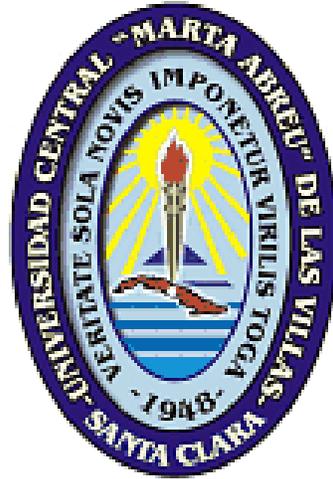


Universidad Central "Marta Abreu" de Las Villas  
Facultad de Matemática, Física y Computación  
Licenciatura en Ciencias de la Computación

---



# SISTEMA PRONÓSTICO DEL PACIENTE PEDIÁTRICO

Autor: Brenda Feliú Cruz

Tutores: MSc. Mabel González Castellanos

Dra. Yanet Rodríguez Sarabia

Consultante: Dr. Adonis Aguirre Roque

Dra. Lisdeily Carmona

---

Año 52 de la Revolución

# Hago Constar

Hago constar que el presente trabajo fue realizado en la Universidad Central Marta Abreu de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencias de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

---

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

---

Firma del tutor

---

Firma del jefe del Seminario

# Dedicatoria

A mis padres.

A Reinier.

# Agradecimientos

A papi por la posibilidad, el cariño y el apoyo.

Al Dr. Adonis por correr junto a mí todo el tiempo.

A Morell por ser tutor sin saberlo.

A Inti por estar siempre a mi lado.

A Reinier por sus cinco sentidos.

# **Pensamiento**

# Resumen

La valoración adecuada de un paciente pediátrico a la hora de ingresar en la Unidad de Cuidados Intensivos (UCI) es de gran interés, sobre todo si se da un pronóstico del nivel de gravedad del mismo atendiendo a rasgos epidemiológicos y clínicos.

En esta investigación se desarrolla un sistema que combina la teoría de conjuntos, conjuntos borrosos y Sistemas Basados en Casos (SBCasos), explotando las ventajas que ofrece cada uno de ellos. La teoría de conjuntos se aplica a la hora de representar aquellos rasgos multievaluados que puede presentar el paciente, ofreciendo la posibilidad de aplicar técnicas desarrolladas específicamente para el trabajo con este tipo de datos. El uso de conjuntos borrosos permite darle más naturalidad al planteamiento del problema al modelar los atributos numéricos como variables lingüísticas y además, posibilita la obtención de un puntaje en la clasificación de un nuevo caso. El SBCasos es el que pronostica el nivel de gravedad de un paciente determinado y justifica la solución dada al mismo citando experiencias previas.

El sistema implementado demostró su efectividad a partir de las pruebas practicadas.

# **Abstract**

The proper assessment of a pediatric patient when entering the Intensive Care Unit (ICU) is of great interest, especially if given a prognosis of the patient's severity in response to epidemiological and clinical features.

This research develops a system combining fuzzy sets and Case-Based Systems (SBCasos) exploiting the advantages of each. The use of fuzzy sets gives a more natural approach to the problem to model numeric attributes as linguistic variables. The SBCasos is the one that predicts the severity of individual patients and justifies the solution given by citing previous experiences. The implemented system proved its effectiveness from the evidence adduced.

# Índice

<b>Introducción</b>	<b>1</b>
Problema de investigación . . . . .	1
Objetivo General . . . . .	1
Objetivos Específicos . . . . .	2
Preguntas de Investigación . . . . .	2
Tareas de Investigación . . . . .	2
Hipótesis de Investigación . . . . .	2
Justificación de la investigación . . . . .	3
Viabilidad de la Investigación . . . . .	3
<b>1. La Inteligencia Artificial en la Solución de Problemas Médicos.</b>	<b>5</b>
1.1. Sistemas pronósticos . . . . .	5
1.2. Programa Inteligente . . . . .	7
1.2.1. Sistemas Basados en el Conocimiento (SBC) . . . . .	8
1.2.1.1. Definición y estructura de un SBC . . . . .	8
1.2.1.2. Sistemas Híbridos . . . . .	11
1.2.1.3. Ventajas y Desventajas de los SBC . . . . .	12
1.3. Lógica Borrosa . . . . .	13
1.3.1. Los conjuntos borrosos . . . . .	14

1.3.2.	Modelación de rasgos numéricos como variables lingüísticas . . . . .	14
1.3.3.	Los sistemas de inferencia borrosos . . . . .	15
1.3.4.	Heurísticas para determinar los parámetros de las funciones de pertenencia a un conjunto borroso . . . . .	17
1.4.	Aprendizaje Automatizado . . . . .	18
1.4.1.	Aprendizaje basado en instancias . . . . .	19
1.4.1.1.	Método de los $k$ vecinos más cercanos. . . . .	20
1.4.1.2.	Relación con SIB . . . . .	21
1.4.2.	Herramienta de aprendizaje automático Weka . . . . .	21
1.4.2.1.	Descripción general de Weka (Waikato Environment for Knowledge Analysis) . . . . .	21
1.4.2.2.	Tipos de datos que Weka implementa . . . . .	22
1.5.	Aplicaciones de los SBCasos a la medicina . . . . .	23
<b>2.</b>	<b>Implementación de Saprím versión 2.0</b>	<b>27</b>
2.1.	Análisis y diseño del sistema desarrollado . . . . .	28
2.1.1.	Modelo del negocio . . . . .	28
2.1.2.	Análisis de actores y casos de uso del negocio . . . . .	29
2.1.3.	Componentes del sistema . . . . .	32
2.2.	Modelación computacional del problema . . . . .	32
2.2.1.	Construcción de la base de casos . . . . .	33
2.2.2.	Modelo FuzzyIBk . . . . .	34
2.2.2.1.	Obtención de la región borrosa . . . . .	35
2.2.2.2.	Proceso de defuzificación . . . . .	36
2.2.2.3.	Etapas de fuzificación . . . . .	37
2.2.2.4.	Etapas de clasificación . . . . .	39

2.2.2.5.	Algunas consideraciones sobre FuzzyIBk . . . . .	41
2.2.3.	Principales clases que componen el sistema . . . . .	42
2.3.	Agregando funcionalidades a WEKA . . . . .	44
2.3.1.	Agregando un nuevo filtro . . . . .	45
2.3.2.	Agregando un nuevo clasificador . . . . .	46
2.4.	Conclusiones Parciales . . . . .	48
<b>3.</b>	<b>Manual de Usuario. Evaluación y Validación de los Resultados</b>	<b>49</b>
3.1.	Manual de Usuario . . . . .	49
3.1.1.	Introducción a Saprim versión 2.0 . . . . .	49
3.1.2.	Facilidades del sistema . . . . .	50
3.1.3.	Instalación y requerimientos del software . . . . .	51
3.1.4.	Interfaz Gráfica de Usuario . . . . .	51
3.1.4.1.	Ventana principal . . . . .	52
3.1.4.2.	Ventana de pronóstico . . . . .	52
3.1.4.3.	Ventana de administración de la base de conocimiento . . . . .	54
3.1.4.4.	Ventana de resultados . . . . .	56
3.1.4.5.	Menús . . . . .	57
3.1.5.	Pronóstico . . . . .	60
3.1.5.1.	Estimar nivel de gravedad . . . . .	60
3.1.5.2.	Manipulación e interpretación de los resultados . . . . .	61
3.1.6.	Actualización de la base de conocimientos . . . . .	61
3.2.	Evaluación y validación del software . . . . .	61
3.2.1.	Resultados Experimentales . . . . .	62
3.2.1.1.	Desempeño de NominalToLinguistic . . . . .	63
3.2.1.2.	FuzzyIBk vs IBk . . . . .	64
3.2.1.3.	Comparación con otros enfoques . . . . .	66
3.2.1.4.	Validación del modelo computacional obtenido . . . . .	67

<b>Conclusiones</b>	<b>68</b>
<b>Recomendaciones</b>	<b>69</b>
<b>Bibliografía</b>	<b>72</b>
<b>Anexos</b>	<b>73</b>

# Introducción

Elevar la calidad de la atención médica constituye una de las directivas específicas de la estrategia que se reitera en la política del Ministerio de Salud Pública (MINSAP). La atención pediátrica reviste gran importancia y prioridad, siendo conveniente, para la atención del paciente, una adecuada evaluación de su estado de gravedad en el momento de su llegada al hospital. Esta evaluación permite orientar al paciente hacia la sala de destino en caso de que sea tributario de ingreso.

De lo anteriormente expuesto se deriva el **Problema Científico** a resolver.

## Planteamiento del Problema

Se desea realizar una estimación a partir de los datos clínicos epidemiológicos del paciente sin incluir ningún tipo de análisis de laboratorio. El pronóstico obtenido será un puntaje que indica los niveles de cuidado, grave o crítico en que se puede encontrar el paciente, sugiriéndole al especialista su ingreso o no en la UCI (Unidad de Cuidados Intensivos). Este resultado conforma una primera apreciación del estado del paciente, permitiéndole al pediatra tomar las medidas adecuadas mientras espera los resultados de laboratorio para complementar su diagnóstico.

## Objetivo General

- Desarrollar un sistema computacional para pronosticar el nivel de gravedad de un paciente en el momento de su llegada al Hospital Infantil “José Luis Miranda” de Santa Clara,

combinando la aplicación de técnicas de la Inteligencia Artificial (IA) con vistas a dar una valoración rápida del estado del paciente.

### **Objetivos Específicos**

- Modelar los rasgos predictores de tipo conjunto con las diversas técnicas adaptadas para este fin.
- Implementar un filtro y un algoritmo de clasificación en la herramienta Weka para el trabajo con rasgo objetivo de tipo borroso.
- Obtener el modelo computacional que cuantifique el nivel de gravedad del paciente.

Para dar respuesta a los objetivos se formulan las siguientes preguntas de investigación:

### **Preguntas de Investigación**

- ¿Las técnicas de IA permitirán a partir de factores de riesgos epidemiológicos y clínicos discriminar entre pacientes graves, críticos y de cuidado?
- ¿Los resultados que ofrece este sistema computacional están al nivel de los obtenidos por los expertos humanos al valorar un paciente?

### **Tareas de Investigación**

- Obtener la muestra de aprendizaje.
- Aplicar el filtro implementado en Weka a la muestra de aprendizaje para convertir el rasgo objetivo en borroso.
- Validar la aplicación de un modelo basado en instancias para pronosticar rasgos objetivos de tipo borroso.

## **Hipótesis de Investigación**

- Es factible a partir de rasgos epidemiológicos y clínicos, discriminar entre pacientes graves, críticos y de cuidado mediante la aplicación de técnicas de clasificación supervisada de la IA.
- Los resultados que ofrece este sistema computacional están al nivel de los obtenidos por los expertos humanos al valorar un paciente.

## **Justificación de la Investigación**

Una adecuada valoración de la gravedad del paciente favorecería la concentración de los recursos humanos y materiales disponibles en la atención del paciente valorado como grave y evitar iatrogenia en los pacientes de cuidado. Por tanto, contar en una institución médica con un sistema computacional que asista satisfactoriamente al especialista en esta tarea, sería de gran utilidad.

## **Viabilidad de la Investigación**

El desarrollo de la informática y en particular de las técnicas de ingeniería del conocimiento y los lenguajes de programación de alto nivel, permiten enfrentar con éxito el desafío de desarrollar una aplicación computacional como la que se pretende. En particular esta aplicación se desarrolla en el marco de la colaboración de especialistas del Hospital Infantil “José Luis Miranda” de Santa Clara y del grupo de IA de la UCLV, este último con una amplia experiencia en el desarrollo de aplicaciones médicas.

Por otra parte, Weka ha sido objeto de estudio de los tesiantes a lo largo de su carrera y por ello han adquirido los conocimientos y las habilidades necesarias para enfrentar esta tarea.

## **Contenido**

En el capítulo 1 se analizará el concepto de sistema pronóstico y se mostrarán algunos de sus principales exponentes. Se analizarán los principales SBC teniendo en cuenta ventajas y desventajas de los mismos, prestando especial atención a los sistemas híbridos. Se explicará en qué

consiste la herramienta de aprendizaje automatizado Weka enfatizando en los tipos de datos que ella maneja. Finalmente se pondrán ejemplos de aplicaciones de los SBC en el campo de las Ciencias Médicas. En el capítulo 2 se propone la modelación computacional, se explica el funcionamiento del resolvidor de problemas implementado y se hace una comparación con otros enfoques alternativos. Por último se hace una descripción de la implementación del Sistema Automatizado Predictor de Riesgo Infantil de Mortalidad (Saprim). En el Capítulo 3 se presenta el manual de usuario de Saprim, así como una evaluación y validación de los resultados obtenidos con el software.

# Capítulo 1

## La Inteligencia Artificial en la Solución de Problemas Médicos.

### 1.1. Sistemas pronósticos

Los niños que necesitan cuidados intensivos suelen estar gravemente enfermos como resultado de un accidente, una enfermedad aguda o una intervención quirúrgica que pone en peligro su vida. Con tratamiento apropiado, la mayoría podrá restablecerse y llevar una vida normal y potencialmente productiva(Rovira, 2001).

Es una necesidad, en cualquier centro hospitalario, establecer prioridades entre los pacientes en base a determinadas características, en especial el nivel de gravedad o riesgo de fallecer. La adecuada valoración de la gravedad tiene gran influencia en la aplicación del tratamiento, ya que implica tomar las medidas necesarias en el momento apropiado, así como la concentración de los recursos materiales y humanos disponibles. Con el fin de asistir al especialista, en esta tarea surgen los sistemas pronósticos(Rovira, 2001).

#### **Concepto**

Es un sistema compuesto por un conjunto de variables, que da como resultado una puntuación predictiva de la mortalidad del paciente, o sea, del riesgo de fallecer. Para construir estas puntuaciones pueden ser usados varios tipos de variables, entre ellas variables clínicas, fisiológicas y de laboratorio(Lacroix/Cotting, 2005).

### **Funcionamiento**

Como se Los puntos asignados a cada variable son proporcionales a su peso para predecir severidad de la enfermedad y riesgo de fallecer. Un ejemplo de dato clínico es la frecuencia cardíaca,

Los pacientes críticos se caracterizan por presentar una alteración del equilibrio corporal. Estas alteraciones se pueden estimar mediante la medición de cuán alejadas se encuentran una o más variables fisiológicas de su rango normal. Muchas puntuaciones o sistemas pronósticos han sido desarrollados a partir de dichas variables(Lacroix/Cotting, 2005).

El funcionamiento de las puntuaciones predictivas se basa en la evaluación de una expresión donde están contenidas variables clínicas y de laboratorio, las cuales describen el estado del paciente. A estos parámetros se les adjudica una puntuación y su contribución al valor final deriva de su importancia específica en relación a la mortalidad. Puntuaciones más elevadas implican más gravedad y por tanto mayor riesgo de fallecimiento(Rovira, 2001).

Se deberá seleccionar los pacientes de cuidados intensivos de mayores probabilidades de recuperación, clasificándolo según las alteraciones fisiológicas o por las acciones que se realizan en los pacientes. Las variables fisiológicas empleadas para conformar los modelos pronósticos son de 4 tipos: datos clínicos, como es el caso de la frecuencia cardíaca, datos fisiológicos como el índice cardíaco, datos de laboratorio como el  $P_aO_2$  y otros parámetros como la escala de coma de GLASGOW.

### **Ejemplos**

Entre los sistemas creados específicamente para los pacientes atendidos en los Servicios de Terapia Intensiva se encuentran: el PSI (Physiologic Stability Index) que constituyó un paso importante en el campo pediátrico, el PRISM (Pediatric Risk of Mortality Score)(Pollack/Getson, 1988) y el PIM (Pediatric Index of Mortality)(Shann/Slater, 1997; Slater/Pearson, 2003). El PRISM constituye una modificación del PSI donde se logró reducir las 24 variables de este último a 14, disminuyendo así el tiempo necesario para su ejecución en cada paciente. Desde la publicación del PRISM, el PSI ha quedado desplazado en lo que se refiere a predicción del riesgo de mortalidad, pero puede usarse con fines descriptivos como medida directa de la severidad de una enfermedad. El PIM y el PRISM constituyen hoy unas de las mejores herramientas para estimar la gravedad en pacientes pediátricos graves.

Los parámetros del PRISM se miden durante las primeras 24 horas de permanencia en la Uni-

dades de Cuidados Intensivos Pediátricos (UCIP) y al aplicar esta escala se toman los peores valores del período considerado, obteniéndose finalmente una puntuación como medida de la severidad de la enfermedad. De las variables presentes en el PRISM al menos 5 de ellas son de laboratorio lo cual trae como consecuencia retraso en la evaluación del paciente(Rovira, 2001).

El PIM, con 10 variables, se usa con el mismo propósito que el PRISM, y comparte con este algunos parámetros predictivos. Sobre cual tiene un mejor desempeño las opiniones de los especialistas varían(Lacroix/Cotting, 2005).

Estos sistemas pronósticos tienen sus limitaciones ya que, para valorizar algunas variables, se hace necesario la ejecución de exámenes de laboratorio que son de costosa ejecución y traumáticos para el paciente. La espera que se genera en la obtención de los resultados de estas pruebas trae consigo un retraso considerable en la emisión del pronóstico. Tanto el PRISM como el PIM han sido validados sólo en algunos países(Lacroix/Cotting, 2005).

## **1.2. Programa Inteligente**

El curso que sigue el desarrollo del mundo actual apunta cada vez más a la interacción entre diversas ciencias. El auge de la computación ha propiciado que esté presente en muchas de las esferas de la sociedad, dando solución a disímiles problemáticas. Las ciencias computacionales han abierto nuevos horizontes en el campo de la medicina permitiendo el desarrollo de nuevas líneas de investigación y el perfeccionamiento de las ya existentes. Entre las ramas de la computación aplicadas a la solución de problemas médicos, la IA se destaca por su efectividad.

La Inteligencia Artificial (IA) es una rama de la computación que se encarga de hacer que la máquina simule ciertas capacidades físicas y mentales de los humanos, es decir, se dedica a la creación de software y hardware que imita el pensamiento humano(García, 2000).

Según Schildt, un “programa inteligente” es uno que muestra un comportamiento similar al humano cuando se enfrenta a un problema. No es necesario que el programa resuelva realmente el problema de la misma forma que el hombre.

Alan Turing se expresa análogamente al expresar que “si durante el intercambio entre una computadora y el usuario, este último cree que está intercambiado con otro humano, entonces se dice que el programa es inteligente”.

Para Forsyth la IA se relaciona con problemas que han escapado de una caracterización matemática.

La IA proporciona técnicas para enfrentar dos clases de problemas: los que por su dimensión hacen inaplicable un algoritmo conocido y los que carecen de algoritmo para resolverlos (Rich, 1994). El poder enfrentar estos problemas hace de la IA un complemento de la computación tradicional. Dentro de los campos de aplicación de la IA se encuentran los Sistemas Basados en el Conocimiento.

### **1.2.1. Sistemas Basados en el Conocimiento (SBC)**

Los Sistemas Basados en el Conocimiento se desarrollaron para dar solución adecuada a problemas orientados a aplicaciones, cosa que los métodos de solución a problemas generales no podían hacer, ya que resultaban insuficientes (Bello, 2002).

El conocimiento representado en los SBC se obtiene a partir de los expertos en el dominio. Un experto es una persona que tiene habilidades que provienen, generalmente, de un conocimiento detallado de los problemas que maneja y de la experiencia adquirida en su desarrollo vital. El conocimiento, en cualquier especialidad, puede ser de dos clases: público o privado. Los expertos humanos poseen conocimiento público (definiciones, hechos y teorías publicadas), pero usualmente tienen además conocimiento privado, el cual resulta de suma utilidad en la solución de los problemas de un dominio específico (Bello, 2002).

#### **1.2.1.1. Definición y estructura de un SBC**

Un Sistema Basado en el Conocimiento según (Bello, 2002) se puede definir como sigue: “Un sistema computarizado que usa conocimiento sobre un dominio para arribar a una solución de un problema de ese dominio. Esta solución es esencialmente la misma que la obtenida por una persona experimentada en el dominio del problema cuando se enfrenta al mismo problema”.

También se puede definir un SBC como : “Un sistema informático que simula el proceso de aprendizaje, de memorización, de razonamiento, de comunicación y de acción de un experto humano en una determinada rama de la ciencia, suministrando de esta forma, un consultor que puede sustituirle con unas ciertas garantías de éxito” (García, 2006).

Sin embargo esto no basta para definir correctamente un SBC, ya que este se diferencia de los programas basados en búsqueda general en aspectos tales como:

- La separación del conocimiento de cómo este es usado (distinción entre conocimiento y estrategia de control).
- El uso de conocimiento muy específico del dominio.
- El conocimiento empleado tiene naturaleza heurística en lugar de algorítmica.
- No se requiere analizar completitud.
- Pueden dar múltiples soluciones

Los Sistemas Basados en el Conocimiento están compuestos por la base de conocimiento (BC), un mecanismo de inferencia (MI), una interfaz de usuario (IU) y módulos opcionales: módulo explicativo (ME), módulo de cálculo de certidumbre (MCC) y módulo de autoaprendizaje (MA)(García, 2003).

$$\text{SBC} = \text{BC} + \text{MI} + \text{IU} + [\text{ME}] + [\text{MCC}] + [\text{MA}]$$

### **Base de conocimiento**

La BC es la componente más importante del SBC, ya que la potencia de un SBC radica en el conocimiento que posee. En dependencia de la naturaleza del conocimiento este se divide en tres tipos: numérico, simbólico y mixto. Cada tipo de conocimiento tiene asociado un formato, el cual indica cómo el conocimiento se representa internamente. A este formato se denomina Forma de Representación del Conocimiento (FRC)(Lezcano).

Dentro de la representación numérica se encuentran las probabilidades, las frecuencias y los pesos de la Redes Neuronales Artificiales (RNA). La representación simbólica comprende entre otros:

- Reglas de producción: son la FRC más popular. Tienen la forma IF <condición> THEN <acción/conclusión>. La interpretación de una regla es que si el antecedente se puede satisfacer entonces se obtiene el consecuente.

- Frames (Marcos): es una estructura de datos compleja que contiene un agregado de información acerca de un objeto. La información almacenada en el frame se distribuye en diferentes campos llamados slots (aspectos), cada slot contiene la información sobre un atributo del objeto que se modela.
- Scripts: Los scripts son estructuras de datos complejas designadas para almacenar el conocimiento sobre una secuencia estereotipada de acciones. Este conocimiento le dice a los sujetos que intervienen en la secuencia de eventos lo que puede suceder en una situación, que evento sigue y que papel debe jugar cada quien en la actividad social que se describe.
- Redes semánticas: una red semántica consiste en puntos llamados nodos, conectados por enlaces llamados arcos que describen las relaciones entre los nodos. Los nodos representan objetos, conceptos o eventos. Los arcos pueden definirse de varias formas, dependiendo de la clase de conocimiento representado.
- Strip: es la generación de una secuencia de acciones o programa de acción para un agente. Las acciones cambian un estado del universo sobre el que se desarrolla la planificación a otro. En el formalismo strip las acciones se modelan en una forma similar a las reglas de producción, y estas cambian una descripción de un estado en otra(Bello, 2002).

Un ejemplo de conocimiento mixto lo constituyen los casos o ejemplos de un dominio de aplicación, estos contendrán datos numéricos, textos y otros.

### **Métodos de solución de problemas (mecanismo de inferencia)**

La búsqueda es una técnica de solución de problemas. La búsqueda de la IA difiere de la búsqueda convencional sobre estructuras de datos, esencialmente en que se realiza en un espacio problema, no en una pieza de dato particular. Se localiza un camino que conecte la descripción inicial del problema con una descripción del estado deseado para el mismo, es decir, el problema resuelto. Este camino representa los pasos para encontrar la solución(Bello, 1993).

La MI es el intérprete del conocimiento almacenado en la BC. Esta implementa algún método de solución de problemas con una dirección (forward o backward) de búsqueda dada.

### **Interfaz de usuario**

Permite al usuario interactuar con el sistema, recibe como entrada un problema y brinda una solución como respuesta.

### **Módulos opcionales**

El MA permite incorporar a la BC los nuevos casos resultantes de generalizaciones. El ME permite conocer por qué y cómo se obtuvo una solución determinada y el MCC manipula la incertidumbre, la cual aparece a causa de información o conocimiento impreciso, información incompleta o por conceptos o palabras que son inexactos principalmente.

Los SBC se diferencian en la Forma de Representación del Conocimiento (FRC), en el Método de Solución del Problema (MSP), en las fuentes de conocimiento y en la forma en que se explica una solución obtenida principalmente. La tabla con los principales SBC se encuentra en el Anexo (3.2.1.4).

#### **1.2.1.2. Sistemas Híbridos**

Un sistema híbrido de inteligencia artificial está formado por la integración de varios subsistemas inteligentes, cada uno de los cuales mantiene su propio lenguaje de representación y un mecanismo diferente de inferir soluciones.

En (Zadeh, 2001) se especifica que Soft Computing es “ una mezcla de distintos métodos que de una forma u otra cooperan desde sus fundamentos con el principal objetivo de aprovechar la tolerancia que conllevan la imprecisión y la incertidumbre, para conseguir manejabilidad, robustez y soluciones de bajo costo”. Es por ello que la tendencia actual es desarrollar sistemas híbridos, como una etapa superior en el desarrollo de los SBC, con la finalidad de obtener un producto que aproveche las ventajas de los enfoques que combina y minimice sus deficiencias (Sarabia, 2007). A continuación se relacionan algunos sistemas híbridos que se han desarrollado para resolver problemas de diferentes esferas:

- En (Kuncheva, 1997) se propone un modelo borroso basado en el método  $k$ -NN para resolver problemas de clasificación.
- Los SIB Mandani y Sugeno (Sugeno, 1983) combinados con las RNA han dado lugar a sistemas como : NEFCLASS, ANFIS, NEFPROX. (Nauck/Klawonn/ Kruse, 1997).
- En (Gonzalez, 2006) se implementa el modelo Fuzzy-SIAC ( Simple Implementation of the Interactive Activation and Competition) que es una implementación borrosa de la variante simplificada del modelo de RNA Activación Interactiva y Competencia utilizada en el modelo original. También se propone el modelo ConFuCiuS (Conecctionist Fuzzy

Case-Based System), que es un modelo conexionista borroso para desarrollar Sistemas Basados en Casos. Se define como una instancia del clasificador basado en los vecinos más cercanos, usando una función de distancia que utiliza los pesos del modelo Fuzzy-SIAC.

- En (Sarabia, 2007) se presentan dos modelos híbridos basados en similitud, cada uno de los cuales combina coherentemente las redes neuronales asociativas, el razonamiento basado en casos y la lógica borrosa; utilizando esquemas de razonamiento diferentes para dar solución a un nuevo problema a partir de un conjunto de instancias conocidas.
- En (Hullemeier, 2007) se refieren a la combinación del aprendizaje basado en instancias con la lógica borrosa.

### **1.2.1.3. Ventajas y Desventajas de los SBC**

Debido a sus características las ventajas de los Sistemas Basados en el Conocimiento se hacen ver en su:

- Disponibilidad: Se puede emplear un SBC en lugares donde el ser humano no puede llegar, además de hacerlo durante un tiempo ilimitado.
- Permanencia del conocimiento: El conocimiento no se pierde con la muerte del experto humano sino que trasciende a través del SBC.
- Fácil modificación (conocimiento explícito y accesible).
- Preservación de la experticidad (constituye una memoria institucional y poseen la capacidad para adquirir nuevo conocimiento y perfeccionar el que poseen).
- Solución de problemas que incluyen datos incompletos.
- Constituye un entrenador en el dominio de aplicación.
- Coste reducido
- Los SBC son fiables.
- Experiencia múltiple: La experiencia de varios especialistas puede estar disponible para trabajar simultáneamente en la solución de problemas.

No obstante existen algunas desventajas comparando a los SBC con el ser humano como pueden ser:

- Lenguaje natural: No pueden mantener una conversación informal.
- Experiencia Sensorial: La información se recibe mediante teclado y ratón. Aunque algunos poseen sistemas de tratamiento de imágenes y quizás de voz no se comparan con las posibilidades del ser humano.
- Las respuestas no siempre son correctas.
- Incapacidad de manejar conocimiento no estructurado.
- El conocimiento se encuentra limitado al dominio de experticidad.

Además, otros problemas desde el punto de vista de su construcción, son que su eficacia depende de la adquisición y representación del conocimiento en una forma abstracta efectiva, además de la traducción del conocimiento en términos que una máquina pueda procesar.

### 1.3. Lógica Borrosa

Los conjuntos clásicos tienen límites bien definidos, por ejemplo : sea P el conjunto de los números reales mayores que 1.80, el cual puede ser expresado matemáticamente de la siguiente forma:

$$P = \{x \mid x > 1,80\} \quad (1.1)$$

el límite de este conjunto no es ambiguo: si el número x es mayor que 1.80, x pertenece a P, en otro caso no pertenece. Este tipo de conjuntos son muy utilizados para diversas aplicaciones en la Matemática y la Computación, sin embargo no representa adecuadamente las impresiones propias de la vida, donde muchos fenómenos se describen de forma abstracta. Se puede representar una colección de personas altas, como aquellas que miden más de 1.80 m mediante la definición del conjunto P expresada en (1.1) , donde x representa la altura de una persona. Si seguimos la definición dicotómica de P entonces una persona que mide 1.81 m es alta, mientras que otra que mide 1.79 m no lo es; sin embargo intuitivamente esta conclusión es irrazonable. La lógica

borrosa permite tratar información imprecisa, como por ejemplo: estatura media, temperatura alta o mucha fuerza en términos de conjuntos borrosos.

### 1.3.1. Los conjuntos borrosos

**Definición 1:** Formalmente un conjunto borroso A definido sobre un universo X es un conjunto de pares de la forma  $(x, \mu_A(x))$  donde  $\mu_A(x)$  es llamada la función de pertenencia (FP) para el conjunto A, asignando a cada elemento de X, un grado de pertenencia en el intervalo  $[0, 1]$ . Al conjunto X se le llama universo de discurso y puede ser un espacio discreto o continuo.

### 1.3.2. Modelación de rasgos numéricos como variables lingüísticas

La teoría de los conjuntos borrosos nos permite utilizar un lenguaje más natural para referirnos a valores de un rasgo (Zadeh, 2000; Zadeh, 1983), es decir, nos permite modelar rasgos numéricos como variables lingüísticas y es precisamente el uso de términos lingüísticos el que posibilita que la situación o evento que se represente en los casos sea más clara y cercana a la realidad.

Cuando se modela un atributo numérico como variable lingüística, consideramos como universo el conjunto de valores que para él aparecen en la BC. Denótese como  $R_p = \{P_1, P_2, \dots, P/R_p\}$  el conjunto de términos lingüísticos definido, donde cada  $P_i$  representa un conjunto borroso y tiene por tanto asociado una función de pertenencia. Véase el siguiente ejemplo para aclarar lo antes planteado: el concepto temperatura corporal del paciente se puede modelar como variable lingüística, definiendo para ello tres términos lingüísticos: baja (“low”), media (“medium”) y alta (“high”), empleando una función de pertenencia trapezoidal.

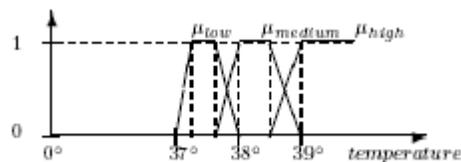


Figura 1.1: Variable lingüística que representa temperatura.

Al adoptar esta alternativa de modelación para rasgos numéricos las ventajas son inmediatas (Bello, 2000), ya que se tiene en qué medida un valor del dominio real del rasgo está próximo

a un valor representativo y si se presenta un problema de diagnóstico médico, es más natural hablar de que un paciente tiene una temperatura “alta”, que hablar de magnitudes numéricas próximas a 38 grados; así como establecer la relación de pacientes con temperatura “alta” y cierta enfermedad. Resulta conveniente en muchas tareas de diagnóstico, considerar intervalos discretos, debido a que los expertos usualmente tienen en cuenta solamente niveles de valores al realizar diagnósticos.

### **1.3.3. Los sistemas de inferencia borrosos**

Los sistemas de inferencia borrosos (SIB) tienen gran aplicación en diferentes campos como el control automático, la predicción, la robótica y el reconocimiento de patrones. La estructura básica de un SIB consta de 3 componentes conceptuales fundamentales (Shing, 1997):

- Base de reglas: contiene una selección de reglas borrosas.
- Base de datos: se definen las funciones de pertenencia usadas en las reglas borrosas.
- Mecanismo de razonamiento: describe el proceso de inferencia, donde a partir de las reglas y los hechos se deriva una conclusión.

El mecanismo de razonamiento de un SIB (ver figura (1.3.3)) se puede describir mediante los siguientes pasos:

- *Cálculo del grado de compatibilidad*: Evalúa los antecedentes de las reglas para determinar el grado de compatibilidad entre el hecho conocido y las expresiones modeladas como conjuntos borrosos.
- *Cálculo del grado de activación de la regla*: Combina los grados de compatibilidad obtenidos para cada antecedente.
- *Modificación del consecuente de la regla*: Afecta el valor del consecuente de la regla teniendo en cuenta el grado de activación de la misma.
- *Agregación*: Une los conjuntos resultantes en el consecuente de cada regla.
- *Defuzificación*: Propone una estrategia para a partir de un conjunto borroso encontrar un valor numérico.

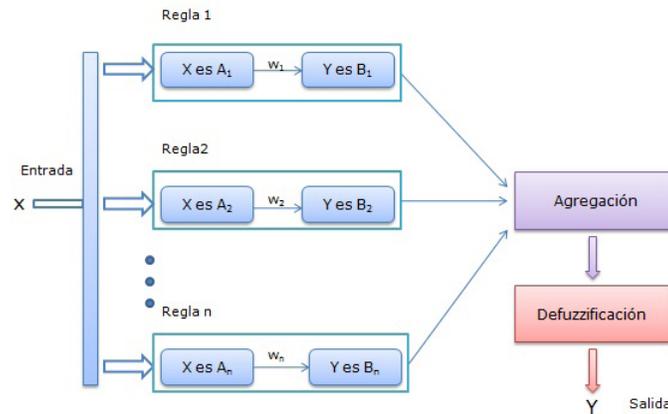


Figura 1.2: Mecanismo de razonamiento de un SIB.

Existen tres tipos fundamentales de sistemas de inferencia borrosos: el modelo de Mamdani, de Sugeno y de Tsukamoto. Estos SIB se diferencian en la forma del consecuente de las reglas borrosas (Nauck, 1997).

- El sistema borroso de Mamdani fue propuesto como un primer intento de controlar un artefacto de vapor, a partir de un grupo de reglas borrosas obtenidas de la experiencia de operadores humanos (Shing, 1997)(Shing/Jang, 1997); se caracteriza por tener el consecuente de cada regla representado mediante conjuntos borrosos. Para obtener el valor de salida (numérico) debe efectuarse algún método de defuzzificación. ( Ver Anexo 1.)
- En el modelo de Sugeno que fue propuesto por Takagi, Sugeno y Kang, una regla típica tiene la siguiente forma: Si  $x$  es  $A$  y  $y$  es  $B$  entonces  $z = f(x;y)$ , donde  $A$  y  $B$  son conjuntos borrosos y  $f$  es una función representada fundamentalmente por un polinomio. El resultado de cada regla es un valor duro; la agregación se realiza como una suma pesada de los valores obtenidos en las salidas parciales. ( Ver Anexo 2)
- El modelo de Tsukamoto es similar al de Mamdani, sin embargo las FP que se utilizan para modelar el consecuente de las reglas deben ser monótonas, el valor de salida de cada regla se obtiene directamente a partir de su grado de activación, y el resultado final como una suma pesada de los valores de salida de las reglas. (Ver Anexo 3.)

### 1.3.4. Heurísticas para determinar los parámetros de las funciones de pertenencia a un conjunto borroso

La teoría de los conjuntos difusos es una herramienta muy útil en la representación de conocimiento pues logra una caracterización matemática bastante acertada de términos del lenguaje natural.

Para obtener una variable lingüística se deben seleccionar los términos lingüísticos y la función de pertenencia correspondiente a cada uno de ellos, tarea que pudiera requerir conocimiento experto en el área de aplicación. Las funciones de pertenencia se pueden representar de forma explícita, gráfica y analítica. La forma explícita se representa por un conjunto de pares  $(x, \mu_A(x))$ . Las figuras 1.4 y 1.5 muestran las representaciones analítica y gráfica para las FP triangular y trapezoidal que son algunas de las más utilizadas para definir conjuntos borrosos.

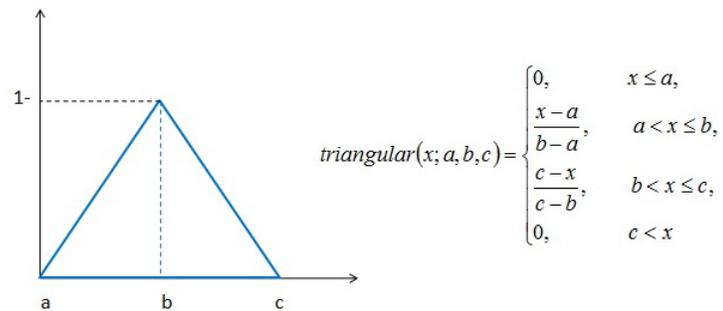


Figura 1.3: Representación analítica y gráfica de la función triangular.

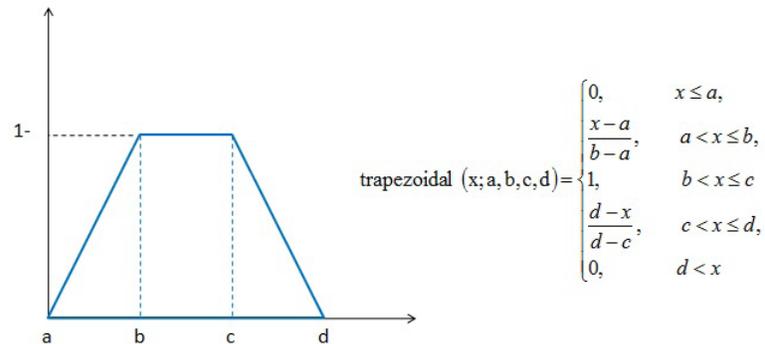


Figura 1.4: Representación analítica y gráfica de la función trapezoidal

Además del tipo de FP se deben definir la cantidad de términos lingüísticos para una variable; en (Pal, 2004) se asume como cantidad cinco, otros autores al referirse a la cantidad de intervalos en que debe discretizarse una variable numérica (conjuntos duros) asumen el valor máximo entre este número y la cantidad de clases definidas en el problema (Detyniecki, 2000). Muchos autores coinciden en que deben utilizarse menos de 7 conjuntos borrosos para representar las variables pues ese es el número máximo de niveles que puede reconocer el ser humano.

## 1.4. Aprendizaje Automatizado

El aprendizaje automatizado estudia como construir sistemas computacionales que se superen mediante la experiencia. En otros términos podemos concluir que un programa ha aprendido a desarrollar una tarea  $T$  si luego de proporcionarle una experiencia  $E$  el sistema es capaz de desempeñarse adecuadamente bien cuando se presentan nuevas situaciones de la tarea. El desempeño se mide empleando una métrica de calidad  $P$ ,  $E$  es generalmente un conjunto de ejemplos de  $T$  y tanto  $T$  como  $E$  y  $P$  deben estar correctamente especificadas en un problema de aprendizaje (Mitchell, 1997).

El aprendizaje automatizado se divide en dos áreas: el aprendizaje no supervisado y el aprendizaje supervisado. El aprendizaje no supervisado estudia los métodos en los que se aprende sin recibir información externa durante el proceso. Existen modelos de redes neuronales en las que el aprendizaje no requiere comparaciones entre la salida ideal y la obtenida, en lugar de ello se emplea algún criterio interno como minimizar el valor de la función definida para la salida de la

red, por lo puede ser considerado como un método no supervisado (Munakata, 2008). También las técnicas de agrupamiento pueden considerarse no supervisadas pues el criterio para ubicar elementos en una misma clase se basa en una observación de la similitud entre ellos.

En el aprendizaje supervisado, la calidad de los ejemplos para establecer las características más adecuadas es crucial. Existen dos tipos de problemas fundamentales dentro del aprendizaje supervisado: clasificación y regresión.

**Clasificación:** Dado un grupo de entradas se debe predecir la clase a la que pertenece ( la clase es una variable discreta). Muchos problemas utilizan 2 clases, por ejemplo, en el caso de los autos deportivos se podrían tener las clases: 0 para cuando el auto no es deportivo y 1 para cuando lo es.

**Regresión:** Dado un conjunto de entradas se predice un valor de salida numérico. Por ejemplo, dado el historial médico de una persona definir el valor de la hemoglobina.

### 1.4.1. Aprendizaje basado en instancias

Los métodos de Aprendizaje Basado en Instancias (IBL) incluyen la técnica de los vecinos más cercanos y la regresión localmente pesada. Generalmente se refieren a las técnicas de IBL como métodos de aprendizaje perezoso, pues se caracterizan por usar descripciones extensionales (las instancias o los casos) de los conceptos sin generar descripciones intencionales (como reglas); el paso de generalización es retrasado para la etapa de solución del problema (Bello, 2007). Según (Hullemeier, 2007) un algoritmo de IBL consiste en:

- *Función de similitud:* para calcular una medida numérica de la similitud entre dos instancias.
- *Función de clasificación:* estima la clase de la nueva instancia, dados los valores de similitud entre ella y los ejemplos almacenados.
- *Actualización del clasificador:* Después de cada clasificación debe decidirse si retener o remover una instancia, basados en los resultados obtenidos en clasificaciones previas.

### Concepto de similitud.

El concepto de similitud determina la distancia entre objetos, se aplica en múltiples áreas científicas; particularmente dentro de la Inteligencia Artificial se utiliza para resolver problemas de clasificación y regresión. Según (Sarabia, 2007) los conceptos de distancia ( $\Delta$ ) y similitud ( $\sigma$ ) son duales. La expresión de similitud en función de la distancia puede representarse matemáticamente como:

$$\sigma(x, y) = 1 - \Delta(x, y) \quad (1.2)$$

A continuación se relacionan algunas de las funciones de distancia más utilizadas a partir de las cuales pueden ser definidas funciones de similitud mediante la ecuación (1.2):

- Distancia Euclidiana

$$\Delta(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

- Distancia Euclidiana pesada

$$\Delta(x, y) = \sqrt{\sum_{i=1}^n w_i \cdot (x_i - y_i)^2}$$

- Distancia de Manhattan

$$\Delta(x, y) = \sum_{i=1}^n |x_i - y_i|$$

#### 1.4.1.1. Método de los $k$ vecinos más cercanos.

El principio de los  $k$ -vecinos más cercanos ( $k$ -NN) es el método más básico de IBL. Este algoritmo asume que todas las instancias corresponden a puntos en un espacio  $n$ -dimensional. El vecino más cercano a una instancia dada puede ser definido en términos de distancia euclidiana. Dada una instancia  $q$  se define el vector de rasgos  $[a_1(q), a_2(q), \dots, a_n(q)]$  donde  $a_r(q)$  denota el valor del  $r$ -ésimo atributo para la instancia  $q$ .

La distancia entre  $q$  y  $x_j$  puede definirse:

$$\Delta(q, x_j) = \sqrt{\sum_{i=1}^n (a_i(q) - a_i(x_j))^2} \quad (1.3)$$

La función de clasificación ( $f$ ) puede ser discreta o real en dependencia del tipo del atributo de decisión. En el caso discreto se determina la clase como la más frecuente entre las instancias seleccionadas. Si la variable de salida es numérica se calcula la media de los valores de salida de las  $k$  instancias más similares.

El método ha continuado perfeccionándose, hay variantes de distancia pesada donde se han utilizado diferentes funciones de distancia, incluyendo variantes modernas que incorporan el tratamiento a variables lingüísticas.

#### **1.4.1.2. Relación con SIB**

Es posible definir las propiedades de un método de IBL en términos de un Sistema Basado en Reglas Borrosas. Como las reglas de un SIB son consecuencia en muchos casos de una generalización a partir de casos (Bello, 2007), puede considerarse que las instancias juegan un papel similar a las reglas. En tal caso, los  $k$  vecinos más cercanos en el método se comportarían como reglas borrosas donde sólo el consecuente sería una variable lingüística y no representaría una etiqueta sino un valor de pertenencia a cada uno de los términos lingüísticos definidos. De aquí que las instancias en el método puedan interpretarse de forma similar a una *base de reglas borrosas* de un SIB.

### **1.4.2. Herramienta de aprendizaje automático Weka**

En el mundo actual, los SBC están siendo ampliamente empleados en la realización de sistemas capaces de dar solución a problemas de diferente índole: en la medicina, en la educación, en el comercio, etc. La herramienta de aprendizaje automatizado Weka contiene una colección de algoritmos de la IA, muchos de los cuales usan SBC para su implementación y resuelven disímiles problemáticas con un margen de error aceptable.

#### **1.4.2.1. Descripción general de Weka (Waikato Environment for Knowledge Analysis)**

Weka es una herramienta de aprendizaje automatizado. Fue Ian Witten, profesor del Departamento de Ciencias de la Computación de la Universidad de Waikato, Nueva Zelanda (1992) quien la creó. La primera versión publicada contaba con algunos algoritmos en C, posteriormente se decidió reescribir Weka en Java y a partir de la versión 3 ya Weka quedó escrita totalmente

en Java. Desde este momento se le comenzaron a hacer mejoras y se le agregaron nuevas facilidades, apareciendo así diferentes versiones. Weka superó los límites del lugar de creación y hoy día en varios lugares del mundo diferentes personas se esfuerzan en la ampliación y perfeccionamiento de esta herramienta.

La herramienta Weka es un ambiente de trabajo para la prueba y validación de algoritmos de la IA. Tiene implementada una colección de algoritmos conocidos, varias maneras para preprocesar los archivos de datos a utilizar por dichos algoritmos, así como facilidades para validar los mismos. Posee interfaces gráficas de usuario (GUI: Graphical User Interface) y cuenta con herramientas para realizar tareas de regresión, clasificación, agrupamiento, asociación y visualización (Matías/Araugo, 2006).

#### **1.4.2.2. Tipos de datos que Weka implementa**

Los tipos de datos que Weka acepta son:

- Entero (Integer): expresa números enteros.
- Numérico (Numeric): expresa números reales.
- Fecha (Date): expresa fechas.
- Cadena (String): expresa cadenas de texto, con las restricciones del tipo String.
- Enumerado: expresa entre llaves y separados por comas los posibles valores (caracteres o cadenas de caracteres) que puede tomar el atributo.

Como resultado de investigaciones en el Centro de Estudios Informáticos de la Universidad Central “Marta Abreu” de las Villas, se han incorporado a Weka dos tipos de datos nuevos:

- Lingüístico (Linguistic): expresa atributos numéricos como variables lingüísticas, es decir, trata datos borrosos.
- Conjunto (Set): expresa entre llaves y separados por comas los posibles valores (caracteres o cadenas de caracteres) que puede tomar el atributo, puede tomar más de un valor, ver González (2007).

Luego de una adecuada valoración de los tipos de datos que se presentan, se puede notar que el tipo de dato conjunto maneja los atributos multievaluados, lo cual resulta de notable ayuda en problemas donde existe esta necesidad, ya que permite una BC con una estructura más comprensible y natural.

## 1.5. Aplicaciones de los SBCasos a la medicina

La influencia de la computación en el desarrollo del pensamiento científico en medicina es notable, ya que se obliga al médico a buscar formas de descripción de los procesos biológicos, con la formalización matemática, más objetiva y precisa que la intuición. Por otra parte, el uso de la informática, herramienta del pensamiento médico, con la gran velocidad de desarrollo de los cálculos matemáticos, ofrece soluciones en el campo de probabilidades que hacen objetivizar realidades en los procesos biológicos, que por otros métodos quedarían inaparentes, o solo a la vista de los clínicos de mayor intuición o agudeza del llamado "ojo clínico".

Muchas son las aplicaciones de los SBC en el campo de las Ciencias Médicas. **EPILEP** es una base de conocimientos, que junto con la máquina de inferencias **ARIES** constituyen un Sistema Experto cuyo objetivo central es servir como consultante a médicos que se enfrenten al complejo diagnóstico de pacientes con ciertos ataques, para establecer si estos son de naturaleza epiléptica o no, así como todas las posibles acciones derivadas de ello. En general, el proyecto ha sido concebido con el objetivo de agrupar las experiencias de un colectivo de especialistas, con vistas a que sea capaz de ayudar en el diagnóstico diferencial con otros tipos de ataques, la clasificación de las crisis, la etiología y el tratamiento, mientras que, paralelamente sea capaz de explicar cuál es el fundamento de su trabajo y conclusiones, posibilitando la adquisición de nuevos conocimientos de forma rápida y sencilla (De la Cruz, 1993).

**EDIMED** se emplea en el diagnóstico de enfermedades pulmonares. Se consideraron catorce síntomas que son comunes en cinco enfermedades diferentes, dependiendo el diagnóstico del grado de incidencia con que cada síntoma se presente. Para cada síntoma (variable lingüística) se establecieron tres valores lingüísticos: bajo, medio y alto) y de acuerdo a la información de un experto (médico especialista) se generaron las tablas que conforman la base de datos que serán consultadas para obtener, mediante el empleo del álgebra difusa, varios diagnósticos con grados de pertenencia entre 0 y 1, siendo el diagnóstico con grado de pertenencia más cercano a 1 el más acertado (Urueta/Contreras, 2000).

**SHRINK** es un sistema basado en casos del tipo solucionador de problemas específico para el diagnóstico en psiquiatría. En este sistema se usa un caso previo para generar una hipótesis sobre el diagnóstico del nuevo caso. La estimación de la similitud se combina con el proceso de clasificación, y todos los casos similares se almacenan en la jerarquía bajo la misma clasificación. Si a pesar de esto, el caso recuperado no es útil, el sistema usa la clasificación secundaria basada en fallos. Un caso que se recupera por esta vía se asume por el sistema que tiene mayor similitud con el problema debido a que ambos en su desarrollo tuvieron un fallo similar en su comportamiento anterior (Rodríguez, 1996).

**BOLERO**, considerado un planificador basado en casos, combina realmente la planificación y el diagnóstico. Aparecen integrados en este producto un sistema basado en casos y un sistema basado en reglas, donde el razonador basado en casos actúa como el planificador del sistema basado en reglas, mientras que éste posee un escaso conocimiento sobre la estrategia de búsqueda a seguir. El sistema basado en casos construye los planes de acuerdo al estado de la memoria de trabajo del sistema basado en reglas, y la máquina de inferencia de este último ejecuta esos planes. Este modelo es muy adecuado para el diagnóstico médico, pues se caracteriza por utilizar información incompleta e imprecisa que hacen necesario poder alterar dinámicamente el plan de búsqueda trazado.

Como parte de un proyecto de investigación conjunta entre la Universidad Central de Las Villas y la Universidad de Oviedo, hace algunos años se creó la “Proyección del Centro de Desarrollo Electrónico hacia la Comunidad” (**PROCDEC**), cuyo objetivo principal es desarrollar un estudio de personas supuestamente normotensas, primero en la ciudad de Santa Clara y luego en toda la nación. El proyecto desarrolla un sistema basado en casos para el diagnóstico de Hipertensión Arterial. La base de casos consta de 14 rasgos predictores y un rasgo objetivo.

La maloclusión se refiere a cualquier grado de contacto irregular de los dientes del maxilar superior con los del maxilar inferior, lo que incluye sobremordidas, submordidas y mordidas cruzadas. Debido a la importancia que tiene la obtención de un diagnóstico certero en la especialidad de ortodoncia, específicamente en el síndrome de maloclusión, por la dificultad que presenta en ocasiones la diferenciación entre sus clases, se propone la implementación de un sistema basado en casos que cuenta con una base de casos clínicos almacenados según la clasificación sindrómica de Moyers y tiene como propósito fundamental el logro de un modelo factible para el apoyo en el diagnóstico y clasificación de este síndrome (1999).

**CASNET** se emplea en oftalmología. **INTERNIST** en medicina interna, **PIP** en afecciones

renales, **AI/RHEUM** en el diagnóstico en reumatología, **SPE** interpreta los resultados de electroforesis de las proteínas del suero producidas por instrumentos de análisis y **TIA** se usa en la terapia de ataques isquémicos (Trias, 1993).

**PROTOS** es un sistema basado en casos del tipo interpretativo para el diagnóstico de trastornos auditivos. Se diseñó de forma tal que los casos se usan para señalar la forma en que se razonaron experiencias previas para llegar a la predicción. En ese dominio de aplicación, muchos de los diagnósticos se manifiestan de forma similar y solo existen diferencias no sustanciales, a partir de la retroalimentación con el usuario. **PROTOS** aprende cuáles son esas diferencias. La similitud entre el problema y los casos recuperados se estima a partir de una explicación de cómo los rasgos brindan una evidencia equivalente para una clasificación. La similitud final se basa en la evaluación heurística de la calidad de la explicación anterior y en la importancia de los rasgos no similares. Sin embargo, sería aconsejable tener en cuenta la importancia de rasgos similares y no solo la importancia de los rasgos no similares. La base de casos se organiza por clases. **PROTOS** primeramente elige la clase correspondiente con el nuevo problema y luego el caso más similar de esa clase. Aprende a clasificar irregularidades de la audición a partir de descripciones de rasgos en términos de síntomas de los pacientes, historia y resultados de pruebas a las cuales se somete al mismo, almacenadas con 200 casos secuenciales (en 24 categorías) (Sánchez, 2006).

**El Sistema para el Pronóstico de Malformaciones Cardiovasculares en Recién Nacidos** es un SBCasos interpretativo. A partir de características de los padres pronostica la presencia de malformaciones cardiovasculares en el recién nacido y la especificación de las mismas en caso afirmativo. El sistema brinda una justificación de la solución (Rodríguez, 1996).

## Conclusiones Parciales

En este capítulo se hizo un recuento de algunos de los principales sistemas pronósticos, se realizó una explicación de conceptos y temas de necesario conocimiento para la comprensión del problema a resolver. Se analizaron los SBC y se mostró la factibilidad e importancia de la aplicación de modelos híbridos a problemas como el que nos ocupa. Se muestran conceptos requeridos para la comprensión de los sistemas basados en una lógica borrosa. Además, se explicó brevemente la herramienta de aprendizaje automatizado Weka y los tipos de datos que ella maneja, dejando bien claro que esta ofrece funcionalidades para trabajar con un tipo de dato para el tratamiento de los atributos multievaluados y lingüísticos. Se demostró, a través de ejemplos concretos, que

muchas son las aplicaciones de los SBC en el campo de las Ciencias Médicas.

## Capítulo 2

# Implementación de Saprim versión 2.0

Elevar la calidad de la atención médica constituye una de las directivas específicas de la estrategia que se reitera en la política del ministerio de Salud Pública de nuestro país, dentro del cual, la atención pediátrica reviste gran importancia y prioridad. La acción del médico ante su paciente incluye, además de llegar al diagnóstico, emitir un pronóstico sobre la posible evolución de la enfermedad y su nivel de gravedad. Atendiendo a esto, se orienta la sala de destino si el paciente es tributario de ingreso. Existen tres escalas de atención: convencional, cuidados intermedios y cuidados intensivos; los procedimientos diagnósticos y/o terapéuticos son diferentes en cada servicio.

En particular, se requiere decidir de forma adecuada y rápida si el paciente debe ingresar en las Unidades de Cuidados Intensivos (UCI), donde se atienden aquellos niños que tienen un estado de salud delicado y con mayor riesgo de fallecer. Los especialistas de este servicio en el Hospital Pediátrico “José Luis Miranda” de Santa Clara, no están satisfechos con los modelos extranjeros propuestos (PSI y PRIMs). El PSI realiza una valoración tomando los valores de 34 variables que se obtienen a partir de datos clínicos y complementarios del paciente; resultando un procedimiento muy engorroso, que incluye exámenes de laboratorio, que no se realizan en nuestro país, o son de muy costosa ejecución y traumáticos para el paciente, que retardarían además, la definición de una conducta inicial a seguir. La aplicación del PRIMs, aunque es una simplificación del PSI que utiliza solamente 14 variables, contempla al menos cinco exámenes de laboratorio. La realización de algunos de estos exámenes no es factible en nuestro país temporal o permanentemente por limitaciones materiales. El PRIMs es una herramienta válida únicamente en Europa y Norteamérica por lo cual no se garantizan resultados satisfactorios al

utilizarla en otras latitudes (Aguirre, 2002).

Un estudio realizado en esta UCIP arrojó que al evaluar el PRISM en un grupo de pacientes que posteriormente fallecieron, solo el 17,5 % fueron dados como alto riesgo de fallecer. Los restantes pacientes fueron ubicados en la escala de bajo y moderado riesgo de mortalidad. Podemos concluir que el PRISM no tuvo un buen desempeño en la predicción de la gravedad en esta UCIP siendo esta una grave deficiencia a tener en cuenta (Aguirre, 2002).

Es por ello que se requiere la creación de un modelo pronóstico propio que ayude a la toma de decisiones en la gestión hospitalaria para asistir al especialista en determinar la necesidad o no del ingreso de un paciente en la sala de cuidados intensivos. Para ello se valoran solamente los datos clínicos y epidemiológicos del paciente, mientras se realizan los exámenes de laboratorio para complementar la valoración del estado actual del mismo.

## **2.1. Análisis y diseño del sistema desarrollado**

El sistema se modeló mediante el uso del *Unified Modeling Language* (UML), el cual es un lenguaje visual estándar muy utilizado para la modelación y desarrollo de software. Las funcionalidades del software se comprenden mejor cuando se identifican los casos de uso del negocio y sus actores. Esto permite describir cómo se usa el sistema desde el punto de vista del usuario, además de proporcionar una mejor comprensión de los requisitos que debe cumplir el software.

### **2.1.1. Modelo del negocio**

Se efectuaron varias sesiones de trabajos con los expertos (médicos o especialistas), en las que se hizo un análisis detallado de su desempeño en el momento de la llegada de un paciente al hospital. Como resultado se identificaron los procesos claves. Se confeccionó un diagrama de actividad en el que se visualiza el flujo de dichos procesos, describiendo así el funcionamiento del negocio, ver (2.1.1).

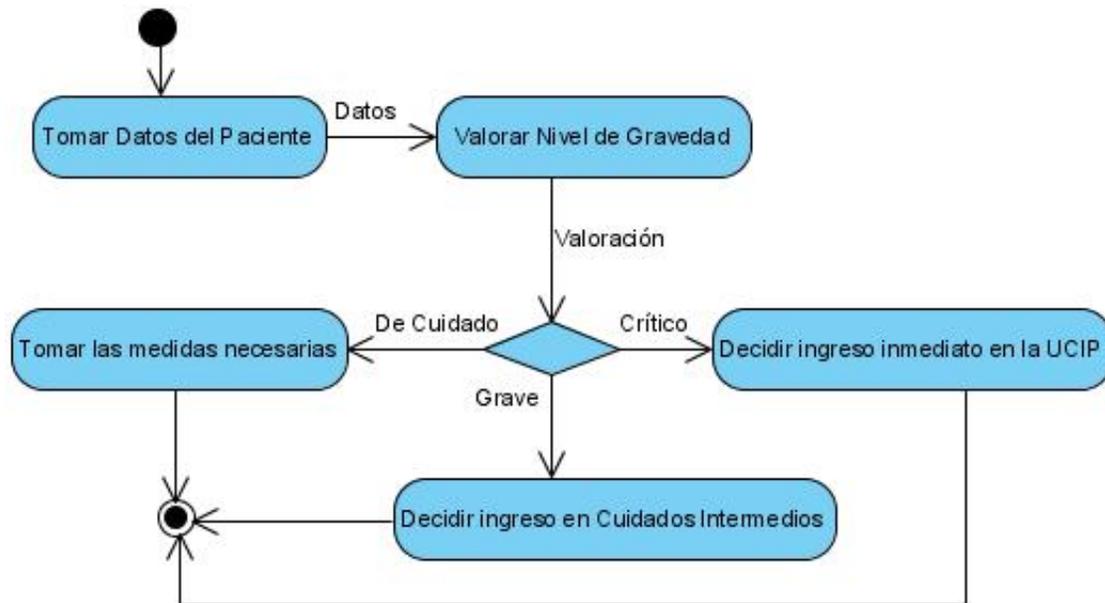


Figura 2.1: Diagrama de actividades.

### 2.1.2. Análisis de actores y casos de uso del negocio

En la solución de este problema se definen tres actores, cada uno con sus respectivos casos de usos. El primero de ellos se denomina “paciente” y es el encargado de suministrar los datos al pediatra. El segundo se denomina “usuario pediatra” y es el actor que interactúa con el software realizando la operación básica del mismo, la cual consiste en arribar a un pronóstico. El tercero se denomina “usuario experto”, que hereda las funciones del usuario pediatra y además, le es posible realizar operaciones de administración como son editar la base de conocimientos y modificar la contraseña. En la figura (2.1.2) se muestran los actores y casos de usos encontrados durante el diseño y la forma en que estos están relacionados.

Los casos de usos se describen a continuación:

- **Suministrar datos:** un usuario paciente debe indicar los síntomas al pediatra para que este los introduzca en el sistema.
- **Actualizar pediatras:** cualquier usuario puede adicionar , modificar o eliminar los nombres de los pediatras registrados en la base del sistema. Para adicionar solo tienen que seleccionar la opción “Nuevo Pediatra” en la ventana principal.

- **Obtener la estimación del nivel de gravedad:** un usuario pediatra, al seleccionar su nombre en la ventana principal del sistema, inicia la ventana de pronóstico y realiza las acciones de selección de datos, que serán procesados para la obtención de un nivel de gravedad resultante. En la figura (2.1.2) se muestra la expansión de este caso de uso “Estimar Nivel de Gravedad”, el cual incluye varias operaciones a realizar por parte del usuario para lograr el resultado.
- **Actualizar la base de conocimiento:** Al seleccionar la opción “Actualizar Base” se puede añadir, eliminar o modificar datos. En la figura (2.1.2) se muestra la expansión del caso de uso “Actualizar base de conocimiento”, ya que este incluye una serie de acciones.
- **Cambiar contraseña:** Este caso de uso comienza cuando el usuario experto selecciona la opción de “Cambiar contraseña” en la ventana principal o en el menú de la ventana de diagnóstico.

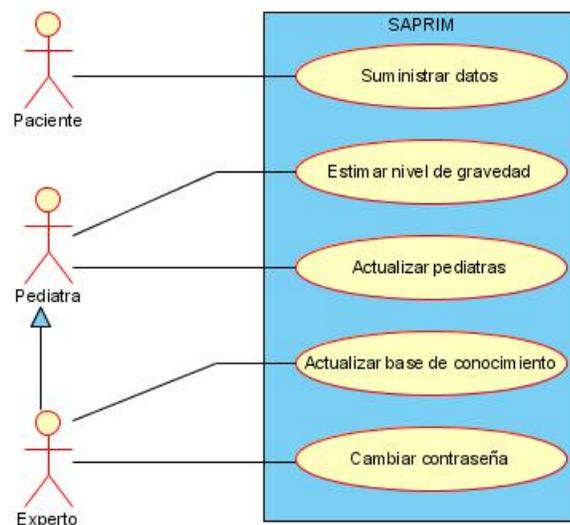


Figura 2.2: Diagrama de Casos de Uso

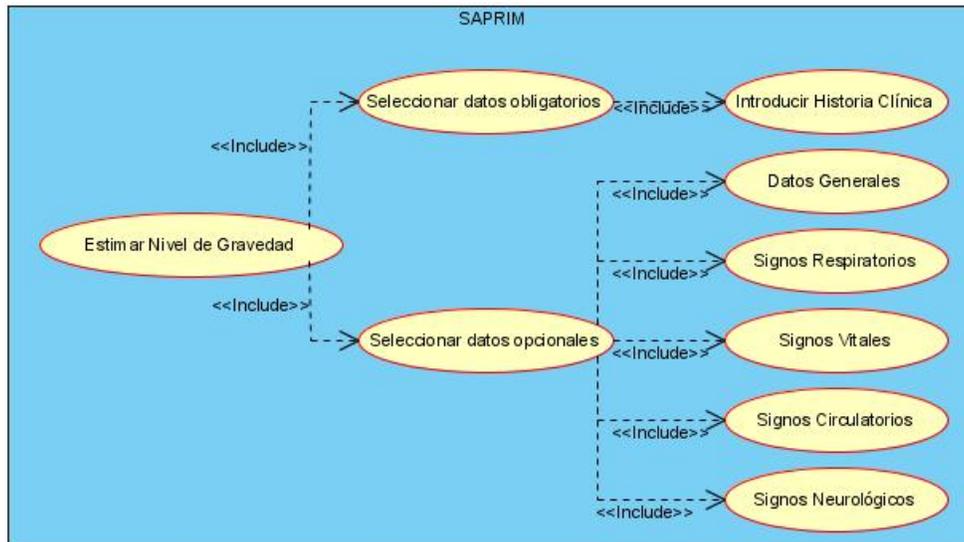


Figura 2.3: Expansión del caso de uso "Estimar Nivel de Gravedad"

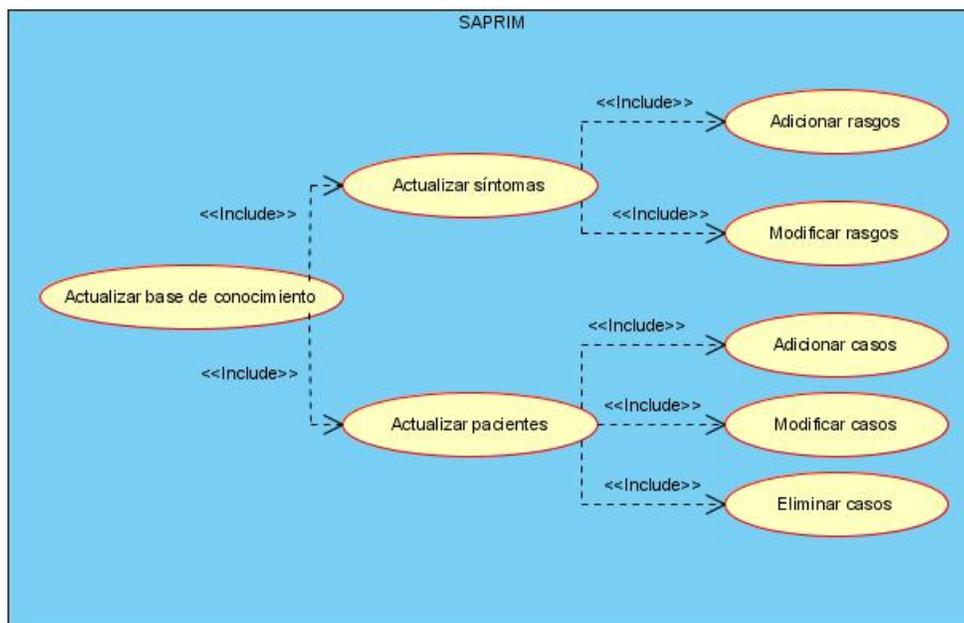


Figura 2.4: Expansión del caso de uso "Actualizar base de conocimiento"

### 2.1.3. Componentes del sistema

Para la realización de la interfaz gráfica se hizo uso del lenguaje Java con su paradigma de la programación orientada a objetos. El ambiente de programación utilizado es el NetBeans en su versión 6.8, que se considera un IDE muy popular entre la comunidad de programadores de Java.

El software está formado por tres componentes básicas, las cuales interactúan entre sí para lograr su funcionalidad. Se utilizó software libre en la mayoría de las componentes que conforman al SBC. La primera de las componentes es la base de conocimientos del sistema, resultado de la experiencia de los especialistas, que constituye una forma de representación basada en casos. En el archivo Infantil.arff se encuentra acumulada la información de los pacientes.

La segunda componente es el ejecutable Weka.jar, que contiene los algoritmos utilizados como mecanismo de inferencia del sistema y constituye el eje central de la aplicación.

El ejecutable del software, de nombre Saprim.jar, constituye la tercera componente del sistema. Se encuentra implementado en lenguaje Java y proporciona la interfaz gráfica de usuario, el mecanismo de comunicación con el paquete Weka, así como las formas de protección y manipulación de datos.

En la figura (2.1.3) se muestra la relación entre los distintos componentes del sistema.

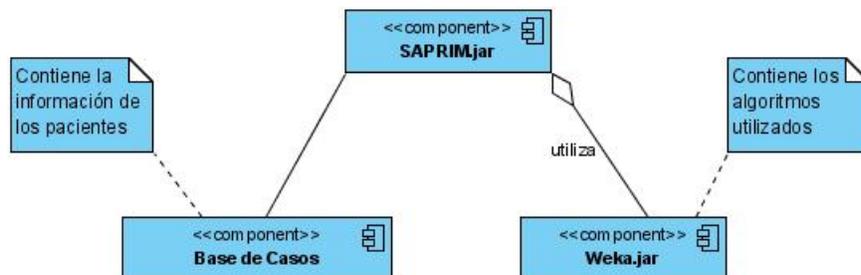


Figura 2.5: Componentes del sistema.

## 2.2. Modelación computacional del problema

Modelar un problema computacionalmente es fundamental, ya que al hacerlo estamos estableciendo las bases teóricas requeridas para lograr resultados satisfactorios. En la modelación se

establece cuál será el método de solución o el enfoque resolvidor del problema, lo que sirve de guía durante la implementación del producto de software final.

### 2.2.1. Construcción de la base de casos

La solución de este problema requiere de los conocimientos teóricos adquiridos por los especialistas y, esencialmente, de la experiencia acumulada con la práctica. Se cuenta con las historias clínicas de los pacientes que han sido ingresados en la UCI del Hospital Pediátrico “José Luis Miranda” desde el año 1998, y en otras salas del mismo, donde es común la ausencia de algunos datos. Un grupo de expertos ha trabajado desde el punto de vista médico esta hipótesis, y como resultado de estas investigaciones se tienen definidas un conjunto de características del paciente, que se deben considerar para decidir si el caso que se presenta requiere o no un ingreso en la sala de cuidados intensivos.

Para captar los datos necesarios de cada paciente, se empleó un modelo como el que se muestra en el Anexo (3.2.1.4) y finalmente se seleccionaron un total de 31 rasgos, de los cuales aparece un resumen en la tabla (2.2.1).

Tipo	Cantidad	Ejemplo	Dominio
Simbólico	22	Sangramiento	Sí, No
Cadena	1	No. Historia Clínica	6 dígitos
Numérico	3	Edad (meses)	0-192
Multievaluado	5	Color de la piel	Ictero, Pálida,...

Cuadro 2.1: Resumen de rasgos

Dos de los rasgos multievaluados presentan un dominio muy amplio, llegando incluso a tener uno de ellos 100 valores posibles. Como consecuencia de ello, se hizo necesaria la utilización del tipo de dato conjunto (“Set”), con el objetivo de facilitar el manejo y comprensión de los atributos multievaluados.

A partir de la muestra recopilada, y con la ayuda de los especialistas, se define una base de casos con 1019 ejemplos (529 “de cuidado”, 352 “grave” y 138 “crítico”). Cada caso se caracteriza por 31 rasgos, uno de los cuales se refiere al rasgo objetivo (clase), definiendo como sus valores posibles: “De Cuidado”, “Grave” y “Crítico”, donde la tercera clasificación significa que los

pacientes deben ser ingresados en la UCIP, la segunda indica que deben ser ingresados en la Unidad de Cuidados Intermedios y en los casos restantes el paciente no debe ser ingresado.

El desbalance presente en la base de casos se debe a que en la vida real aparecen muchos más casos no tributarios de ingreso que los que sí lo son, y de los que ingresan aparecen muchos más de pacientes graves que críticos. Se realizó una estimación de la diferencia entre la cantidad de pacientes de cada clase que aparecen en un año en la sala de urgencias del Hospital Infantil y resultó una proporción de 5:3:1 para pacientes “De cuidado”, “Grave” y “Crítico” respectivamente.

### 2.2.2. Modelo FuzzyIBk

Una vez correctamente definida la estructura de la BC y con la seguridad de estar en presencia de un problema con aprendizaje supervisado, donde las instancias tienen algunos rasgos sin valorizar (lo que resulta usual al tomar datos de historias clínicas), nos resta implementar el resolutor de problemas a utilizar.

Se cuenta con una BC cuyos casos presentan el rasgo objetivo de tipo nominal (clase) con los posibles valores “de Cuidado”, “Grave” o “Crítico” y se pretende obtener la medida en que el paciente pertenece a cada uno de estos valores. Para dar solución a este problema de obtener un valor numérico a partir de una variable nominal, se pretende modelar el rasgo objetivo como una variable lingüística.

El modelo FuzzyIBk se concibe como una variante similar al *k-NN*, que utiliza como elemento novedoso la modelación del rasgo objetivo como una variable lingüística. A continuación se describen sus principales características.

- Se tienen almacenadas instancias de entrenamiento que se representan como pares de atributo - valor.
- Se modela de forma interna el rasgo objetivo utilizando conjuntos borrosos, lo que permite manejar el grado de pertenencia de un valor numérico a cada uno de los términos lingüísticos definidos. Modelar el rasgo objetivo de esta forma y no externamente permite aprovechar las utilidades para la validación del algoritmo, que ofrece la herramienta Weka, para la cual se implementó FuzzyIBk.
- Dado un problema se seleccionan los *k* casos más similares a él.

- Para comparar dos instancias se utiliza una función de similitud basada en la distancia entre ellas.
- El valor de salida se calcula a partir de un proceso de defuzificación.

### 2.2.2.1. Obtención de la región borrosa

#### Función de similitud ( $\sigma$ ).

Como se explicó en el capítulo 1, uno de los elementos que contribuye al buen funcionamiento de los métodos de IBL es la función de similitud utilizada. Existen muchas propuestas que incluyen variantes basadas en la distancia y basadas en la estimación de probabilidades (Sarabia, 2007). El modelo no define ninguna función de similitud en específico, sin embargo permite utilizar cualquier función de distancia que se haya incorporado al ambiente de aprendizaje automatizado Weka, teniendo en cuenta que se debe convertir en una medida de similitud mediante la ecuación (1.2). Actualmente, la mayor parte de las funciones de distancia implementadas en Weka sólo pueden ser utilizadas para resolver problemas de clasificación sin tener en cuenta la presencia de rasgos predictores de tipo conjunto, la distancia HEOMJaccard trabaja con este tipo de dato, ofreciendo buenos resultados y será la variante a utilizar para probar el desempeño de FuzzyIBk. También existen implementadas una serie de funciones de distancia que permiten el trabajo con el tipo de dato conjunto. Se tomará  $\sigma_i = \sigma(x_i, x)$ .

El modelo propuesto se basa en la filosofía de los vecinos más cercanos. Dada una función de similitud  $\sigma$  se obtiene un conjunto  $K \subseteq U$  que contiene  $k$  instancias más similares al problema  $x$ , los elementos de  $K$  deben cumplir que:  $\forall (x_i, y_i) \in K, \nexists (x_j, y_j) \in U - K : \sigma(x_j, x) > \sigma(x_i, x)$  (ver figura 2.2.2.1).

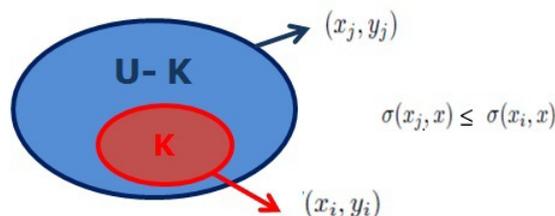


Figura 2.6: Definición de los  $k$  elementos más cercanos a  $x$

### Definición de los conjuntos borrosos

La variable lingüística se define por un número finito de conjuntos borrosos que se pueden construir utilizando funciones de pertenencia de tipo: triangular, trapezoidal o gaussiana. Para calcular cada una de estas funciones se realiza un proceso de agregación donde se obtienen los puntos de corte  $[a_1, a_2, \dots, a_n]$  de cada conjunto borroso. Se pueden seleccionar los valores de  $a_1$  y  $a_n$  siempre teniendo en cuenta que  $a_1 < a_n$ . Estos puntos se suelen denominar como MIN y MAX respectivamente.

En el modelo propuesto se implementa un método para la definición de las FP a partir del rasgo objetivo de tipo nominal. El número de valores que tenga este rasgo será el mismo número de conjunto borrosos que tenga la variable lingüística resultante. Este método crea conjuntos borrosos de igual tamaño.

#### 2.2.2.2. Proceso de defuzificación

El proceso de defuzificación es de vital importancia para el modelo propuesto, pues la solución al problema depende de la calidad de la interpretación de la región borrosa resultante. Al realizar el proceso de agregación se obtuvo el vector  $[a_1, a_2, \dots, a_n]$ , cada una de sus componentes representa un punto de corte, que define una región borrosa  $R = \{R_1 \cup R_2 \cup \dots \cup R_n\}$ , tal que  $R_j$  se obtiene de interceptar  $D_j$  con la recta  $f(x) = a_j$  y se cumple  $R_j(d_i) = \min(a_j, D_j(d_i))$ .

En FuzzyIBk se implementó la estrategia de defuzificación que a continuación se muestra:

- Centro de gravedad: Esta técnica encuentra el punto “balance” de la región, es por ello que es el empleado más frecuentemente, su expresión de cálculo es:

$$z = \frac{\int \mu_R(z) \cdot z \cdot d_z}{\int \mu_R(z) \cdot d_z} \quad (2.1)$$

En (Siler, 2005) se propone una expresión matemática similar a (2.1) para calcular el centro de gravedad, que se obtiene a partir de la representación explícita de una FP, y permite calcular de forma más sencilla y rápida el valor de  $z$ , además mantiene propiedades similares a la expresión (2.1), por todo ello esta variante del centro de gravedad, cuya expresión de cálculo se muestra a continuación, es la que se implementa en FuzzyIBk:

$$z = \frac{\sum_i \mu_R(z_i) \cdot z_i}{\sum_i \mu_R(z_i)} \quad (2.2)$$

En el modelo propuesto se utilizan 15 puntos, tal que la distancia entre dos puntos adyacentes cualesquiera es siempre la misma. El valor de cada punto se calcula mediante la siguiente expresión:

$$z_i = \begin{cases} \text{mín} & i = 0 \\ \text{máx} & i = 15 \\ z_0 + \frac{\text{máx} - \text{mín}}{15} \cdot i & e.o.c \end{cases} \quad (2.3)$$

### 2.2.2.3. Etapa de fuzificación

La definición de las FP, asociadas a los conjuntos borrosos definidos para el rasgo objetivo, puede influir en la eficiencia del modelo. Weka no tiene implementada ninguna variante de filtro que convierta un rasgo nominal en una variable lingüística. Por tal motivo, se desarrolla el filtro `NominalToLinguistic`, que basa su técnica en transformar la variable objetivo en tantos intervalos del mismo tamaño, como valores tenga su dominio y construir funciones simétricas, tal que el punto de máxima pertenencia a un conjunto borroso se encuentra en el centro del intervalo. Se desea que el solapamiento entre dos conjuntos borrosos adyacentes no exceda la cuarta parte de cualquiera de estas regiones. Si bien es un método muy simple, pues no requiere del conocimiento de expertos, por lo general no refleja las características de la variable que se está modelando. `FuzzyIBk` utiliza este filtro para construir las particiones borrosas, sin embargo se proponen nuevas variantes que puedan contribuir a obtener mejores soluciones.

#### **NominalToLinguistic**

Los pasos que intervienen en la fase de fuzificación son los siguientes:

Opciones de entrada:

- Seleccionar el número de vecinos más cercanos a encontrar ( $k$ ).
- Seleccionar una función de distancia.

- Seleccionar la función de pertenencia (FP) para particionar los conjuntos borrosos.
- Seleccionar los valores máximo y mínimo que va a tomar la variable lingüística.

Dado el conjunto de preprocesamiento completo hacer:

1. Calcular los puntos de corte de cada conjunto borroso. Se utiliza la FP seleccionada, así como los valores máximo y mínimo introducidos por el usuario. Se divide el universo de la variable en intervalos de igual ancho.

Dada una instancia de preprocesamiento  $x_t$  hacer:

1. Encontrar los  $k$  vecinos más cercanos  $x_j, j=1 \dots k$  de la instancia dada a partir de la función de distancia seleccionada.
2. Calcular los coeficientes de pertenencia de la instancia  $x_t$  a cada clase  $C_j$  mediante la siguiente relación:

$$D_i(x_j) = \frac{K_j^{(i)}}{K_F}, \quad K_j^{(i)} = \text{card} \left\{ x_n^{(j)} \mid x_n^{(j)} \in V_{K_F}(x_t) \right\} \quad (2.4)$$

donde  $K_F$  define el número de vecinos utilizado en la etapa de fuzificación, mientras que  $K_j^{(i)}$  representa el número de vecinos más cercanos de la instancia  $x_t$  que pertenecen a la clase  $C_j$ .

3. Realizar proceso de defuzificación a partir de los coeficientes previamente calculados.

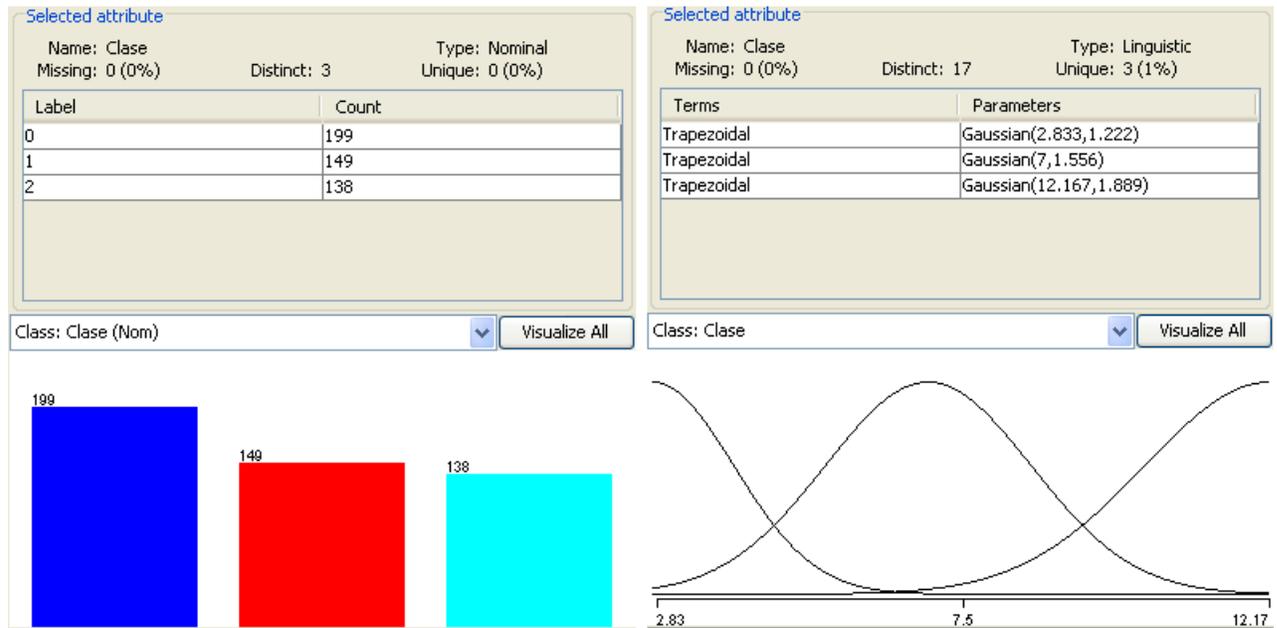


Figura 2.7: Conjunto de instancias antes y después de aplicarle NominalToLinguistic

#### 2.2.2.4. Etapa de clasificación

La clasificación comprende el cálculo de los valores de la función de membresía de un vector de prueba desconocido y la asignación de una clase a este vector.

#### FuzzyIBk. El algoritmo.

El funcionamiento de FuzzyIBk puede resumirse en el siguiente algoritmo.

Opciones de entrada:

- Seleccionar el número de vecinos más cercanos a encontrar ( $k$ ).
- Seleccionar el parámetro de fuzificación ( $\delta$ ).
- Seleccionar una función de distancia.

Dado un problema  $x$  hacer:

1. Obtener los  $k$  vecinos más cercanos  $x_j, j=1 \dots k$  al problema dado, utilizando la función de distancia seleccionada.
2. Hallar la matriz de preprocesamiento  $L_D$  donde las filas  $i = 1 \dots n$  representan los conjuntos borrosos y las columnas  $j = 1 \dots k$  las  $k$  instancias seleccionadas para la inferencia; el elemento  $(D_i(x_j))$  representa el grado de membresía de la instancia  $j$  al conjunto borroso  $i$ .
3. Calcular el vector  $\sigma = [\sigma(x_1, x), \sigma(x_2, x), \dots, \sigma(x_k, x)]$  con los valores de la similitud (ver 1.2) entre  $x$  y cada una de los vecinos más cercanos.
4. Efectuar la operación  $A(L_D, \sigma) = [\alpha_1, \alpha_2, \dots, \alpha_n]$  para obtener la solución borrosa del problema, donde:

$$\alpha_j(x) = \frac{\sum_{t=1}^k D_j(x_t) / \|x - x_t\|^{2/\delta}}{\sum_{t=1}^k 1 / \|x - x_t\|^{2/\delta}} \quad (2.5)$$

donde el parámetro  $\delta$  se usa para controlar la magnitud efectiva de la distancia hacia los vecinos más cercanos y puede ser seleccionado a través de validación cruzada junto con el valor de  $k$ .

5. Una vez obtenidos los coeficientes de pertenencia de la instancia  $x$  para cada una de las clases, se toma una decisión basada en estos coeficientes de la siguiente manera:

$$\alpha_c(x) = \max_j \{ \alpha_j(x) \} \Rightarrow x \in C_c \quad (2.6)$$

La figura (2.2.2.4) muestra el funcionamiento del algoritmo.

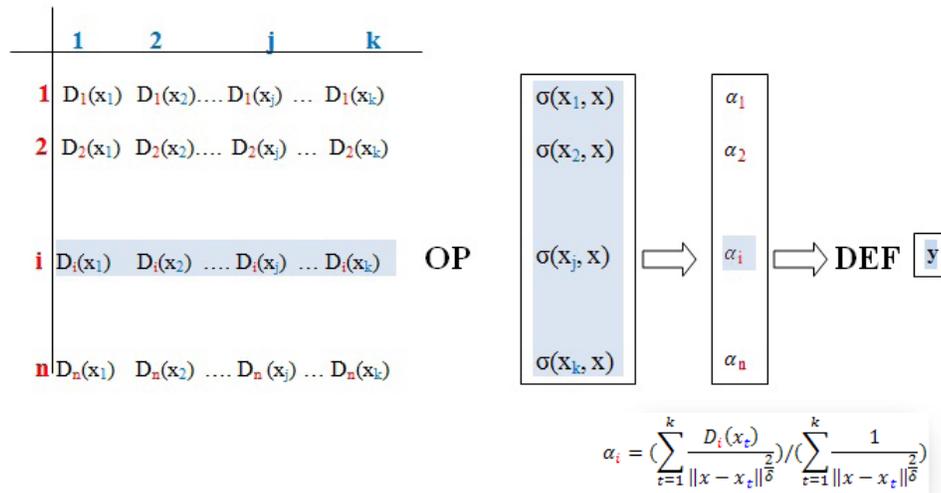


Figura 2.8: Funcionamiento del modelo FuzzyIBk

### 2.2.2.5. Algunas consideraciones sobre FuzzyIBk

El sistema de inferencia del modelo propuesto es muy similar al de un SIB; las propiedades de FuzzyIBk se pueden definir en términos de un Sistema Basado en Reglas Borrosas. Como las reglas de un SIB son consecuencia en muchos casos de una generalización a partir de casos (Bello, 2007), puede considerarse que las instancias juegan un papel similar a las reglas. En tal caso los *k vecinos más cercanos* en FuzzyIBk se comportan como reglas borrosas donde sólo el consecuente es una variable lingüística y no representa una etiqueta sino un valor de pertenencia a cada uno de los términos lingüísticos definidos. De aquí que las instancias en FuzzyIBk puedan interpretarse de forma similar a una *base de reglas borrosas* de un SIB. Además, el grado de cumplimiento del antecedente en una regla tiene su equivalente en FuzzyIBk y el *mecanismo de razonamiento* es muy similar al sistema de inferencia de Mamdani, donde se modela la implicación como conjunción.

El mecanismo de razonamiento de FuzzyIBk en términos de un SIB se puede describir mediante los siguientes pasos :

1. Determinar el *grado de compatibilidad* entre el hecho conocido y el problema a resolver mediante una función de similitud.

2. Se realiza una operación de *agregación* para combinar las salidas borrosas de las  $k$  instancias seleccionadas para la inferencia y obtener una solución borrosa.
3. Emplear una estrategia de *defuzificación* para encontrar un valor numérico a partir de una región borrosa.

Desde el punto de vista de un sistema basado en instancias el modelo se caracteriza por:

1. El uso de descripciones extensionales, en este caso las instancias sin generar reglas u otro tipo de descripciones intencionales.
2. Emplear una función para evaluar la *similitud* entre dos instancias.
3. Utilizar una *función de clasificación* que se define como la unión de la función  $A(L_D, \sigma)$  con el método de defuzificación a partir de los cuales se obtiene una clasificación numérica.

Mediante el modelo propuesto se pretende conseguir un nuevo modelo basado en instancias para la predicción numérica, utilizando un mecanismo de razonamiento propio de los sistemas borrosos.

### 2.2.3. Principales clases que componen el sistema

La aplicación queda diseñada como un sistema orientado a objetos cuyas entidades de mayor importancia lo constituyen las clases que definen los objetos del programa. Las relaciones entre los diferentes objetos se establecen a través de los atributos y métodos definidos en cada clase. En la figura (2.2.3) se muestra el diagrama de las principales clases correspondiente a la aplicación.

A continuación se explica detalladamente la implementación y funcionamiento de las clases del sistema que se muestran en la figura (2.2.3).

El paquete GUI contiene las clases correspondientes a la interfaz gráfica de usuario, las cuales se describen a continuación:

**SystemInit:** Clase encargada de realizar las acciones de inicialización del programa. Su función principal es organizar todos los chequeos de datos, cargar los datos iniciales que se mostrarán en la interfaz gráfica de usuario y mostrar la ventana principal del sistema.

**Security:** Clase encargada de realizar el establecimiento de contraseñas de administración y la comprobación de su validez.

La administración de contraseñas se ejecuta con los métodos siguientes: *savePassword (String password)*;

EL chequeo de la validez de una contraseña se realiza por el método: *public boolean isPasswordValid (String password)*;

**SplashScreen:** Clase encargada de mostrar una ventana de bienvenida al programa donde se muestra el logotipo. Da inicio al sistema y permanecerá mostrando la interfaz de bienvenida hasta que se finalice el proceso de inicialización.

**Welcome:** Esta clase constituye la ventana principal del sistema donde se muestran las distintas opciones.

**VentanaPrincipal:** Clase que constituye la guía durante el proceso de pronóstico del nivel de gravedad. Ofrece diferentes opciones, como Cambiar la contraseña de administración y acceder a la Ventana de administración.

**VentanaSecundaria:** Esta clase implementa la interfaz de administración del sistema y es accesible solo para los usuarios que se autentifiquen.

**DataSet:** Clase que permite la interacción con la biblioteca Weka.jar, facilitando la utilización del clasificador FuzzyIBk y el filtro NominalToLinguistic mediante la implementación local de un objeto que herede de estos.

Los métodos de mayor importancia en esta clase son:

- *init()*: Inicializa las variables necesarias para la inferencia.
- *classify()*: Realiza la inferencia.

**TDescripcionRasgos:** Objeto que almacena la descripción de los rasgos presentes en la base de conocimientos. Se tienen en cuenta la cantidad de rasgos, el tipo de cada uno y sus valores posibles. El tipo de cada rasgo se representa mediante una instancia de la clase TRasgo.

El método más importante de esta clase es:

- buildDescripcionRasgos(): Encargado de instanciar el objeto a partir del fichero DescripcionRasgos.txt

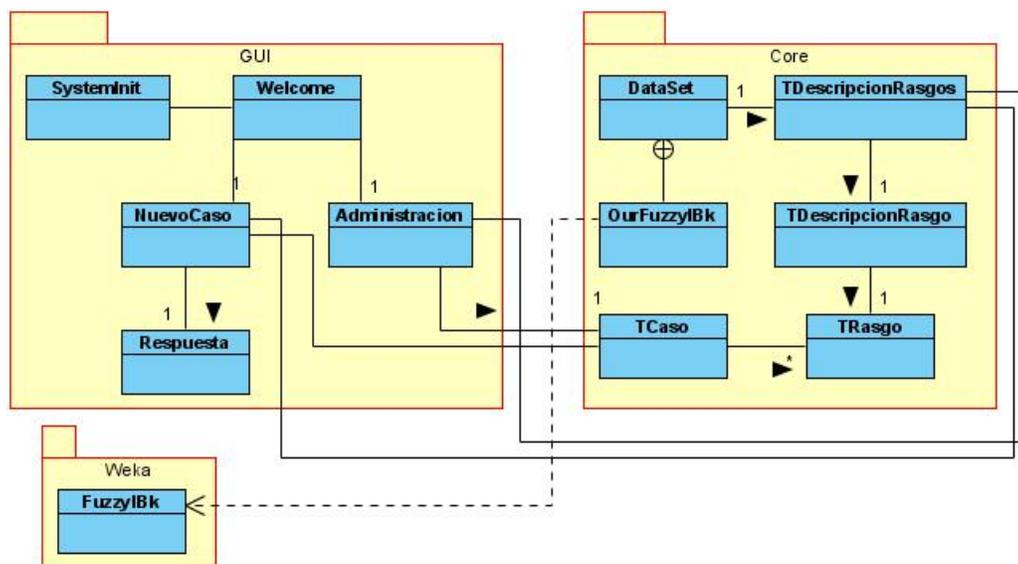


Figura 2.9: Diagrama de las clases principales del sistema.

### 2.3. Agregando funcionalidades a WEKA

Weka es un sistema multiplataforma y de amplio uso probado bajo sistemas operativos Linux, Windows y Macintosh. Puede ser usado desde la perspectiva de usuario mediante las seis interfaces que brinda. A través de la interfaz de preprocesamiento disponible en el modo *Explorer* es posible aplicar cualquiera de los algoritmos de preprocesamiento implementados en Weka, denominados filtros. Estos algoritmos transforman de alguna manera el conjunto de datos de entrada, modificando sus atributos o sus instancias. El filtro que se propone en el epígrafe 2.4 está disponible en dicha interfaz.

Weka se implementa en Java, lenguaje de alto nivel ampliamente difundido y multiplataforma, lo que posibilita que sistemas escritos en este lenguaje como puedan ejecutarse en cualquier plataforma sobre la que haya una máquina virtual de Java disponible. Esta herramienta sigue los preceptos del código abierto (open source), por lo su código fuente está totalmente disponible, permitiendo la modificación del mismo.

Las clases de Weka están organizadas en paquetes. Esta organización hace que añadir, eliminar o modificar elementos no sea una tarea compleja. Está formado por 10 paquetes globales, y dentro de ellos se agrupan otros paquetes que aunque su contenido se ajusta al paquete padre ayudan a organizar aun mejor la estructura de clases e interfaces. Los paquetes globales son: *associations*, *attribute selection*, *classifiers*, *clusterers*, *core*, *datagenerators*, *estimators*, *experiment*, *filters*, *gui*. Para implementar los modelos propuestos se utilizaron los paquetes *core*, *filters* y *classifiers*.

### 2.3.1. Agregando un nuevo filtro

Los filtros supervisados (*supervised*) transforman los atributos teniendo en cuenta la interdependencia entre el atributo clase y los valores de los demás atributos; mientras que los no supervisados (*unsupervised*) transforman sin considerar el atributo clase. Los filtros de atributos realizan el preprocesamiento en dirección a los rasgos del conjunto de datos, significando que los mismos hacen cambios en el número o definición de los atributos. Por otro lado, los filtros de instancias realizan un preprocesamiento orientado a las mismas, por lo que no afectan los atributos del conjunto de datos. Permiten realizar acciones como adicionar, eliminar o modificar instancias.

En (Gonzalez, 2006) se propone una metodología para agregar un filtro a Weka:

1. Crear la clase: situarla en el paquete que le corresponde de acuerdo a las características del algoritmo. El filtro *NominalToLinguistic* se encuentra en el paquete *Filter/unsupervised/attribute*.
2. Hacer uso obligatorio de los paquetes *weka.filters* y *weka.core*: estos paquetes son imprescindibles a la hora de crear un nuevo filtro. El primero de ellos contiene la clase *Filter* que define la estructura general de un esquema de filtrado. En el segundo existen clases importantes como *Instance* e *Instances*, las cuales se utilizan por todos los filtros para manipular los datos.

3. Especificar la clase de la que se hereda: todos los filtros deben tener como superclase la clase abstracta *Filter*, porque es en ella donde se encuentran los métodos que deben ser redefinidos por el nuevo algoritmo.
4. Especificar las interfaces a implementar: de acuerdo a la funcionalidad del filtro deben escogerse las interfaces necesarias. El filtro *NominalToLinguistic* implementan la interfaz *UnsupervisedFilter* y *OptionHandler*.
5. Definir la entrada de parámetros al filtro.
6. Crear los métodos necesarios de acuerdo a las particularidades del filtro.

El filtro propuesto debe ver todas las instancias antes de comenzar a preprocesarlas . Esto obliga a fijar el formato de salida de los datos en el método *batchFinished*, instanciando en el los atributos mediante la clase *LinguisticAttribute*, creando la función de pertenencia que usará cada uno de ellos y calculando además los parámetros que necesita cada función.

En la nueva clase del filtro se implementaron los métodos *buildGaussianTermSet*, *buildTrapezoidalTermSet* y *buildTriangularTermSet*, respectivos a las funciones de pertenencia Gaussiana, Trapezoidal y Triangular, los cuales se encargan de construir un vector de objetos de funciones de pertenencia con sus respectivos parámetros. Este vector es usado para llamar al constructor de la clase *LinguisticAttribute*. Las instancias del conjunto de datos, después de filtrarse, no sufren ningún cambio, colocándose en la cola de salida listas para ser recogidas.

### 2.3.2. Agregando un nuevo clasificador

Weka denomina clasificador a cualquier modelo con capacidad de predecir un valor nominal (clase discreta, clasificación) o un valor numérico (regresión). Los clasificadores son agrupados en diferentes paquetes de acuerdo a la técnica que empleen. Estos paquetes se encuentran dentro del paquete *weka.classifiers*. Ellos son:

- *bayes*: algoritmos de redes bayesianas.
- *functions*: algoritmos de redes neuronales y regresión numérica.
- *lazy*: algoritmos perezosos.

- *meta*: algoritmos meta-clasificadores, es decir, que usan como entrada un clasificador base.
- *misc*: algoritmos de clasificación que no están incluidos en ninguna otra categoría.
- *trees*: los algoritmos de árboles de decisión.
- *rules*: algoritmos que implementan reglas de asociación.

Para crear un nuevo clasificador en Weka se sigue la metodología propuesta en (Gonzalez, 2006):

1. Crear la clase: primeramente se crea una nueva clase con el nombre del clasificador a adicionar y se sitúa dentro de su paquete correspondiente de acuerdo al tipo de clasificador que se implementará, FuzzyIBk sigue el principio de los algoritmos perezosos por ello se colocó en el paquete *lazy*.
2. Hacer uso obligatorio de los paquetes *weka.classifiers* y *weka.core*. FuzzyIBk utiliza clases del paquete *core* para calcular la distancia entre instancias y para acceder a los métodos de la clase *LinguisticAttribute* y agregar métodos nuevos a esta clase. Además, este clasificador hace uso del paquete *Filters* para convertir de forma interna el atributo de decisión en una variable lingüística.
3. Especificar que se hereda de la clase *Classifier*.
4. Especificar las interfaces a implementar. FuzzyIBk implementa la interfaz *OptionHandler* para los parámetros de entrada.
5. Especificar la entrada de parámetros: implementar los métodos de la interfaz *OptionHandler* (*listOptions*, *setOptions*, *getOptions* ).
6. Especificar los métodos a implementar de la clase *Classifier*, de acuerdo a la funcionalidad del clasificador.
7. Nuevos métodos a crear.

## 2.4. Conclusiones Parciales

Se propone un modelo que combina el método de los k vecinos más cercanos del aprendizaje basado en instancias, con el mecanismo de razonamiento de los SIB, para resolver un problema de clasificación.

Se propone un nuevo filtro que permite extender los métodos para la definición de las FP, de esta forma puede ser comprobado el desempeño de FuzzyIBk con diferentes tipos de particiones borrosas.

Al lograr una integración entre las diversas tecnologías del lenguaje Java, se obtiene un software con una interfaz gráfica amigable, que posee un mecanismo de inferencias y una base de conocimiento que posibilitan la realización de pronósticos. Además, se cumplen los requisitos de la creación de sistemas para modificar dicha base de conocimiento y la protección de los datos, obteniéndose un software seguro, flexible y portable.

## Capítulo 3

# Manual de Usuario. Evaluación y Validación de los Resultados

En este capítulo se presenta el manual de usuario que explica todas las facilidades del sistema **Saprim**. Se hace también una validación y evaluación del mismo como garantía de su eficiencia y funcionalidad.

### 3.1. Manual de Usuario

#### 3.1.1. Introducción a Saprim versión 2.0

Saprim es un sistema que utiliza técnicas de la inteligencia artificial para determinar el nivel de gravedad de un paciente en el momento de su llegada al hospital. Posee una base de conocimientos primaria que es el resultado de años de experiencia de expertos en el campo de la medicina, aunque dicha base puede ser actualizada o modificada por aquellos expertos en la materia que estén autorizados por los expertos principales.

Saprim para Windows proporciona un sistema basado en casos que permite el diagnóstico del paciente a través de un entorno gráfico amigable, utilizando menús y cuadros de diálogo sencillos que realizan la mayor parte del trabajo. La mayoría de las tareas se pueden llevar a cabo simplemente seleccionando las opciones con el uso del ratón, ocultando los procesos que pueden resultar difíciles de entender para el usuario final.

Además de las facilidades de la interfaz gráfica, Saprim para Windows proporciona:

**Administrador de Bases de conocimiento:** Sistema que permite realizar modificaciones a la base de conocimiento de Saprim, ofreciendo las posibilidades de:

- Adicionar y modificar valores a los rasgos existentes en la base de conocimientos.
- Adicionar, modificar y eliminar pacientes a la base de conocimientos.
- Adicionar, modificar y eliminar usuarios pediatras que quedarán registrados al momento de atender un nuevo paciente.

**Sistema de seguridad:** Establece una contraseña de administración para restringir el acceso al Administrador de Bases de Conocimiento, de manera tal que, solo el administrador del sistema, en su rol de usuario experto en materia de diagnóstico, podrá modificar la base de conocimiento.

**Manual de Usuario:** Manual de usuario interactivo que permite realizar búsquedas de una o varias palabras entre los tópicos.

### 3.1.2. Facilidades del sistema

El sistema resulta comprensible para cualquier usuario sin importar cuán amplios sean sus conocimientos de computación. La interfaz de Saprim resulta muy cómoda y los términos que se manejan son del conocimiento del usuario (médico o especialista).

Saprim permite, dados los rasgos clínicos y epidemiológicos de un paciente, determinar el nivel de gravedad del mismo, tanto cuantitativamente, en una escala de 0 a 100, como cualitativamente, en los términos: de cuidado, grave o crítico. Para dar una explicación a la solución obtenida se muestran los casos más semejantes a este y se visualiza la medida en que cada rasgo influye en el pronóstico emitido en orden descendente. De los casos más similares encontrados se muestra el índice que ocupan en la base de casos para detectar inconsistencias que puedan estar afectando el pronóstico. El sistema brinda la posibilidad al especialista de administrar la base de conocimiento, permitiendo así la edición de inconsistencias o actualización de términos en los ejemplos almacenados previamente.

### 3.1.3. Instalación y requerimientos del software

Saprim se presenta como un instalador único el cual consiste en un fichero ejecutable que se encargará de configurar su computadora durante la instalación para desempeño satisfactorio. Para instalar el software solo tiene que:

1. Ejecutar el instalador “Saprim Installer.exe”.
2. Seguir los pasos indicados por el asistente de instalación.

Los requisitos del sistema son los siguientes:

- El software es compatible con el sistema operativo Windows Server/2000/XP/Vista/Seven.
- La instalación básica necesita alrededor de 60 megabytes de espacio disponible en disco, aunque pudiera requerir mas dependiendo del tamaño que pueda tener la base de conocimiento a medida que se vaya actualizando.
- 60 megabytes de memoria RAM mínimo.
- Ratón y teclado.

### 3.1.4. Interfaz Gráfica de Usuario

Saprim ofrece una interfaz gráfica de usuario amigable que oculta los procedimientos, dejando a la vista del usuario controles de fácil acceso, que permiten explotar el sistema con solo algunos clics. El software contiene cuatro ventanas esenciales:

- Ventana Principal
- Ventana de Pronóstico.
- Ventana de Administración de la Base de Conocimientos.
- Ventana de Resultados.

Además presenta un conjunto de diálogos que permiten informar al usuario el estado de la acción que está ejecutando.

### 3.1.4.1. Ventana principal

La ventana principal, ver figura 3.1.4.1, se muestra al iniciar el software y a través de ella es posible seleccionar la acción a realizar. En la lista desplegable “Seleccionar Pediatra” aparecen los posibles nombres de los especialistas que realizan las consultas y de los usuarios potenciales de este sistema. El usuario debe seleccionar su nombre en esta lista y si no se encuentra allí, es posible introducirlo mediante la selección del botón “Nuevo Pediatra”. Una vez realizada la acción anterior cualquier usuario puede entrar a “Clasificar Nuevo Caso”. La opción “Administrar Base” solicita una contraseña para acceder a la ventana de administración de la base de conocimiento, solo los expertos principales deben tener esta clave de seguridad, pues las modificaciones realizadas afectarían significativamente la eficacia del sistema.

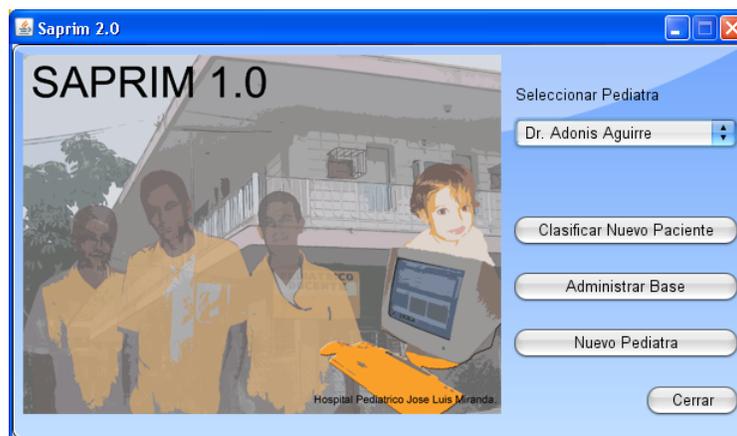


Figura 3.1: Ventana Principal

### 3.1.4.2. Ventana de pronóstico

La ventana de pronóstico, ver figura 3.1.4.2, se muestra al seleccionar la opción “Clasificar Nuevo Caso” en la ventana principal, y a través de ella es posible realizar la función esencial del software, que es determinar el estado del paciente. La ventana de pronóstico está formada por un conjunto de pestañas que permiten ir seleccionando los síntomas que presenta el paciente. Aunque estas pestañas funcionan como un asistente, no es necesaria la selección de los datos en el orden en que aparecen. También es opcional llenar cada campo para clasificar al paciente, pero siempre debe introducirse un valor en el campo de la Historia Clínica.

Cada pestaña consta de diferentes mecanismos de selección de datos, los cuales pueden ser:

- Introducción de datos en áreas de texto: El usuario escribe sobre estas áreas. Es imprescindible llenar el campo de Historia Clínica para la clasificación del paciente.
- Listas desplegables: La selección se lleva a cabo cuando se despliega una lista y se escoge uno de los valores existentes.
- Cuadros de selección múltiple: Para activar estos cuadros, se debe marcar el cuadro de selección “Evaluado”, luego el usuario puede seleccionar tantos valores como presente el paciente.

Cada pestaña representa una agrupación de rasgos relacionados para dar organización al usuario a la hora de seleccionar los datos, ver Anexo (3.2.1.4). Cada uno de estos rasgos consta de valores que se encuentran en la base de conocimientos y que, con solo autenticarse, los expertos pueden modificar en cualquier momento.

Pronóstico - Saprím 2.0

Dr. Adonis Aguirre

Datos Generales | Signos Respiratorios | Signos Vitales | Signos Circulatorios | Signos Neurológicos

Historia Clínica: 12345

Edad

Meses

Años

Modo de Ingreso: Remitido

Temperatura: Hipotermia

Estado de Hidratación: Deshidratado

Estado Nutricional: Desconocido

Color de la Piel

Evaluado

Normal

Palida

Cianosis

Terrosa

Ictero

Estado de Salud Previo

Evaluado

Sano

Adenoiditis\_cronica

Atopia

Anemia

Anplasia\_medular

Diagnóstico

Evaluado

Guillain\_Barre

Hemartrosis

Hematuria

Hemorragia\_suprarrenal

Hernia

Anterior Siguiente Clasificar Cancelar

Figura 3.2: Ventana de pronóstico.

### 3.1.4.3. Ventana de administración de la base de conocimiento

La ventana de administración, ver figura (3.1.4.3), permite al usuario experto modificar el contenido de la base de conocimiento y de esa manera mantener actualizado el software. Esta ventana es accesible solo por el usuario que conozca la clave de seguridad. Es recomendable que solo los usuarios capacitados realicen las operaciones de modificación para que no exista información errónea y así evitar respuestas indeseadas por parte del software en el momento de realizar un pronóstico.

Para acceder a la misma solo tiene que:

1. Seleccionar la opción “Actualizar Base de Conocimiento” en la ventana principal o en el menú “Administración” de la barra de menús de la ventana de pronóstico.
2. El sistema muestra un diálogo solicitando su autenticación como administrador. Introduzca la clave de administración y presione el botón “Aceptar”.
3. El sistema muestra un diálogo indicando si fue satisfactoria o no su autenticación. Presione el botón “Aceptar”.

La interfaz de administración de la base de conocimiento está formada por los siguientes componentes:

– Pestaña “Síntomas”:

- Navegador: Está ubicado en la parte izquierda de la ventana.
- Visor de Valores: Está ubicado en la parte derecha.
- Controles: Están ubicados en la parte derecha inferior. Lo conforman los botones de Añadir, Editar, Modificar.

– Pestaña “Casos”:

- Visor de casos: Se encuentra en el centro de la ventana y consta de tres pestañas que contienen las tablas con los pacientes divididos por su nivel de gravedad. Los casos se dividen en “De Cuidado”, “Graves” y “Críticos”.
- Controles: Están ubicados en la parte inferior. Lo conforman los botones de Adicionar, Modificar, Eliminar.

– Pestaña “Pediatras”:

- Visor de Nombres: Está ubicado en la parte izquierda.
- Controles: Están ubicados en la parte derecha. Lo conforman los botones Adicionar, Editar, Eliminar.

- Controles: Están ubicado en la parte inferior derecha. Lo conforman los botones Guardar, Cancelar.

En la figura 3.2 se muestra cada componente que conforma la Ventana de Administración.

El navegador posee una estructura de árbol a través de la cual se muestra el contenido de la base de conocimiento. Esta estructura organiza los rasgos de la misma manera que las pestañas en la ventana principal.

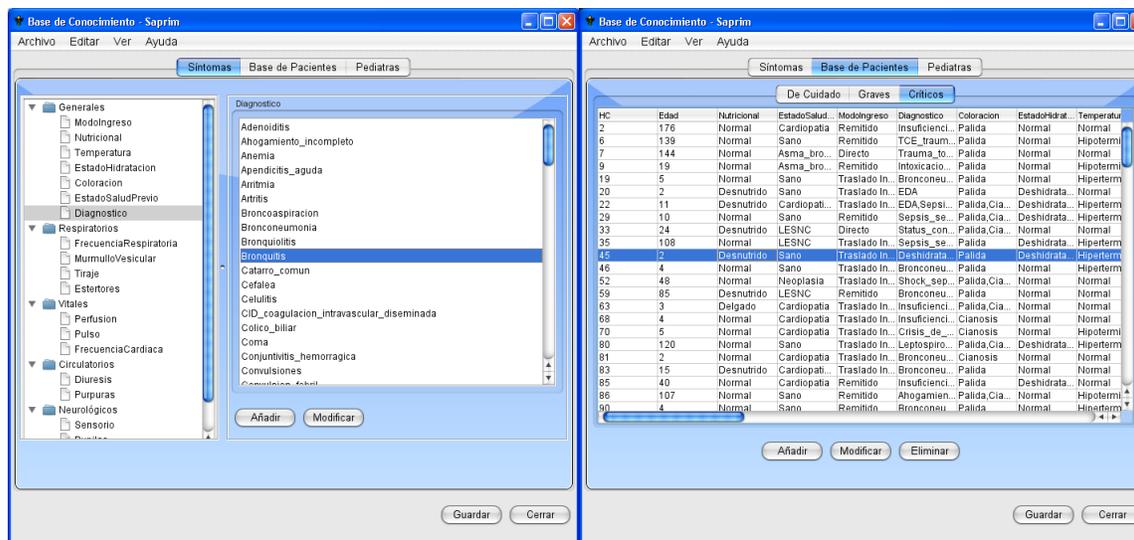


Figura 3.3: Ventana de administración

#### 3.1.4.4. Ventana de resultados

Luego de recopilada toda la información del caso problema sólo resta emitir el pronóstico, para ello debemos elegir “Clasificar” en la ventana de diagnóstico, ver figura 3.1.4.2, donde finalizó el proceso de recopilación de la información. Al seleccionar el botón “Clasificar” se muestra la ventana de la Figura 3.7, en la cual el usuario (médico o especialista) observará una escala de colores con la ubicación del paciente según su nivel de gravedad, este se toma como un valor entre 0 y 15. Además, el sistema ofrece, de acuerdo a este puntaje, el estado del paciente como uno de los valores “De Cuidado”, “Grave” o “Crítico”.

Esta ventana ofrece al usuario la posibilidad de observar cuáles fueron los casos más similares encontrados por el sistema. Además, al autenticarse, el usuario experto puede agregar el nuevo caso evaluado a la base de conocimientos del sistema, si así lo desea.

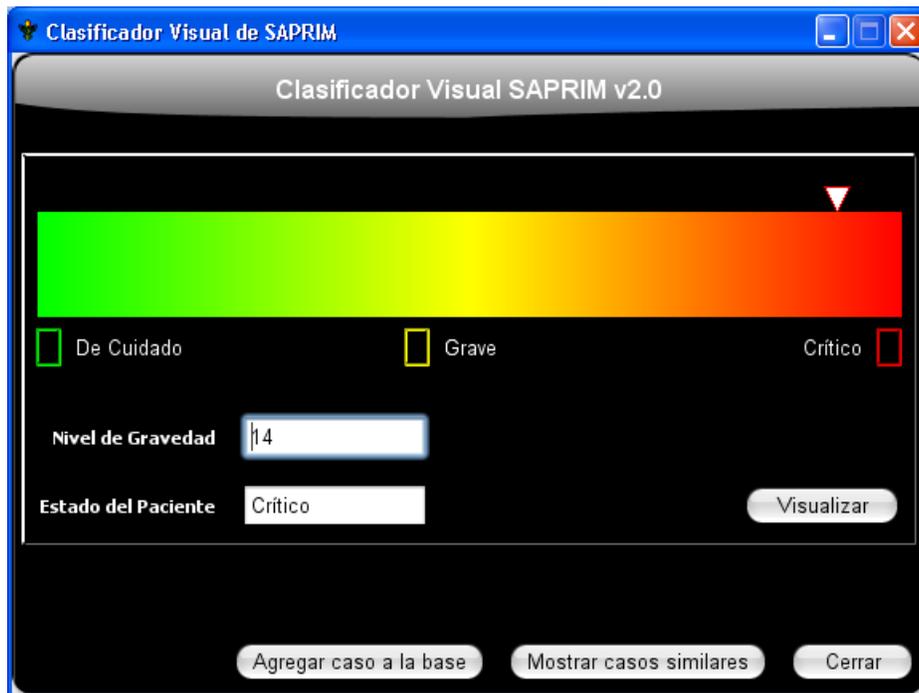


Figura 3.4: Ventana de clasificación visual.

#### 3.1.4.5. Menús

La mayoría de las opciones que ofrece Saprím son accesibles a través de las barras de menús dispuestas en cada ventana. Cada opción de menú se corresponde con el tipo de ventana en la que se encuentra.

La barra de menús de la ventana de pronóstico contiene 5 menús: Archivo, Edición, Administración, Ver y Ayuda.

El menú “Archivo” contiene opciones que permiten efectuar operaciones sobre la ventana actual, estas opciones son:

- “Nuevo Paciente”: Limpia todos los rasgos seleccionados y muestra la primera pestaña de la ventana de pronóstico.
- “Clasificar”: Se procesan los datos seleccionados para obtener el nivel del gravedad del paciente y se muestra la ventana de resultados.

- “Cerrar”: Esta opción cierra la ventana de pronóstico y muestra la ventana principal.

El menú “Edición” contiene opciones que permiten administrar la base de usuarios del sistema, estas opciones son:

- “Pediatras”: Esta opción muestra la ventana de usuarios, permitiendo la adición, modificación y eliminación de pediatras.

El menú “Administración” contiene opciones que permiten administrar el sistema y que requieren de autenticación, estas opciones son:

- “Cambiar contraseña”: Como su nombre indica permite cambiar la contraseña de administración.
- “Administrar Base de Conocimiento”: Esta opción muestra la ventana de administración de la base de conocimiento que permite realizar actualizaciones.

El menú “Ver” de la ventana de pronóstico ofrece opciones relacionadas con los datos almacenados en la base y el caso previamente clasificado:

- “Casos Semejantes”: Esta opción muestra una ventana con los casos más semejantes al paciente previamente clasificado, encontrados por el sistema. Por defecto se encuentra deshabilitada, una vez clasificado algún caso esta opción se habilita.
- “Distribución”: Muestra la ventana de distribución, ver figura (3.1.4.5).



Figura 3.5: Ventana de distribución.

El menú “Ayuda” está presente en todas las ventanas de Saprim y permite el acceso rápido al manual de usuario del sistema. Este contiene una sola opción:

- “Acerca de”: Muestra información general de Saprim
- “Tabla de contenidos”: Esta opción muestra la ventana de ayuda.

La barra de menús de la ventana de administración de la base de conocimiento presenta los menús “Archivo”, “Edición”, “Ver”, “Ayuda” que contienen las opciones correspondientes.

El menú “Archivo” contiene las siguientes opciones:

- “Guardar”: Una vez que haya realizado los cambios deseados en la base de conocimiento, seleccione esta opción para aplicarlos a la base de conocimiento de Saprim. Para ver los cambios aplicados debe reiniciar el sistema.
- “Cerrar”: Esta opción cierra la ventana de pronóstico y muestra la ventana principal.

El menú “Edición” cuenta con:

- “Nuevo Síntoma”: Permite adicionar valores a los rasgos existentes en la base de conocimientos de Saprim.
- “Nuevo Paciente”: Muestra una interfaz con todos los rasgos y sus valores correspondientes, para que el usuario experto seleccione cada uno y agregue el caso resultante a la base de conocimientos. Esto permite ampliar la experiencia del sistema y posibilita la obtención de mejores resultados frente a pacientes con características similares a las recién introducidas.
- “Nuevo Pediatra”: Permite agregar usuarios pediatras al sistema.

### **3.1.5. Pronóstico**

Realizar pronósticos es la tarea principal para la cual se construye Saprim. Este trabajo se lleva a cabo en la ventana de pronóstico y puede ser ejecutado por cualquier usuario.

#### **3.1.5.1. Estimar nivel de gravedad**

Es necesaria la selección precisa de los datos y características que ofrece Saprim para lograr así un resultado que solucione el problema que se presenta al usuario. Para realizar un pronóstico solo tiene que:

1. Introducir el valor de la historia clínica del nuevo paciente a evaluar.
2. Seleccionar los síntomas que presenta el paciente en las diversas pestañas que presenta Saprim. Se puede navegar a través de estas pestañas accediendo a ellas directamente o haciendo uso de los botones “Siguiente” y “Atrás”.
3. Una vez terminada la selección de rasgos, se presiona el botón “Clasificar”, que muestra la ventana de resultados con el nivel de gravedad y el estado del paciente.

### **3.1.5.2. Manipulación e interpretación de los resultados**

El resultado de una evaluación se muestra en la ventana de resultados. En esta se puede encontrar el nivel de gravedad como un valor numérico que puede tomar como mínimo 1 y como máximo 15. Este valor le indica al especialista si el paciente se encuentra en estado de cuidado, grave o crítico. Saprim ofrece también su valoración del estado del paciente como uno de esos tres valores, pero queda abierta la interpretación del especialista a partir del puntaje obtenido.

Si el sistema se encuentra con un paciente cuyo estado puede ser ambiguo, es decir, que muestra rasgos que lo hacen pertenecer tanto a un estado como a otro, le deja esta decisión al pediatra. Una vez que el especialista seleccione su valoración como uno de los dos valores posibles, se muestra la ventana de resultados con la ubicación del paciente en una escala de colores.

Saprim ofrece al experto la posibilidad de agregar el caso evaluado a la base de conocimientos del sistema, esto es posible mediante el botón “Agregar caso a la base”.

El pediatra puede observar los casos que el sistema encontró similares al evaluado mediante la selección del botón “Mostrar casos similares”. Esto muestra una ventana con diversas pestañas, cada una de ellas representa un caso, y en ellas se observan los valores que presenta cada uno y la similitud con nuestro paciente.

### **3.1.6. Actualización de la base de conocimientos**

La actualización de la base de conocimientos incluye los cambios que realiza dentro de ella un usuario experto. En el momento en que se decide modificar la base de conocimiento, Saprim crea una copia de la misma, sobre la cual el usuario experto efectuará los cambios que estime convenientes. De esta manera se evita que la base de conocimientos de Saprim sufra daños durante la modificación.

Una vez realizados los cambios deseados, se debe proceder a aplicarlos mediante la selección del botón “Guardar” y a continuación reiniciar Saprim.

## **3.2. Evaluación y validación del software**

Para lograr la validación del software y por consiguiente la aceptación del mismo, lo primero que hay que tener en cuenta es cuán certero resulta el pronóstico que se obtiene con el sistema

desarrollado. El porcentaje de error en la clasificación emitida por el sistema con respecto a la propuesta por el usuario (como experto del dominio) debe resultar mínimo, es decir, debe existir una coincidencia bastante fuerte entre ambos criterios de clasificación (González, 2007).

### 3.2.1. Resultados Experimentales

Los resultados experimentales se desarrollaron en el ambiente de aprendizaje automatizado Weka. La validación del modelo se dividió en cuatro momentos:

1. Analizar el desempeño del clasificador utilizando diferentes valores de  $k$  en el filtro NominalToLinguistic.
2. Comparar FuzzyIBk con el algoritmo IBk que es un método clásico de IBL.
3. Comparar FuzzyIBk con las diversas técnicas adaptadas para manipular datos de tipo conjunto.
4. Analizar el desempeño del modelo empleando la BC Infantil.

En los experimentos se usaron 8 bases de casos del *UCI Machine Learning Repository* y la base de casos empleada en el sistema Saprim 2.0. Para el modelo propuesto se probó con conjuntos borrosos representados mediante la misma función de pertenencia (triangular, trapezoidal o gaussiana). La función de distancia utilizada fue la HEOMJaccard, que permite el manejo de rasgos predictores de tipo conjunto.

IBk y FuzzyIBk varían su desempeño en dependencia de la cantidad de instancias que se utilicen para la inferencia, para evaluarlos se buscó en cada caso el valor  $k < 15$  óptimo al dividir la base de casos en 10 particiones de las cuales se usaron 9 para aprendizaje y una para entrenamiento. Una vez determinado este valor de  $k$ , se efectuó el test de comparación 10-fold cross-validation que realiza el proceso anterior 10 veces, rotando cada vez el conjunto de prueba.

Los experimentos para determinar: el mejor valor de  $k$  (tabla 3.2), el desempeño de FuzzyIBk en comparación con el IBk estándar (tabla 3.2.1.2), el comportamiento del nuevo clasificador en comparación con otros enfoques y el desempeño del modelo computacional obtenido, se realizaron mediante la prueba 10 times 10-fold cross-validation, disponible en la utilidad Experimenter de Weka. En todos los casos las comparaciones se refieren al error medio absoluto y al porcentaje de clasificación correcta, obtenidos al aplicar este test .

La tabla (3.2.1) muestra las descripciones de los conjuntos de datos que fueron utilizados para las pruebas.

	Conjunto de datos	No. instancias	No. numéricos	No. simbólicos	No. conjunto
1	contact-lenses	24	0	5	0
2	diabetes	768	8	1	0
3	glass	214	9	1	0
4	ionosphere	351	34	1	0
5	iris	154	4	1	0
6	labor	57	8	9	0
7	vote	435	17	0	0
8	weather	14	2	3	0
9	infantil	1019	3	22	5

Cuadro 3.1: Descripción de los conjuntos de datos.

### 3.2.1.1. Desempeño de NominalToLinguistic

La tablas siguiente muestra los resultados de aplicar FuzzyIBk empleando la función de distancia *HEOMJaccard* con diferentes valores de  $k$  a la base de casos de Saprim 2.0, utilizando FP traingulares, trapezoidales y gaussianas. El número de vecinos más cercanos utilizados en la fase de entrenamiento es independiente del seleccionado por el clasificador a través de validación cruzada, por lo que deben realizarse pruebas para obtener el valor de  $k$  óptimo.

k	FP		
	Triangular	Trapezoidal	Gaussiana
1	0.069	0.066	0.065
2	0.084	0.070	0.074
3	0.098	0.067	0.082
4	0.099	0.064	0.084
5	0.106	0.068	0.085

Cuadro 3.2: FuzzyIBk después de preprocesar los datos con NominalToLinguistic.

Para comparar estos 3 grupos de muestras relacionadas se realizó un test de Friedman. En la figura se muestran los resultados.

	Mean Rank
Función Triangular	3.00
Función Trapezoidal	1.20
Función Gaussiana	1.80

Figura 3.6: Resultados de ranqueo entre las tres funciones de pertenencia

Se desea el  $k$  y la FP que menor error produzcan; la tabla de rangos muestra que con  $k=4$  y FP trapezoidal se obtienen mejores resultados. La siguiente figura muestra que con una significación de 0.008 se puede decir que entre estos grupos existen diferencias significativas.

N			5
Chi-Square			8.400
df			2
Asymp. Sig.			.015
Monte Carlo	Sig.		.008
Sig.	99% Confidence Interval	Lower Bound	.006
		Upper Bound	.010

a. Friedman Test

Figura 3.7: Resultados de significación

Por ofrecer el menor error y el menor rango se aplicará FuzzyIBk con la FP Trapezoidal y el valor de  $k=4$ .

### 3.2.1.2. FuzzyIBk vs IBk

Para comparar el desempeño de FuzzyIBk con el del IBk se tendrá en cuenta lo siguiente:

Para ambos modelos se utilizará como función de distancia *HEOMJaccard*.

Para IBk se tomará  $k < 15$  óptimo obtenido mediante el *10 folds cross-validation*. La tabla (3.2.1.2) muestra el porcentaje de clasificaciones correctas y el error medio al realizar la validación

*10 times 10-folds cross-validation* para comparar los modelos FuzzyIBk e IBk; 8 resultados son favorables para el nuevo modelo y 1 para IBk.

Datos	IBk			FuzzyIBk			
	k	%	Error medio	k	FP	%	Error
contact-lenses	1	75	0.2	1	3 Gauss.	91.66	0.2
diabetes	3	73.30	0.3	4	2 Gauss.	87.23	0.2
glass	1	69.15	0.09	2	7 Gauss.	90.18	0.05
ionosphere	2	88.60	0.14	2	2 Trapez.	94.58	0.07
iris	6	96	0.04	4	3 Trapez.	96	0.02
labor	3	91.22	0.17	4	2 Trapez.	92.98	0.19
labor	4	92.64	0.08	3	2 Trapez.	89.47	0.04
weather	1	57.17	0.4	1	3 Trapez.	100	0.3
infantil	1	89.69	0.06	5	3 Trapez.	93.81	0.07

Cuadro 3.3: Comparación entre los modelos FuzzyIBk e IBk

Al aplicar el test de Wilcoxon utilizando los valores de los errores, se obtiene que los resultados para cada conjunto de datos presentan diferencias significativas (tablas 3.2.1.2, 3.2.1.2).

	N	Rango promedio	Suma de rangos
FuzzyIBk - IBk Rangos negativos	6(a)	5.33	32.00
Rangos positivos	2(b)	2.00	4.00
Empates	1(c)		
Total	9		

a FuzzyIBk < IBk

b FuzzyIBk > IBk

c FuzzyIBk = IBk

Cuadro 3.4: Resultado de ranqueo de FuzzyIBk e IBk según el test de Wilcoxon.

	FuzzyIBk - IBk
Z	-1.975(a)
Sig. asintót. (bilateral)	.048
Sig. exacta (unilateral)	.027
Probabilidad en el punto	.012

a Basado en los rangos positivos.

b Prueba de los rangos con signo de Wilcoxon

Cuadro 3.5: Estadísticos de contraste (b).

### 3.2.1.3. Comparación con otros enfoques

Actualmente se están realizando experimentos con nuevos algoritmos creados para facilitar la manipulación de rasgos de tipo conjunto. Este tipo de dato se encuentra presente en muchos problemas de clasificación. Weka cuenta con diversos algoritmos para resolver este tipo de problemas desde diferentes enfoques, permitiendo la manipulación de rasgos multievaluados; el modelo FuzzyIBk es uno de ellos. Se requiere entonces una comparación de las diversas técnicas adaptadas para este fin.

La base de casos empleada en Saprim 2.0 cuenta con 5 rasgos multievaluados, con un gran número de valores cada uno, por lo que este será el conjunto de datos empleado para realizar la comparación.

La tabla (3.2.1.3) muestra los resultados de esta comparación, donde tanto IBk como FuzzyIBk emplean la función de distancia HEOMJaccard y los parámetros que ofrecieron los mejores resultados en la tabla (3.2.1.2), evaluando solamente la base de casos “infantil”.

Algoritmo	%	Error
FuzzyIBk	93.81	0.07
IBk	89.69	0.06
ID3	82.32	0.09
NaiveBayes	91.26	0.06

Cuadro 3.6: Comparación del modelo con otros enfoques.

Además de resolver el problema de investigación que se presenta en este trabajo de diploma, el modelo FuzzyIBk constituye un buen clasificador, pues ofrece el mayor porcentaje de cla-

sificación correcta en comparación con otros enfoques, además, el error producido no muestra diferencias significativas con los errores del resto de los algoritmos. Esto indica que el algoritmo puede considerarse confiable.

El comportamiento del filtro implementado, NominalToLinguistic, se mide mediante la propia evaluación del clasificador FuzzyIBk, ya que este hace uso interno del filtro y sus resultados indican el nivel de correctitud en el desempeño del mismo.

#### 3.2.1.4. Validación del modelo computacional obtenido

Para garantizar una buena clasificación del caso problema es necesaria una buena selección del resolvidor de problemas. En este trabajo se empleó FuzzyIBk y se realizaron las comparaciones necesarias con el resto de los modelos, que podrían haber sido posibles candidatos, y fue precisamente FuzzyIBk el que arrojó un mayor índice de clasificaciones correctas llegando a un 93.81 % aproximadamente.

Se realizaron pruebas sobre una muestra de control y se obtuvieron los siguientes resultados:

	de cuidado	grave	crítico	total
de cuidado real	20	0	0	20
grave real	1	14	0	15
crítico real	0	1	5	6
total	21	15	5	41

<b>Instancias correctamente clasificadas</b>	0.9381
Instancias correctamente clasificadas de cuidado	1
Instancias correctamente clasificadas de grave	0.933
Instancias correctamente clasificadas de crítico	0.833

Cuadro 3.7: Resultados con el grupo de prueba empleando FuzzyIBk

De los 41 casos que tenía la muestra de control el 95.12 % de los mismos fueron clasificados correctamente, un 100 % de los casos de cuidado, un 93.3 % de los casos graves y un 83.3 % de los casos críticos.

Por parte de los usuarios (expertos en el dominio) ha quedado expresada su conformidad y satisfacción con los resultados obtenidos con Saprím y se sienten cómodos en el manejo del sistema. El sistema Saprím ha quedado debidamente validado y evaluado. Su empleo resulta fácil gracias al manual de usuario que explica todas las facilidades del sistema.

# Conclusiones

- Quedaron claros conceptos y temas de necesario conocimiento para la comprensión del problema a resolver.
- Se implementó un sistema para el pronóstico del nivel de gravedad de los pacientes que arriban a la UCI del Hospital Infantil “José Luis Miranda” de Santa Clara, para ello:
  - Se efectuó un diseño adecuado del sistema a desarrollar.
  - Se obtuvo una BC con un balance adecuado en la información recopilada.
  - Se implementó un nuevo algoritmo de preprocesamiento denominado NominalTo-Linguistic para transformar el rasgo objetivo de tipo nominal a borroso.
  - Se implementó un nuevo algoritmo de clasificación denominado FuzzyIBk que maneja y cuantifica datos con rasgo objetivo de tipo borroso.
  - Se eligió FuzzyIBk para discriminar entre pacientes de cuidado, graves y críticos.
  - El sistema implementado pronostica el nivel de gravedad con un 93.81 % de efectividad con la muestra de aprendizaje obtenida.
- Se validó y evaluó el sistema implementado debidamente.
- A partir de factores de riesgos epidemiológicos y clínicos es posible discriminar entre pacientes de cuidado, graves y críticos mediante la aplicación de técnicas de clasificación supervisada de la IA.
- Los resultados que ofrece este sistema computacional están al nivel de los obtenidos por los expertos humanos al valorar un paciente.

## **Recomendaciones**

Se recomienda incorporar un módulo de análisis y autoaprendizaje a Saprim que no requiera la intervención del especialista médico.

## Referencias Bibliográficas

Modelo de un Sistema Basado en Casos para el diagnóstico del Síndrome de Maloclusión. 1999

**Aguirre, A.:** Valoración del paciente pediátrico grave y hallazgos clínico-epidemiológicos a su ingreso. Hospital Pediátrico Universitario Santa Clara 2002

**Bello, R./Morell, C.:** Una arquitectura unificada para el razonamiento borroso. Revista Cubana de Ciencias Informáticas, 2007, 68–83

**Bello, R., C. Morell:** Modelos para el razonamiento basado en casos en presencia de rasgos borrosos. UCLV 2000

**Bello, R.:** Curso Introductorio a las Redes Neuronales Artificiales. UCLV 1993

**Bello, R.:** Aplicaciones de la Inteligencia Artificial. Guadalajara: Universidad de Guadalajara 2002

**Cruz, A.V. De la:** EPILEP Un sistema consultante con Inteligencia Artificial para el diagnóstico y tratamiento de la epilepsia. 1993

**Detyniecki, M.:** Mathematical Aggregation Operators and their application to video querying. Tesis Doctoral, Pierre and Marie Curie University, Paris, France 2000

**García, Z.:** Sistemas Expertos. Aprendizaje e Incertidumbre. 2006

**García, A.:** Sistema híbrido para el pronóstico del nivel de gravedad en pacientes pediátricos al ingresar en una Unidad de Cuidados Intensivos. UCLV 2003

**García, Z.:** Introducción a la Inteligencia Artificial. México: Universidad de Guadalajara 2000

- Gonzalez, H. Matías/Araujo, L. I.:** Extensiones al ambiente de aprendizaje automatizado WEKA. Universidad Central "Martha Abreu ", Villa Clara, Cuba Informe técnico 2006
- González, Mabel:** Sistema pronóstico del paciente pediátrico. 2007
- Hullemeier, E.:** Case -Based Approximate Reasoning. Springer Volumen 44 2007
- Kuncheva, L I./Bezdek, J C.:** A Fuzzy Generalized Nearest Prototipe Classifier. Proc. of 7th IFSA World Congress, 1997, 217–222
- Lacroix, J./Cotting, J.:** Severity of illness and organ dysfunction scoring in children. *Pediatr Crit Care Med*, 2005, p 126–134
- Lezcano, M.:** Prolog y los Sistemas Expertos.
- Matías, H./Araugo, L.I.:** Extensiones al Ambiente de Aprendizaje Automatizado Weka. UCLV 2006
- Mitchell, T.M.:** Machine Learning. 1997
- Munakata, T.:** Fundamentals of the New Artificial Intelligence. Springer 2008
- Nauck, D./Klawonn, F./Kruse R.:** Foundations of Neuro-Fuzzy Systems. Jhon Wiley & Sons. Inc. 1997
- Pal, S. K/Shiu, S. C. K.:** Foundations of soft case-based reasoning. Hoboken, New Jersey. John Wiley & Sons Inc. 2004
- Pollack, M., V. Ruttimann/Getson, P.:** Pediatric Risk of Mortality (PRIMS) score. *Crit Care Med*, 1988, p 1011
- Rich, E.:** Inteligencia Artificial. 2 ed. 1994
- Rodríguez, Y.:** Sistema Basado en Casos para el diagnóstico de malformaciones cardiovasculares en recién nacidos. UCLV 1996
- Rovira, L.E:** Evaluación del paciente ingresado en la unidad de cuidados intensivos. Facultad de Medicina 2001

- Sarabia, Y. Rodriguez:** Generalización de la métrica basada en la diferencia de valores (VDM) para variable lingüísticas y su aplicación en sistemas basados en el conocimiento. Tesis Doctoral, Universidad Central "Marta Abreu" de las Villas, Cuba 2007
- Shann, F., G. Pearson/Slater, A.:** Pediatric Index of Mortality (PIM): A mortality prediction model for children in intensive care. *Intensive Care Med*, 1997, p 201–207
- Shing, J./Jang, R.:** Neuro-Fuzzy and Soft Computing. A computational approach to learning and machine intelligence. Tesis Doctoral, Computer Science Department, Tsing Hua University, Taiwan The MathWorks, Naticks, Massachusetts, USA 1997
- Siler, W./Buckley, J. J.:** Fuzzy Expert System and Fuzzy Reasoning. John Wiley & Sons Inc. Hoboken. New Jersey 2005
- Slater, A., F. Shann/Pearson, G.:** PIM2: A revised version of the Pediatric Index of Mortality. *Intensive Care Med*, 2003, p 278–285
- Sugeno, M./Takagi, T.:** Derivations of fuzzy control rules from human operator's control actions. IFAC Symposium of Fuzzy Information Knowledge Representation and Decision Analysis 1983
- Sánchez, A.:** El problema del diagnóstico en Inteligencia Artificial. 2006
- Trias, R.:** Inteligencia artificial en medicina. Estado actual y perspectivas. *Med Clin Barc*, 1993, p 45–46
- Urueta, L., H. Valdés/Contreras, J.:** La lógica difusa como apoyo a la enseñanza. 2000
- Zadeh, L. A.:** Applied Soft Computing. ASOC, 2001, 1,2
- Zadeh, L.A.:** A Computational Approach to Fuzzy Quantifiers in Natural Language. *Computing and Mathematics with Applications*, 1983, p 149–184
- Zadeh, L.A.:** From Computing with Numbers to Computing with Words -From Manipulation of Measurements to Manipulation of Perceptions. *Intelligent Systems and Soft Computing*, 2000, p 3–40

# Anexos

Cuadro 3.8: Tipos de Sistemas Basados en el Conocimiento (SBC).

<b>Nombre</b>	<b>Forma de Representación del Conocimiento (FRC)</b>	<b>Explicación</b>	<b>Método de solución del Problema (MSP)</b>	<b>Fuentes de Conocimiento</b>
Sistemas Basados en Reglas (SBR)	Reglas de Producción	Reglas activas	Búsqueda primero en profundidad con dirección backward o forward	Expertos, publicaciones, ejemplos
Sistemas basados en Frames (SBF)	Frames		Herencia y procedimientos adjuntos	Expertos, publicaciones, ejemplos
Sistemas basados en casos (SBCasos)	Casos	Casos semejantes	Razonamiento basado en casos (búsqueda por semejanza y adaptación de las soluciones)	Ejemplos
Sistemas basados en probabilidades (SBP)	Probabilidades o frecuencias	Valores de probabilidades condicionales	Teorema de Bayes y otras técnicas de inferencia estadística	Ejemplos
Redes expertas	Pesos y alguna otra FRC	Según el modelo simbólico	Cálculo de niveles de activación de las neuronas	Ejemplos
Sistemas basados en modelos	Modelo del artefacto		Razonamiento basado en modelos	Esquemas estructurales y funcionales del artefacto

Cuadro 3.9: Planilla usada para captar los datos del paciente.

NOMBRE: .....

HC: .....

EDAD: ..... MESES

FECHA DE NACIMIENTO:.....

SEXO:.....

REINGRESO: SI..... NO.....

EN: NORMAL..... DELGADO..... DESNUTRIDO.....OBESO.....

MODO DE INGRESO:

- TRASLADO.....

- DIRECTO: .....

- REMITIDO:

- NO.....

- SI: AMBULANCIA..... MEDIOS PROPIOS.....

DIAGNOSTICO:.....

ESTADO DE SALUD PREVIO: SANO..... ENFERMO.....

TEMPERATURA: NORMAL..... FEBRICOLA..... HIPER..... HIPO.....

COLORACION: PALIDEZ..... NORMAL..... ICTERO..... CIANOSIS..... RUBICUNDEZ.....

HIDRATACION: NORMAL..... DESHIDRATADO..... SOBREHIDRATADO.....

FR: NORMAL..... POLIPNEA..... BRADIPNEA.....

TIRAJE:

- NO.....

- SI:

- SUBCOSTAL.....
- INTERCOSTAL.....
- SUPRAESTERNAL.....

MURMULLO VESICULAR: NORMAL: SI..... NO.....

ESTERTORES: NO..... SI.....

QUEJIDO ESPIRATORIO: SI..... NO.....

ESTRIDOR INSPIRATORIO: SI..... NO.....

ALETEO NASAL: SI..... NO.....

TENSION SISTOLICA.....

TENSION DIASTOLICA.....

PERFUSION TISULAR: BUEN..... REGULAR..... MALA.....

PULSO: NORMAL..... ANORMAL.....

FRECUENCIA CARDIACA: NORMAL..... TAQUI..... BRADI.....

PURPURAS: SI..... NO.....

HEMATURIA: SI..... NO.....

SANGRAMIENTOS: SI..... NO.....

DIURESIS: NORMAL..... OLIGO..... POLI.....

SENSORIO: LIBRE..... DEPRIMIDO..... IRRITABLE.....

FONTANELA: NORMOTENSA..... DEPRIMIDA..... ABOMBADA..... CERRRADA.....

CONVULSIONES: SI..... NO.....

PUPILAS: ISOCORICAS Y REACTIVAS..... MIOSIS..... MIDRIASIS.....

TONO MUSCULAR: NORMAL..... HIPOT..... HIPERT.....

DEFECTO MOTOR: SI..... NO.....

Cuadro 3.10: Distribución de rasgos en la Interfaz Gráfica de Usuario

<b>Pestaña</b>	<b>Rasgos que contiene</b>
Datos Generales	Historia Clínica Edad Modo de Ingreso Temperatura Estado de Hidratación Estado Nutricional  Coloración de la piel Estado de Salud Previo Diagnóstico
Signos Respiratorios	Frecuencia Respiratoria Murmullo Vesicular Estridor Inspiratorio Quejido Espiratorio Aleteo Tiraje Estertores

<b>Pestaña</b>	<b>Rasgos que contiene</b>
Signos Vitales	Perfusión tisular Pulso Frecuencia Cardíaca Tensión Arterial Sistólica Tensión Arterial Diastólica
Signos Circulatorios	Hematurias Sangramientos  Diuresis
Signos Neurológicos	Sensorio Pupilas Tono Muscular Convulsiones Fontanela Defecto Motor