle, C., Caruso, U., Cassanello, M., Castilla, A. M., Ramos, D. E. C., Chakraborty, P., Chandrasekar, R., Ramos, A. C., Cheillan, D., Chien, Y.-H., Childs, T. A., Chrastina, P., Sica, Y. C., de Juan, J. A. C., Colandre, M. E., Espinoza, V. C., Corso, G., Currier, R., Cyr, D., Czuczy, N., D'Apolito, O., Davis, T., de Sain-Van der Velden, M. G., Pecellin, C. D., Di Gangi, I. M., Di Stefano, C. M., Dotsikas, Y., Downing, M., Downs, S. M., Dy, B., Dymerski, M., Rueda, I., Elvers, B., Eaton, R., Eckerd, B. M., El Mougy, F., Eroh, S., Espada, M., Evans, C., Fawbush, S., Fijolek, K. F., Fisher, L., Franzson, L., Frazier, D. M., Garcia, L. R. C., Bermejo, M. S. G.-V., Gavrilov, D., Gerace, R., Giordano, G., Irazabal, Y. G., Greed, L. C., Grier, R., Grycki, E., Gu, X., Gulamali-Majid, F., Hagar, A. F., Han, L., Hannon, W. H., Haslip, C., Hassan, F. A., He, M., Hietala, A., Himstedt, L., Hoffman, G. L., Hoffman, W., Hoggatt, P., Hopkins, P. V., Hougaard, D. M., Hughes, K., Hunt, P. R., Hwu, W.-L., Hynes, J., Ibarra-Gonzalez, I., Ingham, C. A., Ivanova, M., Jacox, W. B., John, C., Johnson, J. P., Jonsson, J. j., Karg, E., Kasper, D., Klopper, B., Katakouzinis, D., Khneisser, I., Knoll, D., Kobayashi, H., Koneski, R., Kozich, V., Kouapei, R., Kohlmuller, D., Kremensky, I., la Marca, G., Lavochkin, M., Lee, S.-Y., Lehotay, D. C., Lemes, A., Lepage, J., Lesko, B., Lewis, B., Lim, C., Linard, S., Lindner, M., Lloyd-Puryear, M. A., Lorey, F., Loukas, Y. L., Luedtke, J., Maffitt, N., Magee, J. F., Manning, A., Manos, S., Marie, S., Hadachi, S. M., Marquardt, G., Martin, S. J., Matern, D., Gibson, S. K. M., Mayne, P., McCallister, T. D., McCann, M., McClure, J., McGill, J. J., McKeever, C. D., McNeilly, B., Morrissey, M. A., Moutsatsou, P., Mulcahy, E. A., Nikoloudis, D., Norgaard-Pedersen, B., Oglesbee, D., Oltarzewski, M., Ombrone, D., Ojodu, J., Papakonstantinou, V., Reoyo, S. P., Park, H.-D., Pasquali, M., Pasquini, E., Patel, P., Pass, K. A., Peterson, C., Pettersen, R. D., Pitt, J. J., Poh, S., Pollak, A., Porter, C., Poston, P. A., Price, R. W., Queijo, C., Quesada, J., Randell, E., Ranieri, E., Raymond, K., Reddic, J. E., Reuben, A., Ricciardi, C., Rinaldo, P., Rivera, J. D., Roberts, A., Rocha, H., Roche, G., Greenberg, C. R., Mellado, J. M. E., Juan-Fita, M. J., Ruiz, C., Ruopolo, M., Rutledge, S. L., Ryu, E., Saban, C., Sahai, I., Garcia-Blanco, M. I. S., Santiago-Borrero, P., Schenone, A., Schoos, R., Schweitzer, B., Scott, P., Seashore, M. R., Seeterlin, M. A., Sesser, D. E., Sevier, D. W., Shone, S. M., Sinclair, G., Skriniska, V. A., Stanley, E. L., Strovel, E. T., Jones, A. L. S., Sunny, S., Takats, Z., Tanyalcin, T., Teofoli, F., Thompson, J. R., Tomashitis, K., Domingos, M. T., Torres, J., Torres, R., Tortorelli, S., Turi, S., Turner, K., Tzanakos, N., Valiente, A. G., Vallance, H., Vela-Amieva, M., Vilarinho, L., von Dobein, U., Vincent, M.-F., Vorster, B. C., Watson, M. S., Webster, D., Weiss, S., Wilcken, B., Wiley, V., Williams, S. K., Wi-

- llis, S. A., Woontner, M., Wright, K., Yahyaoui, R., Yamaguchi, S., Yssel, M., y Zakowicz, W. M. (2011). Clinical validation of cutoff target ranges in newborn screening of metabolic disorders by tandem mass spectrometry: A worldwide collaborative project. *Genetics in Medicine*, 13 (3)(3):230–254.
- Mens, T. y Gorp, P. V. (2005). A taxonomy of model transformation. En *Proceedings of Dagstuhl Seminar on Language Engineering for Model-Driven Software Development*.
- MetaCase (2017). *MetaEdit* +. MetaCase. <https://www.metacase.com/products.html>.
- Meza-Palacios, R., Aguilar-Lasserre, A. A., Ureña-Bogarín, E. L., Vázquez-Rodríguez, C. F., Posada-Gómez, R., y Trujillo-Mata, A. (2017). Development of a fuzzy expert system for the nephropathy control assessment in patients with type 2 diabetes mellitus. *Expert Systems with Applications*, 72:335–343.
- Michalewicz, Z. (1998). *Genetic Algorithms + Data Structures = Evolution Programs (3rd revision)*. Springer Verlag.
- Microsoft (2017). *Microsoft DSL Tools*. Microsoft. <https://msdn.microsoft.com/en-us/library/bb126327.aspx>.
- Miranda, G. H. B. y Felipe, J. C. (2015). Computer-aided diagnosis system based on fuzzy logic for breast cancer categorization. *Computers in Biology and Medicine*, 64:334–346.
- Moreno, F. J. (2002). *Un entorno de desarrollo para sistemas de inferencia complejos basados en lógica difusa*. Tesis doctoral, Universidad de Sevilla.
- Morris, A. H. (2000). Developing and implementing computerized protocols for standardization of clinical decisions. *Annals of Internal Medicine*, 132 (5):373–383.
- Muthukaruppan, S. y Er, M. J. (2012). A hybrid particle swarm optimization based fuzzy expert system for the diagnosis of coronary artery disease. *Expert Systems with Applications*, 39 (14):11657–11665.
- Nguyen, T., Khosravi, A., Creighton, D., y Nahavandi, S. (2015). Medical data classification using interval type-2 fuzzy logic system and wavelets. *Applied Soft Computing*, 30:812–822.
- NICE (2017). *National Institute for Health and Clinical Excellence*. National Institute for Health and Clinical Excellence. <http://www.nice.org.uk/>.
- Novák, V. (2005). *Computational Intelligence, Theory and Applications. Advances in Soft Computing, vol 33*, capítulo Perception-Based Logical Deduction, páginas 237–250. Springer Berlin Heidelberg.
- Novák, V. y Perfilieva, I. (2004). On the semantics of perception-based fuzzy logic deduction. *International Journal of Intelligent Systems*, 19(11):1007–1031.
- NRCC (2015). *FuzzyCLIPS: Fuzzy logic extension of the CLIPS expert system shell*. National Research Council of Canada. <http://archive.nrc-cnrc.gc.ca/eng/ibp/iit/pastprojects/fuzzyreasoning.html>.
- Nuryani, N., Harjito, B., Yahya, I., Lestari, A., Anzihory, E., y Farosi, K. (2015). Atrial fibrillation detection using swarm fuzzy inference system and

- electrocardiographic p-wave features. *Procedia Computer Science*, 72:154–161.
- Obeo (2017). *Acceleo*. Eclipse Foundation. <http://www.eclipse.org/acceleo/>.
- Oluwagbemi, O., Oluwagbemi, F., y Abimbola, O. (2016). Ebinformatics: Ebola fuzzy informatics systems on the diagnosis, prediction and recommendation of appropriate treatments for ebola virus disease (evd). *Informatics in Medicine Unlocked*, 2:12–37.
- OMG (2017a). *Meta-Object Facility*. Object Management Group. <http://www.omg.org/mof/>.
- OMG (2017b). *Object Management Group Model Driven Architecture*. Object Management Group. <http://omg.org/mda>.
- OMG (2017c). *XML Metadata Interchange (XMI)*. Object Management Group. <http://www.omg.org/spec/XMI/>.
- OMIM (2017). *Online Mendelian Inheritance in Man*. Johns Hopkins University. <http://www.omim.org/>.
- Orchard, R. (2001). Fuzzy reasoning in jess: The fuzzyj toolkit and fuzzyjess. En *3rd International Conference on Enterprise Information Systems*.
- Orphanou, K., Stassopoulou, A., y Keravnou, E. (2014). Temporal abstraction and temporal bayesian networks in clinical domains: A survey. *Artificial Intelligence in Medicine*, 60:133–149.
- Pacuit, E. (2012). Voting methods. En Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*.
- Pal, D., Mandana, K. M., Pal, S., Sarkar, D., y Chakraborty, C. (2012). Fuzzy expert system approach for coronary artery disease screening using clinical parameters. *Knowledge-Based Systems*, 36:162–174.
- Pandey, B. y Mishra, R. B. (2009). Knowledge and intelligent computing system in medicine. *Computers in Biology and Medicine*, 39(3):215–230.
- Papageorgiou, E. I. y Salmerón, J. (2013). A review of fuzzy cognitive maps research during the last decade. *IEEE Transactions on Fuzzy Systems*, páginas 66–79.
- Pawade, D. Y., Diwase, T. S., y Pawade, T. R. (2013). Designing and implementation of fuzzy logic based automatic system to estimate dose of anesthesia. En *Confluence 2013: The Next Generation Information Technology Summit (4th International Conference)*, páginas 95–102.
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, San Francisco.
- Pedrycz, W. y Gomide, F. (1998). *An Introduction to Fuzzy Sets: Analysis and Design*. MIT Press.
- Peleg, M., Tu, S., Bury, J., Ciccicarese, P., Fox, J., Greenes, R. A., Hall, R., Johnson, P. D., Jones, N., Kumar, A., Miksch, S., Quaglini, S., Seyfang, A., Shortliffe, E. H., y Stefanelli, M. (2003). Comparing computer-interpretable guideline models: a case-study approach. *Journal of the American Medical Informatics Association*, 10 (1):52–68.
- Pikatzka, J. M. (1992). *MAMUR: Mugapenen betetzean oinarritutako tresna modular multmaila*. Tesis doctoral, Universidad de Deusto, Bilbao.

- Pikatz, J. M., Aldámiz-Echevarría, L., García, J. J., Remiro, M., Alustiza, E., Blarduni, E., Vela, A., y Martul, P. (1998). Diabetes-I: A decision support system for the attendance in the treatment of the diabetes mellitus type I. *Diabetes, Nutrition & Metabolism: Clinical and Experimental*, 11 (1):72–78.
- Pikatz, J. M., Buenestado, D., Segundo, U., Elorz, J., Pérez-Yarza, E. G., y Aldámiz-Echevarría, L. (2011). Traslación de guías clínicas y gestión del conocimiento. En *Actas del III Congreso Nacional de Atención Sanitaria al Paciente Crónico*.
- Pikatz, J. M., Iruetaguena, A., Buenestado, D., Segundo, U., García, J. J., Aldámiz-Echevarría, L., Elorz, J., Barrena, R., y Sanjurjo, P. (2010). Towards an effective knowledge translation of clinical guidelines and complementary information. En *Advances in Intelligent and Soft-Computing series of Springer Verlag*.
- Pikatz, J. M., Sobrado, F., y García, J. J. (2001). Melsis: un marco de trabajo para la construcción de sistemas de ayuda a la toma de decisiones en problemas de monitorización. En *Workshop Apoyo a la Decisión en Ingeniería del Software - Decision Support in Software Engineering*.
- Prakken, H. (2009). An abstract framework for argumentation with structured arguments. Informe técnico, Department of Information and Computing Sciences, Utrecht University.
- Protégé (2017). *Protégé*. Stanford Center for Biomedical Informatics Research at the Stanford University School of Medicine. <http://protege.stanford.edu/>.
- Quaglini, S., Dazzi, L., Gatti, L., Stefanelli, M., Fassino, C., y Tondini, C. (1998). Supporting tools for guideline development and dissemination. *Artificial Intelligence in Medicine*, 14:119–137.
- Rabbi, A. F., Aarabi, A., y Fazel-Rezai, R. (2010). Fuzzy rule-based seizure prediction based on correlation dimension changes in intracranial eeg. En *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, páginas 3301–3304.
- Reilly, E. D. (2003). *Milestones in computer science and information technology*. Greenwood Publishing Group.
- Rico, O. C. (2013). Tratamiento del asma. *Revista Pediátrica de Atención Primaria*, 22:97–103.
- Rokach, L. y Maimon, O. (2007). *Data Mining with Decision Trees: Theory and Applications*. World Scientific.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, páginas 386–408.
- Rosenfeld, R. M. y Shiffman, R. N. (2009). Clinical practice guideline development manual: a quality-driven approach for translating evidence into action. *Otolaryngology – head and neck surgery*, 140 (6 suppl 1):1–43.
- Saeed, F., Salim, N., y Abdo, A. (2012). Voting-based consensus clustering for combining multiple clusterings of chemical structures. *Journal of Cheminformatics*, 4(37).

- Salgado, C. M., Vieira, S. M., Mendonça, L. F., Finkelstein, S., y Sousa, J. M. C. (2016). Ensemble fuzzy models in personalized medicine: Application to vasopressors administration. *Engineering Applications of Artificial Intelligence*, 49:141–148.
- Samuel, O. W., Omisore, M. O., y Ojokoh, B. A. (2013). A web based decision support system driven by fuzzy logic for the diagnosis of typhoid fever. *Expert Systems with Applications*, 40(10):4164–4171.
- Sanz, J. A., Galar, M., Jurio, A., Brugos, A., Pagola, M., y Bustince, H. (2014). Medical diagnosis of cardiovascular diseases using an interval-valued fuzzy rule-based classification system. *Applied Soft Computing*, 20:103–111.
- Savely, R., Culbert, C., Riley, G., Dantes, B., Ly, B., Ortiz, C., Giarratano, J., y López, F. (2017). *CLIPS: A tool for building expert systems*. <http://clipsrules.sourceforge.net/>.
- Seashore, M. R. (2009). *GeneReviews*, capítulo The Organic Acidemias: An Overview. University of Washington.
- Segundo, U., Aldámiz-Echevarría, L., López-Cuadrado, J., Buenestado, D., Andrade, F., Pérez, T. A., Barrena, R., Pérez-Yarza, E. G., y Pikatza, J. M. (2017). Improvement of newborn screening using a fuzzy inference system. *Expert Systems with Applications*, 78:301–318.
- Segundo, U., Buenestado, D., Pikatza, J. M., y Aldámiz-Echevarría, L. (2009). Guía de práctica clínica electrónica para el diagnóstico y tratamiento de la hiperamoniemia. En *Actas del VIII Congreso Nacional de Errores Congénitos del Metabolismo*.
- Segundo, U., Iruetaguena, A., Aldámiz-Echevarría, L., Buenestado, D., y Pikatza, J. M. (2015a). Computerized detection of inborn errors of metabolism in newborns. En *Proceedings of the 12th Annual Meeting Health Technology Assessment international (HTAi 2015)*.
- Segundo, U., López-Cuadrado, J., Aldámiz-Echevarría, L., Pérez, T. A., Buenestado, D., Iruetaguena, A., Barrena, R., y Pikatza, J. M. (2015b). Automatic construction of fuzzy inference systems for computerized clinical guidelines and protocols. *Applied Soft Computing*, 26:257–269.
- Segundo, U., Pikatza, J. M., Buenestado, D., Iruetaguena, A., Barrena, R., García, J. J., Aldámiz-Echevarría, L., y Sanjurjo, P. (2010). Integración de sistemas de clasificación difusos en una guía clínica ejecutable. En *Actas del III Simposio Sobre Lógica Fuzzy y Soft Computing*.
- Seki, H. y Mizumoto, M. (2012). On the properties of sirms connected fuzzy inference method with consequent fuzzy sets. En *IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2012*.
- Seki, H. y Mizumoto, M. (2013). Sirms connected fuzzy inference method adopting emphasis and suppression. *Fuzzy Sets and Systems*, 215:112–126.
- Selic, B. (2003). The pragmatics of model-driven development. *IEEE Software*, 20 (5):19–25.
- Shahar, Y., Miksch, S., y Johnson, P. D. (1998). The asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial Intelligence in Medicine*, 14:29–51.

- Shahar, Y., Young, O., Shalom, E., Galperin, M., Mayaffit, A., Moskovitch, R., y Hessing, A. (2004). A framework for a distributed, hybrid, multiple-ontology clinical-guideline library, and automated guideline-support tools. *Journal of Biomedical Informatics*, 37 (5):325–344.
- Shankar, R. D., Tu, S., Martins, S., Fagan, L. M., Goldstein, M. K., y Musen, M. A. (2001). Integration of textual guideline documents with formal guideline knowledge bases. En *Proceedings of the American Medical Informatics Association Symposium 2001*.
- Shiffman, R. N., Karras, B. T., Agrawal, A., Chen, R., Marengo, L., y Nath, S. (2000). Gem: a proposal for a more comprehensive guideline document model using xml. *Journal of the American Medical Informatics Association*, 7 (5):488–498.
- Shiffman, R. N., Michel, G., Essaihi, A., y Thornquist, E. (2004). Bridging the guideline implementation gap: A systematic, document-centered approach to guideline implementation. *Journal of the American Medical Informatics Association*, 11 (5):418–426.
- Silveira, G. P. y de Barros, L. C. (2015). Analysis of the dengue risk by means of a takagi-sugeno-style model. *Fuzzy Sets and Systems*, 277:122–137.
- Silverman, B. y Jones, M. (1989). E. fix and j.l. hodges (1951): An important contribution to nonparametric discriminant analysis and density estimation: Commentary on fix and hodges. *International Statistical Review*, páginas 233–238.
- Sobrado, F., Pikatza, J. M., Larburu, I., García, J. J., y de Ipiña, D. L. (2004). Towards a clinical practice guideline implementation for asthma treatment. *Lecture Notes in Artificial Intelligence*, 3040:587–596.
- Song, H., Miao, C., Roel, W., Shen, Z., y Catthoor, F. (2010). Implementation of fuzzy cognitive maps based on fuzzy neural network and application in prediction of time series. *IEEE Transactions on Fuzzy Systems*, páginas 233–250.
- Sonnenberg, F. A. y Hagerty, C. G. (2006). Computer-interpretable clinical practice guidelines. where are we and where are we going? *Yearbook of Medical Informatics*, páginas 145–158.
- Soria, J. L. M., Aldámiz-Echevarría, L., Ramos, D. E. C., Serra, J. D., Sánchez, A. F., Lamuño, D. G., Fita, M. J. J., Jiménez, L. M. J., y Cerdá, C. P. (2009). Programas de cribado neonatal en españa: Actualización y propuestas de futuro. Informe técnico, Sociedad Española de Bioquímica Clínica y Patología Molecular (SEQC).
- Stahl, T., Völter, M., Bettin, J., Haase, A., Helsen, S., Czarnecki, K., y Stockfleth, B. V. (2006). *Model-Driven Software Development: Technology, Engineering, Management*. John Wiley & Sons.
- Sutton, D. R. y Fox, J. (2003). The syntax and semantics of the proforma guideline modeling language. *Journal of the American Medical Informatics Association*, 10 (5):433–443.

- Takagi, T. y Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15 (1):116–132.
- Terenziani, P., Carlini, C., y Montani, S. (2002). Towards a comprehensive treatment of temporal constraints in clinical guidelines. En *IEEE International Symposium on Temporal Representation and Reasoning*.
- Terenziani, P., Molino, G., y Torchio, M. (2001). A modular approach for representing and executing clinical guidelines. *Artificial Intelligence in Medicine*, 23 (3):249–276.
- TGL (2017). *Electronical Therapeutical Guidelines*. Therapeutic Guideline Limited. <http://tg.org.au>.
- Tran, N., Michel, G., Krauthammer, M., y Shiffman, R. N. (2009). Embedding the guideline elements model in web ontology language. En *Proceedings of the American Medical Informatics Association 2009*.
- Trapote, R. C. A., Laita, J. A. C., Subías, J. E., Rodríguez, G. M. F., Ángeles García Díaz, Rodríguez, S. G., Vera, C. J. G., Fraile, A. G., Rodríguez, J. D. G., Romero, J. G., Pablo, C. L., Sánchez, J. I. M., Sangrador, C. O., Gancedo, L. R., Fernández, L. M. R., Frago, T. S., y Fuente, B. V. (2011). *Guía de Práctica Clínica sobre Infección del Tracto Urinario (ITU) en la Población Pediátrica*. Ministerio de Sanidad, Política Social e Igualdad.
- Trefz, F. K., Frauendienst-Egger, G., y Götz, H. (2017). *Metabolic & Genetic Information Center*. Metagene. <http://www.metagene.de>.
- Tu, S., Campbell, J., Glasgow, J., Nyman, M. A., McClure, R., McClay, J., Parker, C., Hrabak, K. M., Berg, D., Weida, T., Mansfield, J. G., Musen, M. A., y Abarbanel, R. M. (2007). The sage guideline model: achievements and overview. *Journal of the American Medical Informatics Association*, 14 (5):589–598.
- Tu, S. y Glasgow, J. (2006). Sage guideline model technical specification. Informe técnico, SAGE Consortium.
- UMLS (2017). *Unified Medical Language System*. National Library of Medicine. <https://www.nlm.nih.gov/research/umls/>.
- van den Berg, J., Kaymak, U., y van den Bergh, W. (2002). Fuzzy classification using probability-based rule weighting. En *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems*, páginas 991–996.
- van Leekwijck, W. y Kerre, E. E. (1999). Defuzzification: criteria and classification. *Fuzzy Sets and Systems*, 108.
- Vandvik, P. O., Brandt, L., Kristiansen, A., Guyatt, G., Heen, A. F., Agoritsas, T., y Berntzen, C. F. (2017). *MAGIC*. MAGIC. <http://magicproject.org/>.
- Vapnik, V. (1995). *The nature of statistical learning theory*. Springer Verlag, London.
- Vilarinho, L., Rocha, H., Sousa, C., Marcao, A., Fonseca, H., Bogas, M., y Osório, R. V. (2010). Four years of expanded newborn screening in portugal with tandem mass spectrometry. *Journal of Inherited Metabolic Disease*, 33 (3):133–138.

- Wang, L.-X. (1997). *A Course in Fuzzy Systems and Control*. Prentice Hall.
- Xpand (2017). *Xpand*. Eclipse Foundation.  
<https://eclipse.org/modeling/m2t/?project=xpand>.
- Yardimci, A. (2009). Soft computing in medicine. *Applied Soft Computing*, 9(3):1029–1043.
- Yubazaki, N., Yi, J., y Hirota, K. (1997). Sirms (single input rule modules) connected fuzzy inference model. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 1 (1):23–30.
- Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8 (3):338–253.



## Parte VI

---

### Anexos



## A

---

### Trastornos evaluados en el DD de cribado neonatal

En este anexo se muestran todos los trastornos que se tienen en cuenta en el DD de cribado neonatal. En las tablas A.1 y A.2 se muestran los trastornos de aminoácidos, en las tablas A.3 y A.4 los trastornos de ácidos orgánicos y en la tabla A.5 los trastornos de ácidos grasos. Para cada trastorno se muestra su nombre en inglés, así como su abreviatura más común y, en caso de tenerlo, su código de identificación en el sitio web OMIM (OMIM, 2017).

Hay que tener en cuenta que la acidemia metilmalónica es un caso especial (ver tabla A.4). Se trata de un trastorno genéticamente heterogéneo que incluye tanto a pacientes con un defecto enzimático parcial o completo y pacientes con defectos en la síntesis de la coenzima adenosilcobalamina. Mientras que los pacientes del primer caso se etiquetan como MUT, aquellos que tienen el defecto de la cobalamina se clasifican de acuerdo al grupo de complementación de la coenzima específico (Cbl A, B y Cbl C, D). El trastorno B12 Def (mat) se refiere a recién nacidos que tienen una acidemia metilmalónica debido a un déficit de vitamina B12 de la madre.

#	Trastorno de aminoácido	Abreviatura	Código OMIM
1	5,10-Methylenetetrahydrofolate reductase	MTHFR	607093
2	Argininemia	ARG	207800
3	Arginosuccinic aciduria	ASA	207900
4	Carbamoyl phosphate synthetase I deficiency, Hyperammonemia due to	CPS	237300

**Tabla A.1.** Lista de los trastornos de aminoácidos que se valoran en el DD de cribado neonatal

#	Trastorno de aminoácido	Abreviatura	Código OMIM
5	Citrullinemia, classic	CIT I	215700
6	Citrullinemia, type II, Neonatal-onset	CIT II	605814
7	Glycine encephalopathy	NKHG	605899
8	Homocystinuria due to cystathionine beta-synthase deficiency	HCY	236200
9	Maple syrup urine disease	MSUD	248600
10	Methionine adenosyltransferase deficiency	MET	250850
11	Ornithine transcarbamylase deficiency, Hyperammonemia due to	OTC	311250
12	Phenylketonuria	PKU	261600
13	Pyruvate carboxylase deficiency	PC	266150
14	Tyrosinemia, type I	TYR I	266150
15	Tyrosinemia, type II	TYR II	276600

**Tabla A.2.** Lista de los trastornos de aminoácidos que se valoran en el DD de cribado neonatal (continuación)

#	Trastorno de ácido orgánico	Abreviatura	Código OMIM
1	2-Methylbutyryl-CoA dehydrogenase deficiency	2MBG	610006
2	3-Hydroxy-3-Methylglutaryl-CoA lyase deficiency	HMG	246450
3	3-Methylcrotonyl-CoA carboxylase 1 deficiency	3MCC	210200
4	3-Methylcrotonyl-CoA carboxylase, deficiency, Maternal	3MCC (mat)	

**Tabla A.3.** Lista de los trastornos de ácidos orgánicos que se valoran en el DD de cribado neonatal

#	Trastorno de ácido orgánico	Abreviatura	Código OMIM
5	3-Methylglutaconic aciduria, Type 1	3MGA	250950
6	Alpha-Methylacetoacetic aciduria	BKT	203750
7	Encephalopathy, Ethylmalonic	EE	602473
8	Holocarboxylase synthetase deficiency	MCD	253270
9	Glutamate formiminotransferase deficiency	FIGLU	229100
10	Glutaric acidemia I	GA I	231670
11	Isobutyryl-CoA dehydrogenase deficiency	IBG	611283
12	Isovaleric acidemia	IVA	243500
13	Malonyl-CoA decarboxylase deficiency	MAL	248360
14	Methylmalonic aciduria, cblA type	Cbl A, B	251100
15	Methylmalonic aciduria, cblB type	Cbl A, B	251110
16	Methylmalonic aciduria and homocystinuria, cblC type	Cbl C, D	277400
17	Methylmalonic aciduria and homocystinuria, cblD type	Cbl C, D	277410
18	Methylmalonic aciduria due to methylmalonyl-CoA mutase deficiency	MUT	251000
19	Propionic acidemia	PROP	606054
20	Vitamin B12 deficiency, Maternal	B12 Def (mat)	

**Tabla A.4.** Lista de los trastornos de ácidos orgánicos que se valoran en el DD de cribado neonatal (continuación)

#	Trastorno de ácidos grasos	Abreviatura	Código OMIM
1	3-Hydroxyacyl-CoA dehydrogenase deficiency	SCHAD	231530
2	Acyl-CoA dehydrogenase, Medium chain, Deficiency of	MCAD	201450
3	Acyl-CoA dehydrogenase, Medium chain, Deficiency of, Heterozygous	MCAD (het)	
4	Acyl-CoA dehydrogenase, Short chain, Deficiency of	SCAD	201470
5	Acyl-CoA dehydrogenase, Very long-chain, Deficiency of	VLCAD	201475
6	Carnitine deficiency, Systemic primary	CUD	212140
7	Carnitine deficiency, Systemic primary, Maternal	CUD (mat)	
8	Carnitine palmitoyltransferase I deficiency	CPT I	255120
9	Carnitine palmitoyltransferase II deficiency, Lethal neonatal	CPT II	608836
10	Carnitine-acylcarnitine translocase deficiency	CACT	212138
11	Long chain 3-Hydroxyacyl-CoA dehydrogenase deficiency	LCHAD	609016
12	Mitochondrial trifunctional protein deficiency	TFP	609015
13	Multiple Acyl-CoA dehydrogenase deficiency	MADD	231680

**Tabla A.5.** Lista de los trastornos de ácidos grasos que se valoran en el DD de cribado neonatal

## B

---

### Variables clínicas tenidas en cuenta en el DD de cribado neonatal

En este anexo se muestran todas las variables clínicas que se tienen en cuenta en el DD de cribado neonatal. Las variables se han distribuido en tres conjuntos dependiendo de su tipo. Así, la tabla B.1 muestra las variables clínicas que son aminoácidos, la tabla B.2 muestra las variables que son acilcarnitinas y la tabla B.3 muestra las variables que son proporciones entre las otras variables clínicas.

#	Aminoácido	#	Aminoácido	#	Aminoácido
1	Alanina	6	Glutamina	11	Succinilacetona
2	Arginina	7	Homocisteína	12	Fenilalanina
3	Argininosuccinato	8	Lisina	13	Tirosina
4	Citrulina	9	Metionina	14	Valina
5	Glicina	10	Ornitina	15	Xleucina

**Tabla B.1.** Lista de los aminoácidos que se valoran en el DD de cribado neonatal

#	Acilcarnitina	#	Acilcarnitina	#	Acilcarnitina
1	C0	10	C5:1	19	C14:2
2	C3	11	C6	20	C16
3	C3DC	12	C6DC	21	C16-OH
4	C4	13	C8	22	C16:1-OH
5	C4-OH	14	C10	23	C18
6	C4DC	15	C10:1	24	C18-OH
7	C5	16	C12	25	C18:1
8	C5-OH	17	C14	26	C18:1-OH
9	C5DC	18	C14:1	27	C18:2

**Tabla B.2.** Lista de las acilcarnitinas que se valoran en el DD de cribado neonatal

#	Proporción	#	Proporción	#	Proporción	#	Proporción
1	Arginina / Ornitina	10	Xleucina / Alanina	19	C4 / C8	28	C14:2
2	Argininosuccinato / Arginina	11	Xleucina / Fenilalanina	20	C5 / C0	29	C14:2
3	Citrulina / Arginina	12	C0 / (C16 + C18)	21	C5 / C2	30	C14:2
4	Citrulina / Fenilalanina	13	C3 / Metionina	22	C5 / C3	31	C14:2
5	Fenilalanina / Tirosina	14	C3 / C2	23	C5-OH / C0	32	C14:2
6	Glutamato / Citrulina	15	C3 / C16	24	C5-OH / C8	33	C14:2
7	Glutamina / Citrulina	16	C3DC / C10	25	C18:1		
8	Metionina / Fenilalanina	17	C4 / C2	26	C18:1-OH		
9	Valina / Fenilalanina	18	C4 / C8	27	C18:2		

**Tabla B.3.** Lista de las proporciones que se valoran en el DD de cribado neonatal

## C

### Diagnóstico Diferencial de cribado neonatal

En este anexo se especifica el DD de cribado neonatal. Los trastornos se muestran en la primera columna y para cada uno de ellos en la segunda columna se muestran las variables clínicas que les afectan. Las columnas etiquetadas como {1}, {2}, {3}, {4} y {5} hacen referencia a las cinco fuentes bibliográficas consultadas para definir el DD, (Soria et al., 2009), (Vilarinho et al., 2010), (McHugh et al., 2011), (Lund et al., 2012) y (Trefz et al., 2017), respectivamente. Para cada una de estas fuentes se muestra si la fuente hace referencia a la relación entre la variable clínica y el trastorno y cómo es esa relación. Así, si la celda está vacía significa que la fuente bibliográfica no tiene en cuenta la variable clínica para ese trastorno. Un símbolo positivo (+) indica que la variable clínica tiene un valor elevado respecto al trastorno, mientras que un símbolo negativo (-) significa que la variable clínica tiene un valor por debajo de la normalidad para ese trastorno. Por último, la última columna muestra el peso (en el rango  $[0, 1]$ ) que se ha indicado para cada una de las relaciones entre las variables clínicas y los trastornos en el modelo de DD.

Trastorno	Variable clínica	{1}	{2}	{3}	{4}	{5}	Peso
<b>NHKG</b>	Glicina			+		+	1.0000
	Arginina	+	+	+	+	+	0.6667
	Citrulina / Arginina			-			0.1111
	Argininosuccinato / Arginina			-			0.1111
<b>ARG</b>	Arginina / Ornitina				+		0.1111

**Tabla C.1.** DD de cribado neonatal

Trastorno	Variable clínica	{1}	{2}	{3}	{4}	{5}	Peso
<b>HCY</b>	Metionina	+	+	+		+	0.6250
	Homocisteína			+		+	0.2500
	Metionina / Fenilalanina			+			0.1250
<b>MET</b>	Metionina		+	+			0.6000
	Metionina / Fenilalanina			+			0.4000
<b>MTHFR</b>	Metionina			-		-	0.4000
	Homocisteína			+		+	0.4000
	Metionina / Fenilalanina			-			0.2000
<b>CIT I</b>	Citrulina	+	+	+		+	0.6250
	Citrulina / Arginina			+	+		0.2500
	Citrulina / Fenilalanina			+			0.1250
<b>CIT II</b>	Citrulina	+		+		+	0.5000
	Citrulina / Arginina			+	+		0.2500
	Citrulina / Fenilalanina			+			0.1250
	Arginina			+			0.1250
<b>ASA</b>	Argininosuccinato		+	+	+	+	0.3846
	Citrulina	+		+	+	+	0.3077
	Argininosuccinato / Arginina			+	+		0.1538
	Citrulina / Fenilalanina			+	+		0.0769
	Glutamina					+	0.0385
	Lisina					+	0.0385
<b>PC</b>	Citrulina			+		+	0.3846
	Citrulina / Arginina			+			0.1538
	Citrulina / Fenilalanina			+			0.1538
	Tirosina			+			0.1538
	Alanina					+	0.0769
	Lisina					+	0.0769
<b>OTC</b>	Citrulina			-		-	0.3333
	Glutamato / Citrulina			+			0.1667
	Citrulina / Fenilalanina			-			0.1667
	Glutamina / Citrulina			+			0.1667
	Glutamina					+	0.0833
	Ornitina					-	0.0833

Tabla C.2. DD de cribado neonatal (continuación)

Trastorno	Variable clínica	{1}	{2}	{3}	{4}	{5}	Peso
<b>CPS</b>	Citrulina			-		-	0.3636
	Glutamato / Citrulina			+			0.1818
	Citrulina / Fenilalanina			-			0.1818
	Glutamina / Citrulina			+			0.1818
	Glutamina					+	0.0909
<b>MSUD</b>	Xleucina	+	+	+	+	+	0.3846
	Valina	+	+			+	0.2308
	Xleucina / Alanina			+	+		0.1538
	Xleucina / Fenilalanina			+	+		0.1538
	Valina / Fenilalanina			+			0.0769
<b>PKU</b>	Fenilalanina	+	+	+		+	0.5714
	Fenilalanina / Tirosina		+	+		+	0.4286
<b>TYR I</b>	Tirosina	+	+	+		+	0.6154
	Succinilacetona			+	+		0.3070
	Metionina					+	0.0769
<b>TYR II</b>	Metionina	+	+	+		+	1.0000
<b>3MCC</b>	C5-OH	+	+	+	+	+	0.6250
	C5-OH / C0			+	+		0.2500
	C5-OH / C8			+			0.1250
<b>3MCC (mat)</b>	C5-OH / C0			+			0.3333
	C5-OH / C8			+			0.3333
	C5-OH			+			0.1667
	C0			-			0.1667
<b>3MGA</b>	C5-OH	+		+	+		0.5000
	C5-OH / C0			+	+		0.3333
	C5-OH / C8			+			0.1667
<b>HMG</b>	C5-OH	+	+	+	+		0.4000
	C6DC	+	+	+		+	0.3000
	C5-OH / C0			+	+		0.2000
	C5-OH / C8			+			0.1000
<b>MCD</b>	C5-OH		+	+	+		0.4286
	C5-OH / C0			+	+		0.2857
	C5-OH / C8			+			0.1428
	C3			+			0.1428

Tabla C.3. DD de cribado neonatal (continuación)

Trastorno	Variable clínica	{1}	{2}	{3}	{4}	{5}	Peso
<b>BKT</b>	C5:1			+	+	+	0.4286
	C5-OH			+		+	0.2857
	C4-OH			+			0.1428
	C5-OH / C8			+			0.1428
<b>Cbl C, D</b>	C3	+	+	+			0.2857
	C3 / C2	+		+			0.1905
	C3 / C16	+		+			0.1905
	C3 / Metionina		+	+			0.1905
	Metionina		-			-	0.0952
	Homocisteína					+	0.0476
<b>B12 Def (mat)</b>	C3			+			0.4000
	C3 / C2			+			0.2000
	C3 / C16			+			0.2000
	C3 / Metionina			+			0.2000
<b>MUT</b>	C3	+	+	+	+	+	0.3846
	C3 / C2	+	+	+	+		0.3078
	C3 / C16	+		+			0.1538
	C4DC				+	+	0.1538
<b>Cbl A, B</b>	C3 / C2	+		+			0.4000
	C3 / C16	+		+			0.2667
	C3	+		+			0.2667
	Glicina					+	0.0667
<b>PROP</b>	C3	+	+	+	+	+	0.4167
	C3 / C2	+	+	+	+		0.3333
	C3 / C16	+		+			0.1667
	Glutamina					+	0.0417
	Glicina					+	0.0417
<b>2MBG</b>	C5			+			0.3750
	C5 / C2			+			0.2500
	C5 / C0			+			0.1250
	C5 / C3			+			0.1250
	C5-OH	+					0.1250

Tabla C.4. DD de cribado neonatal (continuación)

Trastorno	Variable clínica	{1}	{2}	{3}	{4}	{5}	Peso
<b>IVA</b>	C5	+	+	+		+	0.5000
	C5 / C3			+			0.2500
	C5 / C2			+			0.1250
	C5 / C0			+			0.1250
<b>EE</b>	C4			+			0.4286
	C4 / C3			+			0.2857
	C4 / C2			+			0.1428
	C5 / C3			+			0.1428
<b>FIGLU</b>	C4 / C3			+			0.4286
	C4			+			0.2857
	C4 / C2			+			0.1428
	C4 / C8			+			0.1428
<b>IBG</b>	C4	+		+		+	0.5555
	C4 / C3			+			0.2222
	C4 / C2			+			0.1111
	C4 / C8			+			0.1111
<b>GA I</b>	C5DC	+	+	+	+	+	0.5000
	C5DC / C16			+			0.2500
	C5DC / C5-OH			+			0.1667
	C5DC / C8			+			0.0833
<b>MAL</b>	C3DC / C10			+			0.6667
	C3DC			+			0.3333
<b>CUD</b>	C0	-	-	-	-		0.4444
	C3			-			0.2222
	C18:1			-			0.2222
	C3 / Metionina			-			0.1111
<b>CUD (mat)</b>	C0			-			0.4286
	C18:2			-			0.2857
	C18:1			-			0.1428
	C3 / Metionina			-			0.1428

Tabla C.5. DD de cribado neonatal (continuación)

Trastorno	Variable clínica	{1}	{2}	{3}	{4}	{5}	Peso
<b>CPT I</b>	C0 / (C16 + C18)	+	+	+	+		0.2857
	C16	-		-		-	0.2381
	C0	+		+			0.1905
	C3 / C16			+			0.1428
	C18	-					0.1428
<b>CPT II</b>	C16	+			+	+	0.2000
	C18:1	+			+	+	0.2000
	(C16 + C18:1) / C2			+			0.1333
	C14:1 / C2			+			0.1333
	C18	+					0.0667
	C16-OH			+			0.0667
	C18:2			+			0.0667
	C0 / (C16 + C18)		-				0.0667
C16:1-OH			+			0.0667	
<b>CACT</b>	C16	+		+	+		0.3000
	(C16 + C18:1) / C2			+			0.2000
	C18	+					0.1000
	C18:1	+			+		0.1000
	C18:2	+					0.1000
	C5DC / C5-OH			+			0.1000
	C18:1-OH			+			0.1000
<b>MADD</b>	C10	+	+	+		+	0.1379
	C4	+	+		+	+	0.1379
	C12		+	+		+	0.1034
	C5	+	+		+		0.1034
	C8	+	+		+		0.1034
	C16		+		+	+	0.1034
	C6	+	+				0.0689
	C14		+			+	0.0689
	C14:1 / C2			+			0.0345
	C8 / C2			+			0.0345
	C18		+				0.0345
	C5DC				+		0.0345
	C14:1				+		0.0345

Tabla C.6. DD de cribado neonatal (continuación)

Trastorno	Variable clínica	{1}	{2}	{3}	{4}	{5}	Peso
<b>VLCAD</b>	C14:1	+	+	+	+	+	0.3125
	C14:2	+	+	+		+	0.2500
	C14:1 / C16	+		+			0.1250
	C14	+				+	0.1250
	C14:1 / C2			+	+		0.1250
	C18:1					+	0.0312
	C16					+	0.0312
<b>LCHAD</b>	C18:1-OH	+	+	+	+	+	0.2703
	C16-OH	+	+	+	+	+	0.2703
	C18-OH	+	+	+		+	0.2162
	C16-OH / C16	+	+	+			0.1622
	C16:1-OH	+					0.0540
	C18:1					+	0.0270
<b>TFP</b>	C18:1-OH			+	+		0.3333
	C16-OH			+	+		0.3333
	C16-OH / C16			+			0.1667
	C18-OH			+			0.1667
<b>MSCHAD</b>	C5DC / C16			+			0.4286
	C4-OH			+			0.2857
	C3 / C16			+			0.1428
	C5DC / C8			+			0.1428
<b>MCAD</b>	C8	+	+	+	+	+	0.3448
	C8 / C10	+	+	+	+	+	0.3448
	C6			+		+	0.1379
	C8 / C2			+			0.0690
	C10			+			0.0690
	C10:1					+	0.0345
<b>MCAD (het)</b>	C8			+			0.4286
	C10			+			0.2857
	C8 / C2			+			0.1428
	C6			+			0.1428
<b>SCAD</b>	C4	+		+	+	+	0.2727
	C4 / C2	+		+	+	+	0.2727
	C4 / C3	+		+	+	+	0.2727
	C4 / C8	+		+			0.1818

Tabla C.7. DD de cribado neonatal (continuación)



## D

---

### Resultados obtenidos por el SID de cribado neonatal

En este anexo se muestran los resultados obtenidos por el SID de cribado neonatal para las pruebas realizadas en el capítulo 12. Las muestras de cribado se han etiquetado como *Cribado\_XX* y las muestras clínicas se han etiquetado como *Clinica\_XX*. La primera columna presenta el código de la muestra, mientras la segunda columna indica el resultado esperado y la tercera columna el resultado obtenido por el SID. La cuarta columna muestra la probabilidad de los resultados de la tercera columna (cuando es necesario) y finalmente, la quinta columna indica el Índice de Calidad de la Información (ICI) de los resultados.

Sólo se muestran los resultados para aquellas variables de salida que hayan obtenido una probabilidad mayor que 50 %. En caso de que todas ellas obtengan una probabilidad por debajo de ese umbral el resultado de la muestra se considera normal.

<b>Muestra</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	<b>Probabilidad %</b>	<b>ICI %</b>
<b>Cribado_1</b>	TYR	TYR II	86,11	86,11
		TYR II	86,11	86,11
<b>Cribado_2</b>	TYR	TYR I	53,37	91,07
<b>Cribado_3</b>	Normal	Normal		
<b>Cribado_4</b>	Normal	Normal		
<b>Cribado_5</b>	Normal	Normal		
<b>Cribado_6</b>	Normal	Normal		
<b>Cribado_7</b>	Normal	Normal		

**Tabla D.1.** Resultados obtenidos por el SID de cribado neonatal

Muestra	Resultado esperado	Resultado obtenido	Probabilidad %	ICI %
Cribado_8	Normal	Normal		
Cribado_9	Normal	Normal		
Cribado_10	Normal	Normal		
Cribado_11	Normal	Normal		
Cribado_12	Normal	Normal		
Cribado_13	Normal	Normal		
Cribado_14	Normal	Normal		
Cribado_15	Normal	Normal		
Cribado_16	Normal	Normal		
Cribado_17	Normal	Normal		
Cribado_18	Normal	Normal		
Cribado_19	Normal	Normal		
Cribado_20	Normal	Normal		
Cribado_21	Normal	Normal		
Cribado_22	TYR	TYR II	76,58	86,11
Cribado_23	Normal	Normal		
Cribado_24	Normal	Normal		
Cribado_25	Normal	Normal		
Cribado_26	Normal	Normal		
Cribado_27	Normal	Normal		
Cribado_28	Normal	Normal		
Cribado_29	Normal	Normal		
		TYR II	67,36	86,11
Cribado_30	TYR	ARG	50,92	92,31
Cribado_31	Normal	Normal		
Cribado_32	Normal	Normal		
Cribado_33	Normal	Normal		
Cribado_34	Normal	Normal		

Tabla D.2. Resultados obtenidos por el SID de cribado neonatal (continuación)

Muestra	Resultado esperado	Resultado obtenido	Probabilidad %	ICI %
Cribado_35	TYR	TYR II	86,11	86,11
		TYR I	55,36	91,07
Cribado_36	TYR	TYR II	86,11	86,11
		TYR II	86,11	86,11
		TYR I	55,36	91,07
Cribado_37	TYR	TYR II	86,11	86,11
		TYR I	54,06	91,07
Cribado_38	Normal	Normal		
Cribado_39	Normal	Normal		
Cribado_40	Normal	Normal		
Cribado_41	Normal	Normal		
Cribado_42	TYR	TYR II	86,11	86,11
Cribado_43	Normal	Normal		
Cribado_44	Normal	Normal		
Cribado_45	Normal	Normal		
Cribado_46	Normal	Normal		
Cribado_47	Normal	Normal		
Cribado_48	Normal	Normal		
Cribado_49	Normal	Normal		
Cribado_50	Normal	Normal		
Cribado_51	Normal	Normal		
Cribado_52	Normal	Normal		
Cribado_53	Normal	Normal		
Cribado_54	Normal	Normal		
Cribado_55	Normal	Normal		
Cribado_56	Normal	Normal		
Cribado_57	Normal	Normal		
Cribado_58	Normal	Normal		
Cribado_59	Normal	Normal		
Cribado_60	Normal	Normal		

Tabla D.3. Resultados obtenidos por el SID de cribado neonatal (continuación)

<b>Muestra</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	<b>Probabilidad %</b>	<b>ICI %</b>
<b>Cribado_61</b>	Normal	Normal		
<b>Cribado_62</b>	Normal	Normal		
<b>Cribado_63</b>	Normal	Normal		
<b>Cribado_64</b>	Normal	Normal		
<b>Cribado_65</b>	Normal	Normal		
<b>Cribado_66</b>	Normal	Normal		
<b>Cribado_67</b>	Normal	Normal		
<b>Cribado_68</b>	Normal	Normal		
<b>Cribado_69</b>	Normal	Normal		
<b>Cribado_70</b>	Normal	Normal		
<b>Cribado_71</b>	Normal	Normal		
<b>Cribado_72</b>	Normal	Normal		
<b>Cribado_73</b>	Normal	Normal		
<b>Cribado_74</b>	Normal	Normal		
<b>Cribado_75</b>	Normal	Normal		
<b>Cribado_76</b>	Normal	Normal		
<b>Cribado_77</b>	Normal	Normal		
<b>Cribado_78</b>	Normal	Normal		
<b>Cribado_79</b>	Normal	Normal		
<b>Cribado_80</b>	Normal	Normal		
<b>Cribado_81</b>	Normal	Normal		
<b>Cribado_82</b>	Normal	Normal		
<b>Cribado_83</b>	Normal	Normal		
<b>Cribado_84</b>	Normal	Normal		
<b>Cribado_85</b>	Normal	Normal		
<b>Cribado_86</b>	Normal	Normal		
<b>Cribado_87</b>	Normal	Normal		
<b>Cribado_88</b>	Normal	Normal		
<b>Cribado_89</b>	Normal	Normal		
<b>Cribado_90</b>	Normal	Normal		

Tabla D.4. Resultados obtenidos por el SID de cribado neonatal (continuación)

<b>Muestra</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	<b>Probabilidad %</b>	<b>ICI %</b>
<b>Cribado_91</b>	Normal	Normal		
<b>Cribado_92</b>	TYR	TYR I	53,48	91,07
<b>Cribado_93</b>	TYR	TYR II	86,11	86,11
<b>Cribado_94</b>	Normal	Normal		
<b>Cribado_95</b>	Normal	Normal		
<b>Cribado_96</b>	Normal	Normal		
<b>Cribado_97</b>	Normal	Normal		
<b>Cribado_98</b>	Normal	Normal		
<b>Cribado_99</b>	Normal	Normal		
<b>Cribado_100</b>	Normal	Normal		
<b>Cribado_101</b>	Normal	Normal		
<b>Cribado_102</b>	Normal	Normal		
<b>Cribado_103</b>	Normal	Normal		
<b>Cribado_104</b>	Normal	Normal		
<b>Cribado_105</b>	Normal	Normal		
<b>Clinica_1</b>	Normal	Normal		
<b>Clinica_2</b>	Normal	Normal		
		GA I	100,00	100,00
		CPT I	81,05	99,12
		MSCHAD	68,09	98,04
<b>Clinica_3</b>	GA I	MAL	66,67	100,00
<b>Clinica_4</b>	GA I	GA I	100,00	100,00
<b>Clinica_5</b>	Normal	Normal		
<b>Clinica_6</b>	Normal	Normal		
<b>Clinica_7</b>	Normal	Normal		
<b>Clinica_8</b>	Normal	Normal		
<b>Clinica_9</b>	Normal	Normal		
<b>Clinica_10</b>	Normal	Normal		

Tabla D.5. Resultados obtenidos por el SID de cribado neonatal (continuación)

<b>Muestra</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	<b>Probabilidad %</b>	<b>ICI %</b>
<b>Clinica_11</b>	Normal	Normal		
		Cbl C, D	100,00	100,00
		B12 Def (mat)	100,00	100,00
		PROP	98,59	100,00
		Cbl A, B	98,39	100,00
		MUT	91,04	100,00
		MCD	76,48	100,00
<b>Clinica_12</b>	PROP	3MCC	75,54	100,00
<b>Clinica_13</b>	Normal	Normal		
<b>Clinica_14</b>	Normal	Normal		
<b>Clinica_15</b>	Normal	Normal		
		Cbl C, D	100,00	100,00
		PROP	100,00	100,00
		B12 Def (mat)	100,00	100,00
		Cbl A, B	100,00	100,00
<b>Clinica_16</b>	PROP	MUT	92,54	100,00
		Cbl C, D	100,00	100,00
		B12 Def (mat)	99,21	100,00
		Cbl A, B	97,80	100,00
		PROP	96,04	100,00
<b>Clinica_17</b>	PROP	MUT	90,50	100,00
		Cbl A, B	72,58	98,39
		CUD	72,22	100,00
		CUD (mat)	66,00	70,00
		Cbl C, D	60,00	100,00
		PROP	55,99	92,96
		B12 Def (mat)	54,35	97,83
<b>Clinica_18</b>	PROP	MUT	52,24	98,51

Tabla D.6. Resultados obtenidos por el SID de cribado neonatal (continuación)

Muestra	Resultado esperado	Resultado obtenido	Probabilidad %	ICI %
Clinica_19	IVA	IVA	75,00	75,00
		2MBG	71,43	85,71
		CPT I	66,37	85,84
Clinica_20	Normal	Normal		
Clinica_21	MCAD	MCAD	77,29	100,00
		MCAD (het)	71,43	100,00
		CPT I	57,47	100,00
Clinica_22	HMG	3MCC	91,67	100,00
		3MGA	90,91	100,00
		3MCC (mat)	81,82	100,00
		MCD	78,31	84,34
		HMG	61,73	62,96
		CPT I	59,29	85,84
		2M3HBA	55,56	66,67
Clinica_23	MUT	Cbl C, D	100,00	100,00
		B12 Def (mat)	100,00	100,00
		MUT	98,51	100,00
		Cbl A, B	98,39	100,00
		PROP	95,95	100,00
		GA I	75,00	100,00
		BKT	62,97	100,00
		MSCHAD	58,82	98,04
Clinica_24	GA I	GA I	100,00	100,00
		CPT I	77,25	99,12
		MSCHAD	63,52	98,04
		MAL	59,01	100,00
Clinica_25	B12 Def (mat)	CPT I	81,00	99,12
		MSCHAD	58,82	98,04
		Cbl A, B	57,10	100,00

Tabla D.7. Resultados obtenidos por el SID de cribado neonatal (continuación)

<b>Muestra</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	<b>Probabilidad %</b>	<b>ICI %</b>
<b>Clinica_26</b>	CUD	CUD	72,22	72,22
		CUD (mat)	68,00	68,00
<b>Clinica_27</b>	MCAD	CUD	97,09	100,00
		MCAD	87,91	100,00
		MCAD (het)	67,05	100,00
		CUD (mat)	65,49	70,00
<b>Clinica_28</b>	PROP	Cbl C, D	100,00	100,00
		B12 Def (mat)	100,00	100,00
		PROP	98,44	100,00
		Cbl A, B	98,22	100,00
		MUT	83,43	100,00
		CPT I	76,96	99,12
		MSCHAD	52,56	98,04
<b>Clinica_29</b>	Normal	Normal		
<b>Clinica_30</b>	Normal	Normal		
<b>Clinica_31</b>	MUT	Cbl A, B	95,57	100,00
		Cbl C, D	95,34	100,00
		B12 Def (mat)	94,94	100,00
		MUT	94,16	100,00
		PROP	87,45	100,00
<b>Clinica_32</b>	MCAD	MCAD	50,00	100,00
<b>Clinica_33</b>	GA II	MCAD (het)	92,78	100,00
		GA I	59,12	100,00
		MSCHAD	58,82	98,04
		MCAD	53,28	100,00
<b>Clinica_34</b>	Normal	Normal		
<b>Clinica_35</b>	CUD	CUD	52,00	100,00

Tabla D.8. Resultados obtenidos por el SID de cribado neonatal (continuación)

<b>Muestra</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	<b>Probabilidad %</b>	<b>ICI %</b>
<b>Clinica_36</b>	LCHAD	LCHAD	98,61	100,00
		TFP	97,72	100,00
		GA I	85,66	100,00
		VLCAD	77,35	97,96
		CACT	67,37	87,90
		CPT II	62,48	93,06
		GA II	56,40	99,56
<b>Clinica_37</b>	VLCAD	CUD	98,15	100,00
		CUD (mat)	68,00	70,00
<b>Clinica_38</b>	Normal	Normal		
<b>Clinica_39</b>	MCAD	MCAD	84,08	100,00
		CUD	71,13	100,00
		MCAD (het)	65,34	100,00
<b>Clinica_40</b>	PROP	Cbl C, D	100,00	100,00
		B12 Def (mat)	100,00	100,00
		PROP	98,59	100,00
		Cbl A, B	98,39	100,00
		MUT	91,04	100,00
		CPT I	81,04	99,12
		MSCHAD	58,82	100,00
<b>Clinica_41</b>	GA I	GA I	100,00	100,00
		CPT I	53,98	99,12
<b>Clinica_42</b>	GA II	MCAD (het)	98,27	100,00
		MCAD	64,85	100,00
		CPT I	57,37	99,12
		Cbl A, B	56,11	100,00
		MSCHAD	55,31	98,04

**Tabla D.9.** Resultados obtenidos por el SID de cribado neonatal (continuación)

<b>Muestra</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	<b>Probabilidad %</b>	<b>ICI %</b>
<b>Clinica_43</b>	Normal	Normal		
		Cbl C, D	100,00	100,00
		B12 Def (mat)	100,00	100,00
		MUT	98,51	100,00
		Cbl A, B	98,39	100,00
		PROP	91,55	100,00
		GA I	79,06	100,00
		3MGA	72,73	100,00
		MCD	69,88	100,00
		BKT	69,77	100,00
		3MCC	66,67	100,00
		MSCHAD	65,60	100,00
		3MCC (mat)	54,55	100,00
		<b>Clinica_44</b>	MUT	HMG
3MCC	96,75			100,00
<b>Clinica_45</b>	MCD	3MGA	96,45	100,00
		3MCC (mat)	87,36	100,00
		MCD	82,56	100,00
		HMG	59,42	100,00
		MCAD	67,95	100,00
<b>Clinica_46</b>	MCAD	MCAD (het)	58,82	100,00
<b>Clinica_47</b>	Cbl A, B	CPT I	77,27	99,12
		Cbl C, D	61,04	100,00
		MUT	59,37	100,00
		Cbl A, B	58,24	100,00
		B12 Def (mat)	58,17	100,00
		MSCHAD	56,97	98,04
		PROP	51,08	100,00

Tabla D.10. Resultados obtenidos por el SID de cribado neonatal (continuación)