

**CONSTRUCCIÓN DEL MOTOR DE INFERENCIA DIFUSO DEL PROTOTIPO  
SOFTWARE DE SIMULACIÓN PARA EL ESTUDIO DE LA POLUCIÓN DEL  
AIRE MID-CASS.**

**FERNEY FARID FUENTES MORELO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA**

**2007**

**CONSTRUCCIÓN DEL MOTOR DE INFERENCIA DIFUSO DEL PROTOTIPO  
SOFTWARE DE SIMULACIÓN PARA EL ESTUDIO DE LA POLUCIÓN DEL  
AIRE MID-CASS.**

**FERNEY FARID FUENTES MORELO**

**Trabajo de grado para optar por el título de:  
Ingeniero de Sistemas**

**Directora**

**MARCELA JAIMES BECERRA**

**Ingeniera de Sistemas**

**Codirector**

**Ph.D. JORGE LUIS CHACÓN VELASCO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA**

**2007**

DEDICATORIA

*De todo corazón a mis padres:*

*Un hombre perseverante,  
y como tu me dices “un hombre nunca se cansa”,  
mi padre Eber Fuentes.*

*El corazón más grande que he conocido:*

*mi madre Ladys Morelo*

*A una de las personas que más he querido:*

*mi abuelo Rafael Donato Morelo Cintero*

*a mi abuela Carlota Ortega*

*Y a mis hermanos:*

*Fred Fernando, Farly Faneth y Ever Antonio*

## AGRADECIMIENTOS

*A Dios por toda la sabiduría,*

*A mi familia por todo su amor, apoyo y comprensión.*

*A la familia Muñoz Avellaneda,*

*especialmente a Martha, Alfonso, Bárbara y María.*

*A mi novia Lucía,*

*A mis compañeros de universidad y amigos:*

*Rodrigo, Jairo, Maru, Nini, Lizeth, Jenny Paola, Mario,...*

*A mis compañeros de la selección de rugby,*

*donde encontré grandes hermanos.*

*A Benjamín Mancera y su familia*

*A Alejandro Niño*

*A Sandra Bandera e hijos: Julieth y Jirsen*

## CONTENIDO

	Pág.
<b>INTRODUCCIÓN</b>	1
<b>1. ASPECTOS GENERALES</b>	2
1.1. ANTECEDENTES Y DESCRIPCIÓN DEL PROBLEMA	2
1.2. OBJETIVOS DEL PROYECTO	6
1.2.1. Objetivo General	6
1.2.2. Objetivos Específicos	7
1.3. JUSTIFICACIÓN, IMPACTO Y VIABILIDAD	7
1.3.1. Justificación	7
1.3.2. Impacto y viabilidad	9
<b>2. MARCO TEÓRICO</b>	11
2.1. INTRODUCCIÓN	11
2.2. INTELIGENCIA ARTIFICIAL	12
2.2.1. Historia de la IA	12
2.2.2. Escuelas del pensamiento	13
2.3. SISTEMAS EXPERTOS	15
2.3.1. Estructura básica de un SE	16
2.4. LÓGICA BORROSA	21
2.4.1. Historia	22
2.4.2. Conjuntos borrosos o difusos	24
2.4.3. Propiedades de los conjuntos borrosos	27
2.4.4. Funciones de pertenencia o membresía	28

2.4.5. Operaciones entre conjuntos borrosos	32
2.4.6. Modificadores lingüísticos	37
2.4.7. Variable lingüística	38
2.4.8. Proceso de inferencia borrosa	40
<b>2.5. SISTEMA DE INFERENCIA BORROSA</b>	<b>44</b>
2.5.1. Tipos de sistemas basados en reglas borrosas	45
2.5.2. Elementos de un Sistema de Inferencia Borrosa	48
2.5.3. Interface de Borrosificación o Borrosificador	49
2.5.4. Base de conocimiento	50
2.5.5. Motor de inferencias	51
2.5.6. Interface de Desemborronado o Desborrosificación	52
<b>3. METODOLOGÍA Y PLAN DE TRABAJO</b>	<b>54</b>
3.1. METODOLOGÍA	54
3.2. PROGRAMACIÓN ORIENTADA A OBJETOS	56
<b>3.3. LENGUAJE UNIFICADO DE MODELADO UML</b>	<b>57</b>
3.4. PLAN DE TRABAJO	59
3.4.1. FASE I: Concepto Inicial	59
3.4.2. FASE II: Construcción	60
3.4.3. FASE III: Prueba del Sistema	61
3.4.4. FASE IV: Documentación	61
<b>4. FASE DE DE CONSTRUCCIÓN</b>	<b>62</b>
4.1. PRIMER PROTOTIPO	62
4.1.1. Requisitos	62
4.1.2. Diseño	72
4.1.3. Implementación	74

4.1.4. Evaluación	74
4.2. SEGUNDO PROTOTIPO	76
4.2.1. Requisitos	77
4.2.2. Diseño	83
4.2.3. Implementación	84
4.2.4. Validación final	91
<b>5. FASE DE PRUEBAS</b>	<b>92</b>
5.1. PRELIMINAR	92
5.2. MODELO DE LA CONTAMINACIÓN DEL AIRE	93
5.2.1. Datos de entrada	93
5.2.2. Descripción de los datos de entrada	93
5.2.3. Contaminantes criterio	96
5.2.4. Gases contaminantes	96
5.2.5. Unidades de los contaminantes criterio	98
5.3. EJECUCIÓN DE PRUEBAS	99
5.3.1. Características del archivo generado	99
5.3.2. Características del prototipo	99
5.4. RESULTADOS	108
5.5. EJECUCIÓN DE LOS MODIFICADORES	115
5.5.1. Modificador not	116
5.5.2. Modificador dilatación	118
5.5.3. Modificador concentración	119
<b>6. CONCLUSIONES</b>	<b>121</b>
<b>7. RECOMENDACIONES</b>	<b>123</b>
<b>8. BIBLIOGRAFÍA</b>	<b>125</b>
<b>ANEXOS</b>	<b>127</b>

## LISTA DE TABLAS

	<b>Pág.</b>
Tabla 1. Velocidad del viento estación de monitoreo Terraza Mercadefam	95
Tabla 2. Unidades de los contaminantes criterio	98
Tabla 3. Abreviaturas para Variables Lingüísticas de entrada	99
Tabla 4. Abreviaturas para Variables Lingüísticas de salida	99
Tabla 5. Sinopsis parámetros del modelo.	101
Tabla 6. Sinopsis de resultados	113
Tabla 7. Sinopsis de resultados NOT	117

## LISTA DE FIGURAS

	<b>Pág.</b>
Figura 1. Diagrama de Paquetes	4
Figura 2. Ramas de la Inteligencia Artificial	11
Figura 3. Elementos que componen el sistema experto	16
Figura 4. Comparación entre la lógica clásica y lógica borrosa.	25
Figura 5. Funciones características más habituales: (a) triangular, (b) trapezoidal, (c) gaussiana y (d) sigmoial	27
Figura 6. Función de pertenencia Triangular	29
Figura 7. Función de Pertenencia Trapezoidal	30
Figura 8. Función de pertenencia Gaussiana	30
Figura 9. Función de Pertenencia Curva de Bell	31
Figura 10. Función de Pertenencia tipo Sigmoial	32
Figura 11. Operaciones borrosas	33
Figura 12. Modificadores aplicados sobre un conjunto borroso	38
Figura 13. Modificador Lingüístico	38
Figura 14. Definición de la variable lingüística Temperatura	39
Figura 15. Representación gráfica del método de interpolación	43
Figura 16. Sistema de Lógica Borrosa	48
Figura 17. Esquema de flujo de trabajo	55
Figura 18. Diagrama de UML.	58
Figura 19. Casos de uso cvar y oper	69
Figura 20. Casos de uso para el primer prototipo	70
Figura 21. Diagramas de UML para el prier prototipo.	73
Figura 22. Variable Lingüística en forma gráfica	75
Figura 23. Variable Lingüística e inserción de métodos.	76
Figura 24. Diagrama de casos de uso de la interface Gráfica de	78

Usuario.

Figura 25. Casos de uso para el segundo prototipo	81
Figura 26. Módulos para y cvar	82
Figura 27. Módulos para y cvar integrados a la interface principal	82
Figura 28. Diagramas de UML para el segundo prototipo.	83
Figura 29. Tabs, pestañas o lengüetas.	84
Figura 30. Tab Modelo Fuzzy.	87
Figura 31. Interface gráfica integrada a los subpaquetes	88
Figura 32. Módulo de parámetros y variables lingüísticas	89
Figura 33. Tab editor de reglas	89
Figura 34. Tab resultados	90
Figura 35. Tab Gráfica 3D.	91
Figura 36. Contaminantes emitidos por los automóviles.	96
Figura 37. Configuración del modelo.	102
Figura 38. Variable de entrada Volumen Tráfico	103
Figura 39. Variable de entrada Intensidad Solar	103
Figura 40. Variable de entrada Velocidad viento	104
Figura 41. Variable de entrada Temperatura	104
Figura 42. Variable de salida concentración de CO	105
Figura 43. Variable de salida concentración de Ozono	105
Figura 44. Base de reglas	106
Figura 45. Resultados gráficos y numéricos.	107
Figura 46. Gráfico tridimensional.	107
Figura 47. Configuración inicial: arriba Prototipo vs abajo Matlab	108
Figura 48. Volumen tráfico: arriba Prototipo vs abajo Matlab	109
Figura 49. Intensidad solar: arriba Prototipo vs abajo Matlab	109
Figura 50. Velocidad Viento: arriba Prototipo vs abajo Matlab	110
Figura 51. Temperatura: arriba Prototipo vs abajo Matlab	110
Figura 52. CO: arriba Prototipo vs abajo Matlab	111
Figura 53. O <sub>3</sub> : arriba Prototipo vs abajo Matlab	111
Figura 54. Reglas: arriba Prototipo vs abajo Matlab	112
Figura 55. Resultados: arriba Prototipo vs abajo Matlab	112

Figura 56. Gráfico 3D: arriba Matlab vs abajo Prototipo	113
Figura 57. Proyecto distribuido en el escritorio (Toolbox Fuzzy Logic de Matlab)	114
Figura 58. Proyecto distribuido en el escritorio (Prototipo Fuzzy Logic)	115
Figura 59. Modificador not en Tab Reglas	116
Figura 60. Modificador not en Tab Resultados	117
Figura 61. Modificador dilatación en Tab Reglas	118
Figura 62. Modificador dilatación en Tab Resultados	118
Figura 63. Modificador concentración en Tab Reglas	119
Figura 64. Modificador concentración en Tab Resultados	120

## RESUMEN

**TITULO:**

CONSTRUCCIÓN DEL MOTOR DE INFERENCIA DIFUSO DEL PROTOTIPO SOFTWARE DE SIMULACIÓN PARA EL ESTUDIO DE LA POLUCIÓN DEL AIRE MID-CASS\*

**AUTOR:**

Ferney Farid Fuentes Morelo\*\*

**PALABRAS CLAVES:**

Sistemas Expertos, inteligencia artificial, razonamiento aproximado, polución del aire atmosférico, modelos de simulación, Lógica borrosa.

**DESCRIPCIÓN:**

Actualmente la contaminación ambiental es un problema que está afectando a gran escala el planeta y la vida humana, por esto debemos corregir los hábitos, practicas o aparatos que la ocasionan, para ello debemos conocer los factores que crean este fenómeno y como lo estamos afectando, en este caso los agentes móviles.

Para este fin utilizamos una metodología conocida como lógica difusa, recreando un prototipo capaz de modelar la situación. En este documento se expone la metodología, diseño y pruebas que se realizaron para el prototipo desarrollado y se presenta un modelo que muestra los principales factores que intervienen en este fenómeno y los contaminantes criterio que se producen.

Este prototipo desarrollado nos da la capacidad de construir un modelo y medir los efectos que proyectan en cada contaminante criterio, con un valor agregado que es la posibilidad de recrear casi cualquier otro fenómeno modelable con la metodología de lógica borrosa.

Este proyecto está asociado al grupo GIEMA de Ingeniería Mecánica y al CEIAM de Ingeniería Química y financiado por el DIF y servirá como apoyo en la toma de decisiones por parte de las autoridades locales como la CDMB, el mismo GIEMA y CEIAM.

---

\* Trabajo de grado

\*\*Facultad de Ingenierías Físico-Mecánicas, Escuela de Ingeniería de Sistemas e Informática y Escuela de Ingeniería Mecánica. Universidad Industrial de Santander. UIS.  
Director(a): Ing. Marcela Jaimes Becerra, Codirector: Ph.D: Jorge Luíz Cachón V.

## ABSTRACT

**TITLE:**

CONSTRUCTION OF THE DIFFUSE INFERENCE MOTOR FOR PROTOTYPE SIMULATION SOFTWARE FOR STUDING AIR POLLUTION MID-CASS\*

**AUTHOR:**

Ferney Farid Fuentes Morelo\*\*

**KEY WORDS:**

Expert systems, artificial intelligence, approximated reasoning, pollution of the atmospheric air, simulation models, fuzzy Logic.

**DESCRIPTION:**

Nowadays environmental pollution is a problem affecting the planet and human life on a large scale. For this reason we must correct habits, practices and hardware that causes it. To achieve this goal we must understand the factors that create this phenomena and how we are affecting then, by using movable agents.

For this aim we used a well-known methodology known as fuzzy logic, recreating a prototype able to model the situation. In this document the methodology, design and tests that were made for the developed prototype are demonstrated along with a model that shows the main factors which contribute to this phenomenon with polluting agents criterion included.

This developed prototype gives us the capacity to construct a model and to measure the effects that are projected in each polluting criterion, with an added value being the possibility of recreating almost any other modelable phenomena with the methodology of fuzzy logic.

This project is associated with the GIEMA the School of Mechanical Engineering and with the CEIAM the School of Chemical Engineering and financed by the DIF and will serve to support decision-making process of local authorities like the CDMB, GIEMA and CEIAM.

---

\* Degree project.

\*\*Faculty of Physical Mechanical Engineering, Systems Engineering and the school of Computer Science and Mechanical Engineering. Industrial University of Santander. UIS.  
Director(a): Marcela Jaimes Becerra, Eng. Co-director: Jorge Luís Cachón V., Ph.D:

## INTRODUCCIÓN

Este trabajo hace parte de un proyecto de maestría en informática y ciencias de la computación en el Grupo de Investigación en Energía y Medio Ambiente (GIEMA), que tiene como título “Modelado y simulación de la contaminación del aire atmosférico producido por fuentes móviles del área metropolitana de Bucaramanga”. Se trata del problema de la contaminación ambiental del área metropolitana de Bucaramanga causada específicamente por el parque automotriz; este proyecto busca una forma de acceder al conocimiento de los fenómenos que producen la contaminación del aire atmosférico mediante un prototipo utilizando la Lógica Difusa.

El trabajo de maestría comprende entre sus objetivos el desarrollo de un prototipo software de simulación que usa la Lógica Difusa para la contaminación del aire. Este se dividió en paquetes y este proyecto de grado se centra en la construcción del Motor de Inferencias de dicha herramienta computacional.

El procesamiento, control y la predicción de la información habían sido abordados hasta hace algunos años por la inteligencia artificial clásica, sin embargo la complejidad de este modelo en abordar tareas del mundo real donde la información se presenta de forma escasa ha originado el desarrollo de la llamada *Soft Computing* o “Computación Blanda” que comprende el estudio de las redes neuronales, sistemas borrosos y los algoritmos genéticos, tecnología aplicada actualmente a señales telefónicas, reconocimiento óptico de caracteres, ayuda al pilotaje de aviones, así como estabilización de imágenes de video y el control de lavadoras.

## **1. ASPECTOS GENERALES**

### **1.1. ANTECEDENTES Y DESCRIPCIÓN DEL PROBLEMA**

El derecho a la vida y a la salud de cada persona es innegable y es un deber, de todo aquel que tenga en sus manos la posibilidad de contribuir a la humanidad a tener una mejor calidad de vida y ayudar a preservarla.

Los constantes cambios tecnológicos, la automatización de las empresas y su acelerada expansión a todos los campos, han producido bruscas alteraciones en la naturaleza y especialmente una fuerte contaminación, los cuales se ven reflejados en los drásticos y repentinos cambios climáticos y en el calentamiento global. Es un deber de las personas que cuentan con los medios, aprovechar todos los recursos tecnológicos, para poder responder a las necesidades actuales, que es una exigencia tanto de la comunidad como del medio ambiente.

Las principales fuentes de contaminación son las industrias, los desperdicios, el parque automotriz (carros, motos, etc.), entre otros; pero el parque automotor produce cifras muy alarmantes, convirtiéndose en uno de los principales problemas que afectan a los habitantes de la ciudad, por ésta razón se pensó en hacer un estudio de la contaminación ambiental de Bucaramanga y su área metropolitana, específicamente la producida por fuentes móviles<sup>1</sup>.

”Actualmente el parque automotor de la meseta de Bucaramanga está comprendido en su mayor parte por automóviles públicos y privados y tiene un

---

<sup>1</sup> Término especializado utilizado en contaminación del aire para referirse al parque automotor.

número aproximado de 124.400 vehículos<sup>2</sup>. Y si se multiplica esta cantidad por los kilómetros recorridos, para luego evaluar los contaminantes generados en la quema de combustibles fósiles, se observa una contribución considerable del parque automotor a la contaminación del aire atmosférico de la meseta de Bucaramanga”<sup>3</sup>.

En la actualidad algunas organizaciones como la CDMB<sup>4</sup> y el CEIAM<sup>5</sup>, están llevando a cabo mediciones de los contaminantes emitidos tanto, por fuentes fijas como por fuentes móviles y los estudios correspondientes. Sin embargo, estos datos sólo permiten conocer en qué grado van aumentando las emisiones año tras año y no se cuenta con una alternativa que permita a estas organizaciones anticiparse a los acontecimientos, es decir, conocer los datos antes de que el daño se produzca.

“Por lo tanto el problema radica en que existe una contribución significativa en la contaminación del aire del área metropolitana producida por fuentes móviles; por lo que se requiere conocer su forma de dispersión influenciada por las condiciones meteorológicas locales”<sup>6</sup>.

Dadas estas manifestaciones de la naturaleza y la evidente contaminación ambiental, se ve la necesidad de crear una herramienta basada en la lógica difusa, que pueda apoyar los estudios ambientales, de la siguiente manera:

- ✓ Qua a través de esta sea posible, estudiar la influencia que tienen las variables que intervienen en **la contaminación del aire** (meteorológicas y otras de acuerdo al tipo de fuente contaminante) de una zona del área metropolitana de Bucaramanga.
- ✓ Ayudar a predecir las variaciones de las concentraciones de los contaminantes.

---

<sup>2</sup> Informe de cantidad de vehículos por clase y modelo, de la Dirección de Transporte y Transito, del Ministerio de Transporte, 2005.

<sup>3</sup> y <sup>6</sup>Jaimés, Marcela. Propuesta de investigación. Modelado y simulación de la contaminación del aire atmosférico producido por fuentes móviles del área metropolitana de Bucaramanga. UIS. Bucaramanga.

<sup>4</sup> CDMB: Corporación Autónoma para la Defensa de la Meseta de Bucaramanga.

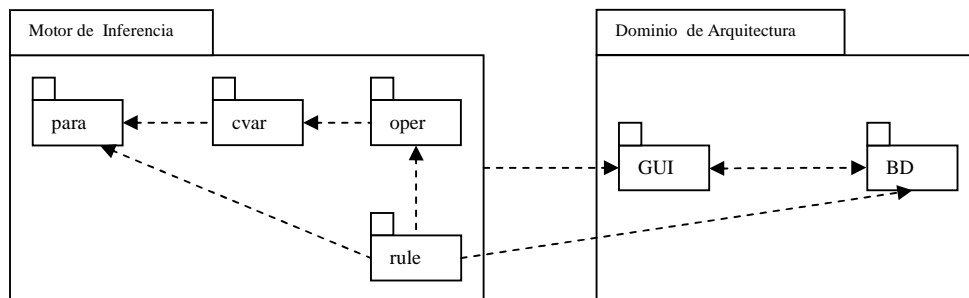
<sup>5</sup> CEIAM: Centro de Estudios e Investigaciones Ambientales.

Con el manejo de estos criterios se podría lograr:

- ✓ Una herramienta de apoyo preventivo para las autoridades competentes.
- ✓ Tomar medidas para poder corregir o disminuir los índices de contaminación, que hoy en día son cada vez más altos.

Como se mencionó anteriormente, este proyecto hace parte de un proyecto de maestría en informática y ciencias de la computación, que tiene como título “Modelado y simulación de la contaminación del aire atmosférico producido por fuentes móviles del área metropolitana de Bucaramanga”, cuyo objetivo general “Contribuir al desarrollo de modelos de predicción de fenómenos de la contaminación del aire producida por fuentes móviles en el área metropolitana de Bucaramanga” y uno de sus objetivos específicos es “Desarrollar un prototipo software de simulación empleando la lógica borrosa para fenómenos ambientales que sirva de base para futuras investigaciones en la calidad del aire”. El cual se identificará de ahora en adelante con la sigla CASS, que resulta del nombre prototipo software de simulación de la contaminación del aire.

En la literatura sobre las “tecnologías modernas para el desarrollo de software” se recomienda dividir un proyecto con objeto de distribuir el trabajo. A continuación se presenta el diagrama de paquetes de CASS donde se aprecia la división de este prototipo software (**Figura 1**):



**Figura 1.** Diagrama de Paquetes

El paquete **Dominio de Arquitectura** agrupa los componentes para la plataforma del software sobre los cuales la aplicación funciona:

- I. La **Interface Gráfica del Usuario (GUI)**: Comprende todo lo relacionado con las ventanas que interactúan con el usuario.
- II. La **Base de Datos**: Para administrar y almacenar toda la información referente a las variables de mayor incidencia en la polución del aire del casco urbano de la meseta de Bucaramanga producida por el flujo de los diferentes tipo de vehículos, aunque, esta base de datos es posible que se desarrolle en paralelo a este proyecto se planea que en un futuro esté integrada a este prototipo software de simulación.
- III. El paquete **Motor de Inferencia** en el que se asocian de manera sistemática las diferentes clases que concatenan las variables lingüísticas que están relacionadas con términos lingüísticos, con los diferentes operadores borrosos disponibles en la herramienta computacional, para finalmente asociar un conjunto de reglas de inferencia con los operadores y variables necesarias para generar una salida binaria o borrosa según como lo desee el usuario.

Este prototipo software de simulación se ha dividido en módulos, de tal manera que algunos de ellos fueron desarrollados como proyectos de grado por estudiantes de ingeniería de sistemas e informática, los cuales son:

- Desarrollo e Implementación de la Interface Gráfica del Usuario y su Integración con el Paquete “Motor de Inferencia Difuso” del Prototipo Sistema Experto Para La Polución Del Aire.
- Construcción del Motor de Inferencia Difuso del Prototipo Software de Simulación para el estudio de la Polución del Aire.

En este proyecto de manera específica se construyeron los subpaquetes “Variables” y “Operadores” del “Motor de Inferencia” y se trabajó en la integración con el segmento de “Reglas” que también pertenece al “Motor de Inferencia”. Además de la implementación del subpaquete “Para” perteneciente al “Motor de Inferencia”. “Para” provee al desarrollador las funciones básicas con el objeto de manejar “las tareas administrativas” necesarias para la definición de variables lingüísticas. Entre estas tareas se incluyen: Crear, inicializar, añadir y remover conjuntos de variables lingüísticas y además funciones básicas de entrada/salida. También se desarrollará el diseño e implementación de la ventana donde se representa la salida de los posibles sistemas de inferencia difusa.

El módulo “Variables” se enfoca en los detalles de una variable lingüística.

Se plantea que CASS tendrá mayor funcionalidad para un área representativa del casco urbano de la meseta de Bucaramanga, que fue seleccionada como tal por los grupos GIEMA Y CEIAM, por ser esta la zona de mayor congestión y además de la cual contamos con más y mejores datos.

## **1.2. OBJETIVOS DEL PROYECTO**

### **1.2.1. Objetivo General**

Desarrollar el módulo “Motor de Inferencia” que será parte de un prototipo software basado en la Lógica Difusa para modelar la Contaminación del Aire del área Metropolitana de Bucaramanga.

### **1.2.2. Objetivos Específicos**

Los objetivos específicos que se van a desarrollar dentro de este proyecto corresponden a los siguientes casos de uso del software titulado Prototipo Software de Simulación de la Contaminación del Aire (CASS). Los cuales son:

- Graficar las funciones de pertenencia de acuerdo a la función elegida por el usuario (Triangular, Trapezoidal, Gaussiana, Curva de Bell y Singleton) y algunos modificadores como el NOT y otros de dilatación y concentración.
- Elaborar el procedimiento que permita al software almacenar la configuración del término lingüístico y asociarlo a la variable a la cual pertenece.
- Implementar los operadores borrosos AND, OR, Max-Min y un proceso de desemborronado.

Además de estos casos de uso se realizarán los siguientes objetivos:

- Integración de los subpaquetes “Variables” y “Operadores” con el subpaquete “Reglas”, los cuales pertenecen al paquete Motor de Inferencia.
- Diseñar e implementar la interface de usuario para que se puedan representar los resultados del sistema de inferencia en forma numérica y gráfica.

## **1.3. JUSTIFICACIÓN, IMPACTO Y VIABILIDAD**

### **1.3.1. Justificación**

La Lógica difusa ha venido cobrando importancia en los últimos años debido a su fácil aplicación en la simulación de problemas de alta complejidad, tanto que hoy en día es una teoría muy utilizada y con muy buenos resultados.

Esta acogida se debe a que permite manejar información de difícil precisión, hace posible modelar y simular fenómenos complejos, imita el comportamiento de la lógica humana que puede procesar información incompleta ó inexacta. La Lógica Difusa ha sido probada para ser particularmente útil en sistemas expertos y otras aplicaciones de inteligencia artificial.

La lógica difusa también permite reducir costos en cuanto a la toma de datos, ya que esta se basa en la percepción y puede generar datos a partir de información parcial ó escasa, a diferencia de otras tecnologías de Computación Flexible. Adicionalmente, existen otras metodologías como la estadística para modelar y analizar los fenómenos ambientales y estimar algunos parámetros necesarios para predecir el comportamiento y las variaciones de éstos, también para estudiar los resultados obtenidos de los modelos de predicción y obtener nueva información relevante en la toma de decisiones. Sin embargo, el uso de esta técnica de simulación es costosa por cuanto requiere de una buena cantidad de datos experimentales que no siempre están disponibles.

Con este proyecto, se pretende desarrollar un primer prototipo software de simulación para modelar la polución del aire en Bucaramanga y el área metropolitana, producida por las fuentes móviles. Este prototipo, basado en la Lógica Difusa tiene como módulos centrales los subpaquetes "Variables" y "Operadores", los cuales constituyen el principal objetivo de este proyecto.

Los sistemas expertos disponibles, como el Toolbox Fuzzy Logic de Matlab que es una herramienta muy costosa, actualmente tiene un valor de 250.00 dólares y como requisito previo se debe adquirir MATLAB<sup>®</sup> con un valor de 625.00 dólares<sup>7</sup>, para un total de 875.00 dólares. Además, posee otras limitaciones, en especial para modelar fenómenos de esta naturaleza, debido a la carencia de ciertas funciones como son la mayoría de los modificadores. Otros simuladores más robustos tienen costos aún más elevados, por otra parte tampoco existe ningún software que se especialice en modelar con lógica difusa la

---

<sup>7</sup> The MathWorks. URL: <http://www.mathworks.com/store/productIndexLink.do>

contaminación del aire. Por tanto se puede decir que CASS es pionero en este campo. Además, tiene por objetivo la construcción de una alternativa software que será un prototipo adaptado a las necesidades del problema en cuestión, que permitirá a los grupos GIEMA<sup>8</sup> y CEIAM tener acceso a un sistema experto difuso que sirva como instrumento para la simulación de la contaminación del aire y en general para su estudio.

### **1.3.2. Impacto y viabilidad**

El prototipo software a desarrollar, servirá como una herramienta de apoyo a las organizaciones encargadas de realizar los estudios correspondientes a la contaminación del aire de la ciudad de Bucaramanga y su área metropolitana.

Se espera con este proyecto hacer un aporte significativo a la forma de analizar los datos obtenidos. Y tener acceso a una herramienta capaz de dar una aproximación al fenómeno de la contaminación del aire en Bucaramanga y de esta manera poder arrojar resultados que por otros métodos o procedimientos sería muy costoso o incompleto. Cabe anotar que este proyecto es la primera versión que se realiza con este tipo de estudios y quedará funcional para la ciudad de Bucaramanga ya que, a través de él es posible, conocer el comportamiento de las variables.

En el aspecto económico, este proyecto se considera viable debido a la alta demanda de computadores en el mercado que hace que los costos disminuyan, logrando que cualquier persona tenga acceso a ellos, además las tecnologías de desarrollo software a utilizar, son de libre distribución por lo tanto, no generan costos. Otra ventaja es que posee independencia de la plataforma.

En cuanto al aspecto social, se pretende con esta herramienta lograr una mejor calidad de vida, específicamente en cuanto a la salud de los habitantes de

---

<sup>8</sup> GIEMA (Grupo de Investigación en Energía y Medio Ambiente).

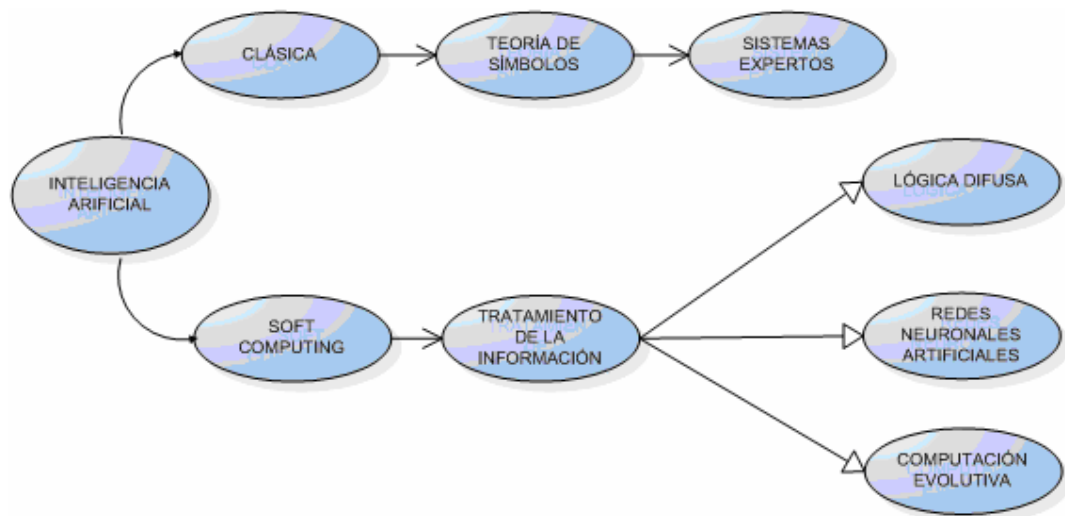
Bucaramanga y su área metropolitana y esperamos que este proyecto se extienda y pueda ser utilizado en otras ciudades del país ó fuera de él.

## 2. MARCO TEÓRICO

### 2.1. INTRODUCCIÓN

La necesidad de intervenir para hacer algo referente a la petición que nos está haciendo la naturaleza mediante los fenómenos que están ocurriendo y más específicamente con el de la contaminación ambiental lleva a la realización de este proyecto mediante la utilización de este método ya que la lógica difusa permite incorporar el concepto de incertidumbre o confianza, que integra: la imprecisión en la medición y la subjetividad que caracteriza al control lingüístico y donde el procesamiento de las reglas conduce a conclusiones con factores de confianza.

La lógica borrosa es una rama de la inteligencia artificial la cual comprende (Figura 2):



**Figura 2.** Ramas de la Inteligencia Artificial

## **2.2. INTELIGENCIA ARTIFICIAL**

La Inteligencia Artificial (IA), tiene por objeto el estudio del comportamiento inteligente de los computadores. A su vez, el comportamiento inteligente supone percibir, razonar, aprender, comunicarse y actuar en entornos complejos. “Una de las metas a largo plazo de la IA es el desarrollo de máquinas que puedan hacer todas estas cosas igual, o quizás incluso mejor, que los humanos. Otra meta de la IA es llegar a comprender este tipo de comportamiento, sea en las máquinas, en los seres humanos o en otros animales. Por tanto, la IA persigue al mismo tiempo metas científicas y metas de ingeniería; nosotros nos dedicaremos a la IA como ingeniería”.<sup>9</sup>

Las tecnologías de la IA incluyen, entre muchísimas, brazos robotizados con varios grados de libertad, demostración de teoremas y sistemas expertos que diagnostican enfermedades y prescriben remedios, así como la simulación de la inteligencia natural a través de dos vertientes: la computación conexionista (redes neuronales) y la computación simbólica (aplicación de las reglas lógicas). La Inteligencia Artificial se deriva de la vida artificial, de la ciencia y tecnología de la computación<sup>10</sup>.

### **2.2.1. Historia de la IA**

Los primeros desarrollos en inteligencia artificial comenzaron a mediados de los años 1950 con el trabajo de Alan Turing, el matemático británico que diseñó el primer computador electrónico digital y funcional del mundo en 1940, a partir de esta fecha la ciencia ha pasado por diversas situaciones:

- El término fue inventado en 1956 por John McCarthy, Marvin Minsky y Claude Shannon en la Conferencia de Darmouth, un congreso en el que se hicieron previsiones triunfalistas a diez años que jamás se cumplieron, lo que provocó el abandono casi total de las investigaciones durante quince años.

---

<sup>9</sup> Nilsson, Nils J. Inteligencia Artificial. Una nueva síntesis. McGraw Hill. España, 2001

<sup>10</sup> ETSIIT. “Inteligencia Artificial”, URL: <http://etsiit.ugr.es/alumnos/mlii/inteligencia.htm>

- En 1980 la historia se repitió con el desafío japonés de la quinta generación de computadoras, que dio lugar al auge de los sistemas expertos pero que no alcanzó muchos de sus objetivos, por lo que este campo sufrió una nueva interrupción en los años noventa.
- En la actualidad se está tan lejos de cumplir la prueba de Turing como cuando se formuló: *Existirá Inteligencia Artificial cuando no seamos capaces de distinguir entre un ser humano y un programa de computadora en una conversación a ciegas.*
- Como anécdota, muchos de los investigadores sobre IA sostienen que *"la inteligencia es un programa capaz de ser ejecutado independientemente de la máquina que lo ejecute, computador o cerebro"*.

La IA incluye características humanas tales como el aprendizaje, la adaptación, el razonamiento, la autocorrección, el mejoramiento implícito, y la percepción modular del mundo. Así, podemos hablar ya no sólo de un objetivo, sino de muchos, dependiendo del punto de vista o utilidad que pueda encontrarse a la IA.

Algunos ejemplos se encuentran en el área de control de sistemas, planificación automática, la habilidad de responder a diagnósticos y a consultas de los consumidores, reconocimiento de escritura, reconocimiento del habla y reconocimiento de patrones. De este modo, se ha convertido en una disciplina científica, enfocada en proveer soluciones a problemas de la vida diaria. Los sistemas de IA actualmente son parte de la rutina en campos como economía, medicina, ingeniería y la milicia, y se ha usado en gran variedad de aplicaciones de software, juegos de estrategia como ajedrez de computador y otros videojuegos.

### **2.2.2. Escuelas del pensamiento**

La IA se divide en dos escuelas de pensamiento, la inteligencia artificial clásica o convencional y la inteligencia computacional.

### **2.2.2.1. Inteligencia artificial clásica o convencional**

La inteligencia artificial convencional tiene que ver con métodos que actualmente se conocen como máquinas de aprendizaje, se caracteriza por el formalismo y el análisis estadístico. Algunos métodos de esta rama incluyen:

- **Sistemas expertos:** aplican capacidad de razonamiento para lograr una conclusión. Un sistema experto puede procesar una gran cantidad de información conocida y proveer conclusiones basadas en ésta.
- **Razonamiento basado en casos:** se ocupa del estudio de los mecanismos mentales necesarios para repetir lo que se ha hecho o vivido con anterioridad, ya sea por experiencia propia o por casos concretos recopilados en la bibliografía o en la sabiduría popular. Los diversos casos son del tipo "Si X, entonces Y" con algunas adaptaciones y críticas según las experiencias previas en el resultado de cada una de dichas reglas.
- **Red bayesiana:** un modelo de representación del conocimiento basado en teoría de la probabilidad.
- **Inteligencia artificial basada en comportamientos:** método modular para construir sistemas de IA manualmente.

### **2.2.2.2. Inteligencia artificial computacional**

La inteligencia computacional implica desarrollo o aprendizaje iterativo (*p.ej.* modificaciones iterativas de los parámetros en sistemas conexionistas). El aprendizaje se realiza basándose en datos empíricos. Algunos métodos de esta rama incluyen:

- **Redes neuronales:** sistemas con grandes capacidades de reconocimiento de patrones.
- **Sistemas difusos:** técnicas para lograr el razonamiento bajo incertidumbre. Ha sido ampliamente usada en la industria moderna y en productos de consumo masivo, como las lavadoras.

- Computación evolutiva: aplica conceptos inspirados en la biología, tales como población, mutación y supervivencia del más apto para generar soluciones sucesivamente mejores para un problema. Estos métodos a su vez se dividen en algoritmos evolutivos (ej. algoritmos genéticos) e inteligencia colectiva (ej. algoritmos hormiga)<sup>11</sup>.

### 2.3. SISTEMAS EXPERTOS

Un sistema experto (SE) es una aplicación informática que soluciona problemas complicados que de otra manera exigirían ampliamente la pericia humana. Para lograr esto, se simula el proceso de razonamiento humano mediante la aplicación específica de *conocimientos* y de *inferencias*.<sup>12</sup>

Internamente, un SE ideal se puede caracterizar como un sistema que comprende:

- Amplio conocimiento específico a partir del campo de interés;
- Aplicación de técnicas de búsqueda;
- Soporte para Análisis heurístico;
- Habilidad para inferir nuevos conocimientos a partir de conocimientos ya existentes;
- Procesamiento de símbolos;
- Capacidad para explicar su propio razonamiento.

El conocimiento del experto se organiza en una *base de conocimientos*, y en función de los datos disponibles de la aplicación (*base de hechos*) se imita la forma de actuar del experto explorando en la base de conocimientos hasta encontrar la solución (*motor de inferencia*). Los resultados finales y la forma en que se obtienen se expresan a través de la *interface hombre-máquina*<sup>13</sup>.

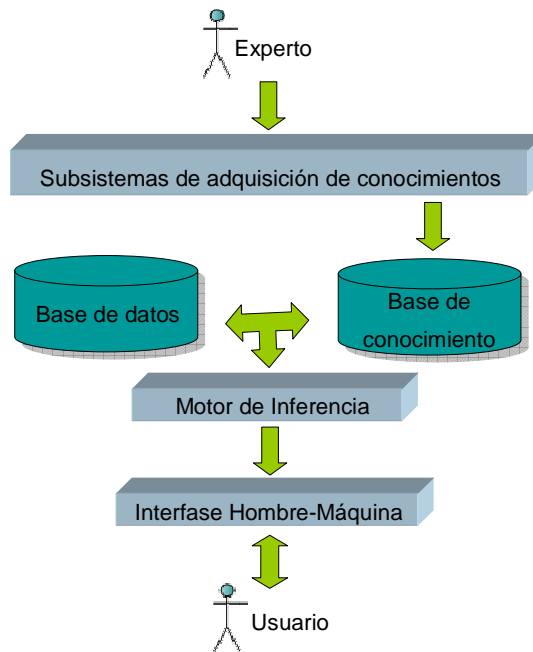
---

<sup>11</sup> WIKIPEDIA. "Inteligencia Artificial". URL: [http://es.wikipedia.org/wiki/Inteligencia\\_artificial](http://es.wikipedia.org/wiki/Inteligencia_artificial)

<sup>12</sup> Rolston, David W. Principios de inteligencia artificial y sistemas expertos. McGraw Hill, Colombia, 1990.

<sup>13</sup> CIDS: Centro de investigación en desarrollo de sistemas. "Sistemas expertos basados en reglas". URL: [http://ci2a.upeu.edu.pe/topicos/topico\\_sistbareglas.htm](http://ci2a.upeu.edu.pe/topicos/topico_sistbareglas.htm)

### 2.3.1. Estructura de un Sistema Experto



**Figura 3.** Elementos que componen el sistema experto.

#### 2.3.1.1. Subsistema de adquisición de conocimientos

El módulo de adquisición del conocimiento permite que se puedan añadir, eliminar o modificar elementos de conocimiento (en la mayoría de los casos reglas) en el sistema experto. Si el entorno es dinámico es muy necesario, puesto que, el sistema funcionará correctamente sólo si se mantiene actualizado su conocimiento. El módulo de adquisición permite efectuar ese mantenimiento, anotando en la base de conocimientos los cambios que se producen<sup>14</sup>.

Todos los conocimientos que se obtienen deben ser estructurados de una forma correcta, todo este conocimiento se almacena en lo que se conoce como la base de conocimientos.

---

<sup>14</sup> Chapa, S. "Arquitectura de Sistemas Expertos".  
URL: [http://delta.cs.cinvestav.mx/~schapa/red/intro\\_lm/node46.html](http://delta.cs.cinvestav.mx/~schapa/red/intro_lm/node46.html)

### 2.3.1.2. Base de Conocimientos

La base de conocimientos contiene el conocimiento especializado extraído del experto en el dominio. Es decir, contiene conocimiento general sobre el dominio en el que se trabaja<sup>15</sup>.

El enfoque más común para representar el dominio del conocimiento que se requiere para un Sistema Experto son las reglas de producción. Estas pueden ser referidas como reglas “situación-acción” o “Si-Entonces”. De esta forma, con frecuencia una base de conocimientos está fraguada principalmente de reglas, las cuales son invocadas por un casamiento de patrones con las particularidades de las tareas circunstanciales que van apareciendo en la base de datos global.

Un ejemplo de regla de producción es:

Si (\*Persona edad 18) and (\*Persona sexo masculino)  
Entonces (\*Persona hacer Servicio militar)

La regla anterior determina que si una persona tiene 18 años y además es de sexo masculino, deberá hacer el servicio militar.

Existen reglas de producción que indican bajo que condiciones deben considerarse unas reglas en vez de otras, estas reglas son conocidas como Meta reglas. Un ejemplo de Meta regla es:

Si (paciente edad\_avanzada) entonces (reglas enfermedades\_ancianos)

La Meta regla anterior determina que si un paciente es de edad avanzada, deberán aplicarse las reglas de enfermedades en ancianos, en lugar de utilizar las de enfermedades en jóvenes por ejemplo.

---

<sup>15</sup> Samper, J. "Sistemas Expertos: El conocimiento al poder". Psicología.com  
URL: [http://www.psicologia.com/articulos/ar-jsamper01\\_2.htm](http://www.psicologia.com/articulos/ar-jsamper01_2.htm)

La fuerza de un Sistema Experto yace en el conocimiento específico del dominio del problema. Casi todos los sistemas existentes basados en reglas contienen un centenar de ellas y generalmente se obtienen de entrevistas con expertos durante un tiempo largo. En cualquier Sistema Experto, las reglas se conectan una a otra por ligas de asociación para formar redes de reglas. Una vez que han sido ensambladas tales redes, entonces tenemos una representación de un cuerpo de conocimientos que es substancial.

Generalmente, un experto tiene muchas reglas de criterio o empíricas por lo que el soporte suele ser incompleto para la evidencia que se dispone. En tales casos una técnica es la de vincular valores numéricos a cada regla para indicar el grado de certidumbre que ésta tiene.

En la operación de Sistemas Expertos los valores de certidumbre se combinan unos a otros a través de la red, considerando también, la certidumbre de los datos de entrada del problema, para llegar finalmente a un valor de certidumbre de la solución final<sup>16</sup>.

Pero no es suficiente con conocer la base de conocimientos para entender la estructura y el funcionamiento de los Sistemas Expertos, otro de los componentes esenciales de estos sistemas, es la base de datos o base de hechos.

### **2.3.1.3. Base de Datos o Base de Hechos**

La base de datos o base de hechos es una parte de la memoria del ordenador que se utiliza para almacenar los datos recibidos inicialmente para la resolución de un problema.

---

<sup>16</sup> Chapa, S. "Arquitectura de Sistemas Expertos".  
URL: [http://delta.cs.cinvestav.mx/~schapa/red/intro\\_lm/node46.html](http://delta.cs.cinvestav.mx/~schapa/red/intro_lm/node46.html)

Contiene conocimiento sobre el caso concreto en que se trabaja. También se registrarán en ella las conclusiones intermedias y los datos generados en el proceso de inferencia. Al memorizar todos los resultados intermedios, conserva el vestigio de los razonamientos efectuados; por lo tanto, se puede utilizar explicar las deducciones y el comportamiento del sistema<sup>17</sup>.

Todos estos datos no son suficientes, si no se tiene un sistema encargado de procesar y manipular toda la información para generar los resultados deseados, este sistema es conocido como Motor de inferencia.

#### **2.3.1.4. Motor de Inferencia**

El motor de inferencia es el "supervisor", un programa que está entre el usuario y la base de conocimientos, y que extrae conclusiones a partir de los datos simbólicos que están almacenados en las bases de hechos y de conocimiento. Dependen en gran medida de la representación elegida; por su importancia analizaremos los motores de inferencia basados en reglas<sup>18</sup>.

Existen varios algoritmos de búsqueda a lo largo de las reglas para inferir conclusiones a partir de los hechos y las reglas. Todos los algoritmos son del tipo "*pattern-matching*", van disparando reglas a medida que se cumplen las condiciones. Se pueden diferenciar dos mecanismos de inferencia: Encadenamiento hacia delante y encadenamiento hacia atrás.

La llamada "técnica de encadenamiento hacia adelante" consiste en aplicar al conocimiento-base (organizado en forma de reglas de producción), junto con otro conocimiento disponible, el esquema inferencial modus ponens. Esta estrategia se denomina "encadenamiento hacia adelante" o "razonamiento de datos dirigidos", porque comienza con los datos conocidos y aplica el modus

---

<sup>17</sup> Samper, J. "Sistemas Expertos: El conocimiento al poder". Psicología.com  
URL: [http://www.psicologia.com/articulos/ar-jsamper01\\_2.htm](http://www.psicologia.com/articulos/ar-jsamper01_2.htm)

<sup>18</sup> Canca, J. "Sistemas Expertos". Esi2.us.es. URL: <http://www.esi2.us.es/~dco/sistemas.htm>

ponens sucesivamente hasta obtener los resultados que se siguen. Las reglas se aplican "en paralelo", o sea, en cualquier iteración una regla toma los datos cuales eran al principio del ciclo, por lo tanto el conocimiento-base y el sistema no dependen del orden en el que las reglas son establecidas, almacenadas o procesadas. Esta técnica suele utilizarse cuando la cantidad de datos es potencialmente muy grande, y resulta de interés algún conocimiento específico tomado en consideración (caso típico en los problemas de diagnóstico; MYCIN, por ejemplo).

La técnica del "encadenamiento hacia atrás" consiste en tratar de probar un dato (o conocimiento) engarzándolo en las reglas-base con el esquema de inferencia modus ponens, o sea, tomando al dato como un consecuente y buscando en el conocimiento-base el correspondiente antecedente, a través de los pasos correspondientes.

Estas dos formas de inferencia se corresponden con los dos métodos lógicos clásicos conocidos por varios nombres: método resolutivo / método compositivo; análisis / síntesis. La distinción se basa en la relación direccional entre objetivos y datos. Y ambas formas pueden combinarse en el razonamiento. Cabe partir de un supuesto inicial, inferir una conclusión mediante un razonamiento hacia adelante y luego establecer un encadenamiento hacia atrás hasta encontrar los datos que confirman esa conclusión<sup>19</sup>.

Es importante destacar que la base de conocimientos y el motor de inferencia son sistemas independientes, por lo que se han desarrollado herramientas que únicamente necesitan la implementación del conocimiento, llevando incorporado el motor de inferencia.

---

<sup>19</sup> Velarde, J. "Sistemas Expertos" . Ucm.es  
URL: [http://www.ucm.es/info/eurotheo/diccionario/S/sistemas\\_expertos.htm](http://www.ucm.es/info/eurotheo/diccionario/S/sistemas_expertos.htm)

La forma en que se obtiene o se presenta la información debe ser tal, que facilite la comprensión de los datos y la forma como se ingresa la información, por lo cuál es necesaria la interface Hombre-Máquina.

#### **2.3.1.5. Interface Hombre-Máquina**

La Interface establece una comunicación sencilla entre el usuario y el sistema. El usuario puede consultar con el sistema a través de menús, gráficos, o algún otro tipo de interfaces, y éste le responde con resultados. "También es interesante mostrar la forma en que extrae las conclusiones a partir de los hechos. En sistemas productivos se cuida la forma de presentar al operador las órdenes obtenidas del sistema experto, debido a que la información excesiva o confusa dificulta la actuación en tiempo real"<sup>20</sup>.

### **2.4. LÓGICA BORROSA**

Lógica borrosa o difusa (*fuzzy logic*) y conjuntos nebulosos o borrosos (*fuzzy sets*).

La lógica borrosa (LB) es una rama de la inteligencia artificial que se fundamenta en el concepto "Todo es cuestión de grado", lo cual permite manejar información vaga o de difícil especificación si quisiéramos hacer cambiar con esta información el funcionamiento o el estado de un sistema específico. Es entonces posible con la lógica borrosa gobernar un sistema por medio de reglas de 'sentido común' las cuales se refieren a cantidades indefinidas.

Para ello se basa en la teoría de conjuntos que posibilita imitar el comportamiento de la lógica humana. El término "borroso" procede de la palabra inglesa "*fuzzy*" que significa "confuso, difuso, indefinido o desenfocado.

---

<sup>20</sup> Canca, J. "Sistemas Expertos". Esi2.us.es. URL: <http://www.esi2.us.es/~dco/sistemas.htm>

Se fundamenta en los denominados conjuntos borrosos y un sistema de inferencia borroso basado en reglas de la forma " SI..... ENTONCES.....", donde los valores lingüísticos de la premisa y el consecuente están definidos por conjuntos borrosos, es así como las reglas siempre convierten un conjunto borroso en otro.

#### **2.4.1. Historia**

La lógica difusa fue investigada, por primera vez, a mediados de los años sesenta en la Universidad de Berkeley (California) por el ingeniero Lotfy A. Zadeh cuando se dió cuenta de lo que él llamó principio de incompatibilidad: "Conforme la complejidad de un sistema aumenta, nuestra capacidad para ser precisos y construir instrucciones sobre su comportamiento disminuye hasta el umbral más allá del cual, la precisión y el significado son características excluyentes". Introdujo entonces el concepto de conjunto difuso (*Fuzzy Set*) bajo el que reside la idea de que los elementos sobre los que se construye el pensamiento humano no son números sino etiquetas lingüísticas. La lógica difusa permite representar el conocimiento común, que es mayoritariamente del tipo lingüístico cualitativo y no necesariamente cuantitativo, en un lenguaje matemático a través de la teoría de conjuntos difusos y funciones características asociadas a ellos. Permite trabajar a la vez con datos numéricos y términos lingüísticos; los términos lingüísticos son inherentemente menos precisos que los datos numéricos pero en muchas ocasiones aportan una información más útil para el razonamiento humano<sup>21</sup>.

Esta teoría nos permite manejar y procesar ciertos tipos de información en los cuales se manejen términos inexactos, imprecisos o subjetivos. De una manera similar a como lo hace el cerebro humano, es posible ordenar un razonamiento basado en reglas imprecisas y en datos incompletos.

---

<sup>21</sup> Conceptos fundamentales de Lógica difusa. URL: [http://www.tdx.cesca.es/TESIS\\_UPC/AVAILABLE/TDX-0207105-105056//04Rpp04de11.pdf](http://www.tdx.cesca.es/TESIS_UPC/AVAILABLE/TDX-0207105-105056//04Rpp04de11.pdf)

Para ello debemos ampliar la teoría de conjuntos y la lógica booleana de manera que un individuo pueda pertenecer parcialmente a un conjunto y que las operaciones lógicas además de unos y ceros, puedan ser 0,01 o 0,75. Nos comunicamos y coordinamos acciones con datos como "...eres demasiado joven para hacer eso..."; Cuanto es "demasiado"?; Que es "joven"?

Con los conjuntos nebulosos (o borrosos, o *fuzzy sets*) podemos definir subconjuntos, de una manera tal que cualquier elemento pueda pertenecer a ellos en diferentes grados.

Con reglas difusas (o *fuzzy rules*), es posible procesar las relaciones entre las variables *fuzzy* y producir una salida nebulosa (o borrosos, o *fuzzy*).

Y lo mejor de todo es que a partir de esas salidas nebulosas, podemos proporcionar cantidades binarias y cantidades continuas, como el estado de un interruptor o una cantidad de dinero.

Las reglas involucradas en un sistema borroso pueden ser desarrolladas con sistemas adaptativos, que aprenden al 'observar' como las personas operan los dispositivos, o simplemente ser formuladas por un experto humano. En general, la lógica borrosa se aplica tanto a sistemas de control como para modelar cualquier sistema continuo de ingeniería, física, biología o economía, definiendo así un sistema matemático que modela funciones no lineales, convirtiendo unas entradas en salidas acordes con los planteamientos lógicos que usan el razonamiento aproximado.

La lógica borrosa es más flexible que la lógica clásica o bivaluada; define la realidad en diferentes grados de verdad siguiendo patrones de razonamiento similares a los del pensamiento humano. Permite también cierta "imprecisión" en la representación de un problema y aún así llegar a una muy buena solución, maneja la incertidumbre y la imprecisión, también se reconocen más que simples valores verdaderos y falsos y las proposiciones pueden ser representadas con grados de veracidad o falsedad. Por ejemplo, la sentencia

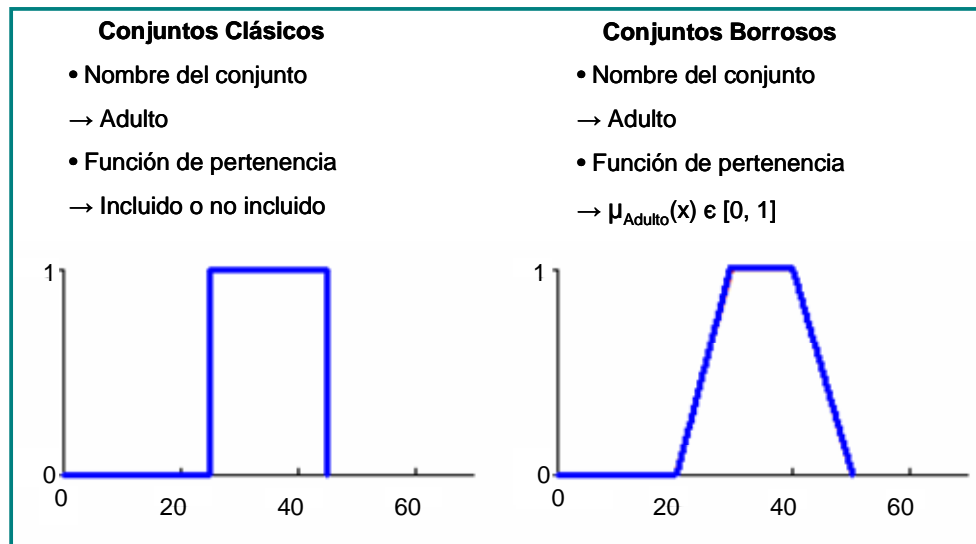
"hoy es un día soleado" puede ser 100% verdad si no hay nubes, 80% verdad si hay pocas nubes, 50% verdad si existe neblina y 0% si llueve todo el día.

Los sistemas de control borrosos tienen una gran variedad de aplicaciones que van desde la estimación de parámetros, toma de decisiones, sistemas mecánicos de control tales como el aire acondicionado o lavadoras automáticas, hasta el control de automóviles o casas "inteligentes". Las nociones como "más bien caliente" o "poco frío" pueden formularse matemáticamente y ser procesados por computadoras. De esta manera, se ha realizado un intento de aplicar una forma más humana de pensar en la programación de computadoras.

#### **2.4.2. Conjuntos borrosos o difusos.**

Los conjuntos difusos pueden ser considerados como una generalización de los conjuntos clásicos: la teoría clásica de conjuntos sólo contempla la pertenencia o no pertenencia de un elemento a un conjunto, sin embargo la teoría de conjuntos difusos contempla la pertenencia parcial de un elemento a un conjunto, es decir, cada elemento presenta un **grado de pertenencia** a un conjunto difuso que puede tomar cualquier valor entre 0 y 1.

El siguiente ejemplo basado en el conjunto "adultos" permite establecer una diferencia entre la lógica borrosa y la lógica clásica. En él se ilustra que con la teoría de la lógica clásica al conjunto "adultos" solo pertenecen hombres con una edad menor a un valor definido, y que cada persona podría dar su apreciación, por ejemplo 20 años; así cualquier hombre menor a esta edad, aunque sea por unos meses, no pertenecería a este conjunto, mientras que, la lógica difusa no le determina al conjunto "adulto" un límite claro y asigna mediante una función que define el cambio de "no adulto" a "adulto" imprimiéndole un grado de pertenencia entre 0 y 1. (**Figura 4**).



**Figura 4.** Comparación entre la lógica clásica y lógica borrosa.

Este grado de pertenencia se define mediante la función característica asociada al conjunto difuso: para cada valor que pueda tomar un elemento o variable de entrada  $x$  la función característica  $\mu_A(x)$  el grado de pertenencia de este valor de  $x$  al conjunto difuso  $A$ .

Desde el punto de vista formal el concepto de conjunto difuso es una generalización del concepto clásico de conjunto. La diferencia fundamental estriba en que, mientras que en la teoría clásica de conjuntos un determinado elemento puede pertenecer a un conjunto o no hacerlo, en la teoría de conjuntos difusos un elemento puede pertenecer a más de un conjunto con diferentes grados de pertenencia.

Entonces, el conjunto  $A$  es matemáticamente equivalente a su función de pertenencia  $\mu_A(x)$ , por lo tanto, conocer  $\mu_A(x)$  es lo mismo que conocer  $A$ .

Un conjunto difuso en el universo de discurso  $U$  se caracteriza por una función de pertenencia  $\mu_A(x)$  que toma valores en el intervalo  $[0,1]$ , y puede representarse como un conjunto de pares ordenados de un elemento  $x$  y su valor de pertenencia al conjunto:

$$A = \{(x, \mu_A(x)) \mid x \in U\} \quad (1)$$

Donde:

$A$ : es el nombre del conjunto. Puede ser cualquier letra mayúscula.

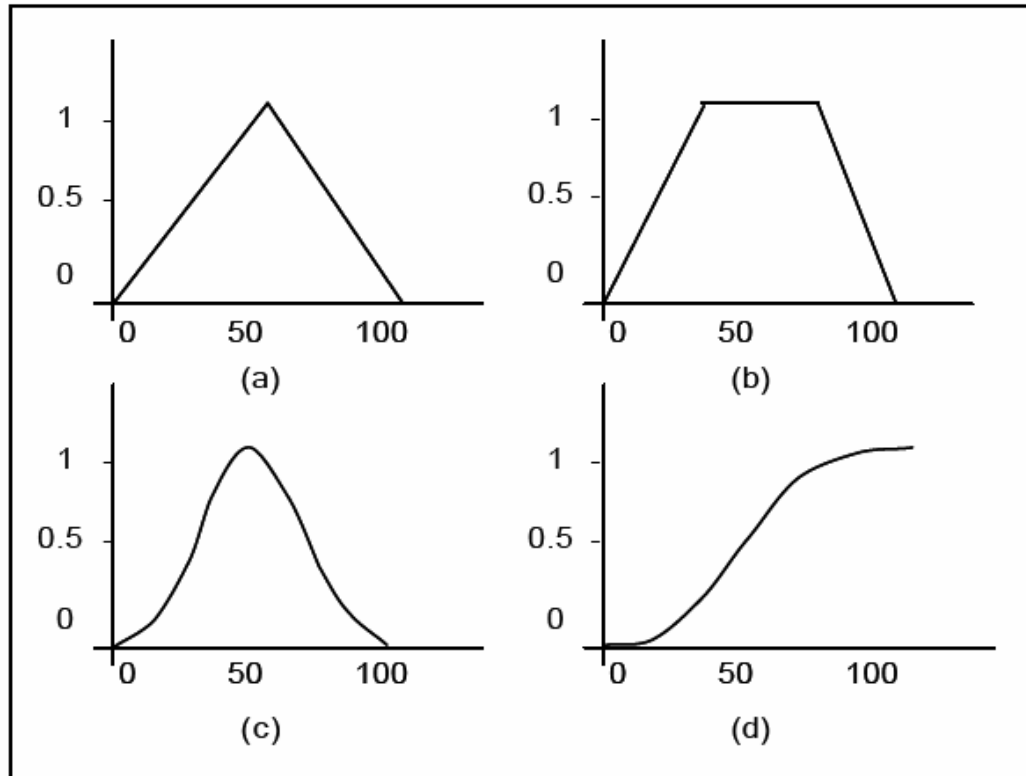
$\mu_A$ : es la función de membresía del conjunto, define el grado de pertenencia de un elemento al conjunto.

$U$  : Universo del discurso.

La función característica proporciona una medida del grado de similaridad de un elemento de  $U$  con el conjunto difuso. La forma de la función característica utilizada, depende del criterio aplicado en la resolución de cada problema y variará en función de la cultura, geografía, época o punto de vista del usuario. La única condición que debe cumplir una función característica es que tome valores entre 0 y 1, con continuidad. Las funciones características más comúnmente utilizadas por su simplicidad matemática y su manejabilidad son: triangular, trapezoidal, gaussiana, sigmoideal, gamma, pi, campana etc... (figura 5). Conceptualmente existen dos aproximaciones para determinar la función característica asociada a un conjunto: la primera aproximación está basada en el conocimiento humano de los expertos, y la segunda aproximación es utilizar una colección de datos para diseñar la función.

El número de funciones características asociadas a una misma variable es elegido por el experto: a mayor número de funciones características tendremos mayor resolución pero también mayor complejidad computacional; además estas funciones pueden estar solapadas o no, el hecho de estar solapadas pone de manifiesto un aspecto clave de la lógica difusa: una variable puede

pertenecer con diferentes grados a varios conjuntos difusos a la vez , es decir, “el vaso puede estar medio lleno y medio vacío a la vez”.



**Figura 5.** Funciones características más habituales: (a) triangular, (b) trapezoidal, (c) gaussiana y (d) sigmoideal

### 2.4.3. Propiedades de los conjuntos borrosos.

Las leyes y propiedades que cumplen los conjuntos clásicos no siempre se cumplen en el caso de los conjuntos borrosos. A continuación se analizará qué leyes verifican los conjuntos borrosos y cuales no:

1. Propiedad conmutativa: Siempre se verifica, debido a que las t-normas y las t-conormas son conmutativas por definición.
2. Propiedad asociativa: también se verifica, debido a que las t-normas y las t-conormas también son asociativas conmutativas por definición.

3. Leyes de idempotencia: se cumplen si se elige el mínimo y el máximo como operadores para la intersección y la unión respectivamente. Pero si se escoge por ejemplo el producto algebraico y la suma algebraica, no se cumplen.
4. Leyes de absorción: se cumplen si se elige el mínimo y el máximo como operadores para la intersección y la unión respectivamente. Con otras normas no ocurre necesariamente lo mismo.
5. Propiedad distributiva: se cumplen si se elige el mínimo y el máximo como operadores para la intersección y la unión respectivamente. Con otras normas no ocurre necesariamente lo mismo.
6. Propiedad de absorción e identidad: siempre se cumplen por la última propiedad de t-norma y t-conorma.
7. Involución del complemento: se cumple si el operador elegido es el estándar.
8. Leyes de Morgan: se garantiza su cumplimiento si la t-norma y la s-norma elegidas, derivan una de la otra. Es decir,  $T(x, y) = 1 - S(1 - x, 1 - y)$ .
9. Leyes complementarias: no se cumplen. Es quizás la consecuencia más clara de introducir la borrosidad a los conjuntos.

Se puede comprobar fácilmente que en el caso de que los conjuntos sean clásicos (función de pertenencia restringida a 0 o 1) las diferencias entre las diversas normas desaparecen, convirtiéndose en los operadores intersección y unión clásicos.

#### **2.4.4. Funciones de pertenencia o membresía.**

Las funciones de pertenencia simbolizan la manera gráfica de representar un conjunto borroso. Estas funciones son expresadas por medio de fórmulas matemáticas, son definidas para el espacio unidimensional, bidimensional y multidimensional.

Las más utilizadas son las triangulares, trapezoidales, tipo campana de Gauss, tipo curva de Bell, singleton, sigmoïdal, entre otras.

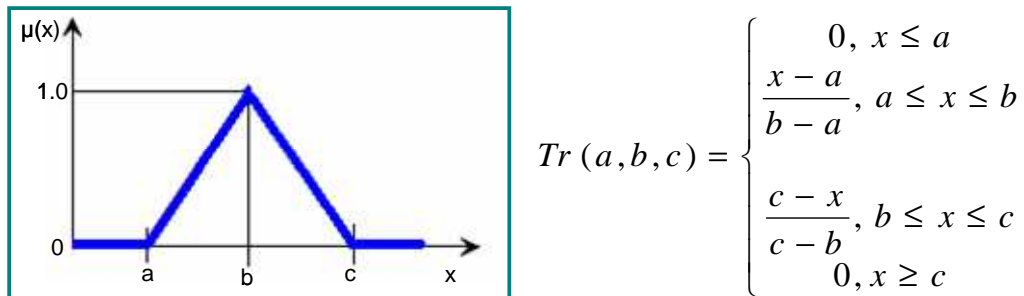
$$\mu_A(x) \rightarrow [0,1]$$

La función de membresía es de tal modo que  $\mu_A(x) \in [0,1]$  es el grado con el que un elemento  $x \in \mu$  pertenece al conjunto borroso **A**.

Cuando  $\mu_A(x) = 0$  el elemento no pertenece al conjunto, y si  $\mu_A(x) = 1$  pertenece totalmente. A continuación se mostraran las funciones de membresía más comunes:

#### 2.4.4.1. Función de Pertenencia Triangular.

Una función de pertenencia triangular se define en función de los vértices a través de tres parámetros (a,b,c) de la siguiente manera:



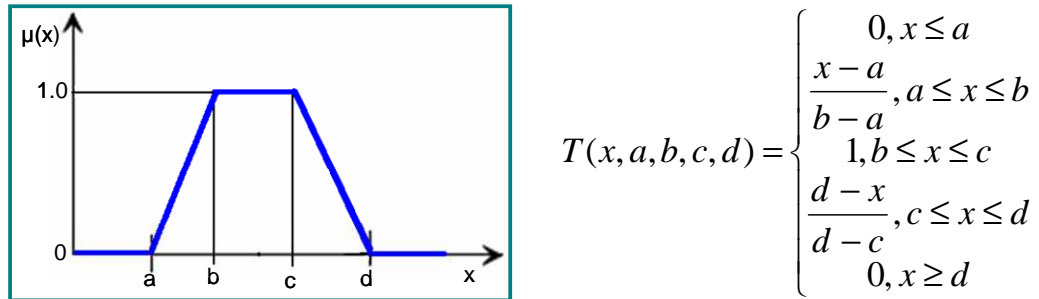
**Figura 6.** Función de Pertenencia Triangular.

Los parámetros a, b, c determinan los tres ángulos de la función de pertenencia triangular, con  $(a,b,c) > 0$  y  $a < b < c$ .

#### 2.4.4.2. Función de Pertenencia Trapezoidal

La función de pertenencia trapezoidal se define en función de los vértices a través de cuatro parámetros (a, b, c, d).

Su forma gráfica quedaría así:



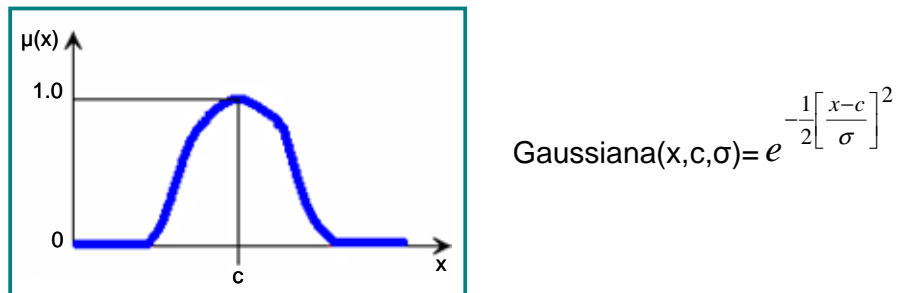
**Figura 7.** Función de Pertenencia Trapezoidal

Los parámetros (a, b, c, d) determinan las coordenadas de los cuatro ángulos de la función de pertenencia trapezoidal, con (a, b, c, d) > 0, y a < b < c < d.

Las funciones de pertenencia triangular y trapezoidal son las más utilizadas en las aplicaciones industriales reales pero presentan la desventaja de no ser muy suaves en los puntos donde se forman los ángulos.

#### 2.4.4.3. Función Gaussiana o Campana de Gauss.

La función de pertenencia tipo gaussiana o campana de gauss se determina por dos parámetros (c, σ). Su forma gráfica y su ecuación serían así:



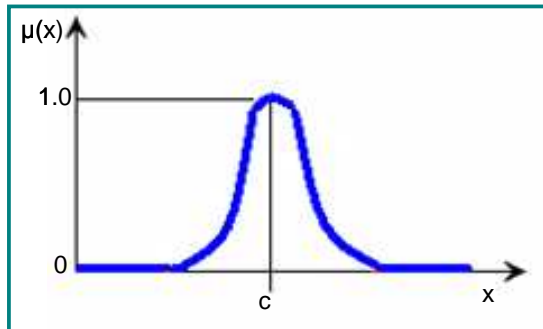
**Figura 8.** Función de pertenencia Gaussiana.

Las funciones pertenencia tipo campana de gauss se determinan completamente por los parámetros  $\mathbf{c}$  y  $\sigma$ , donde  $\mathbf{c}$  representa el centro y  $\sigma$  representa el ancho de la función de pertenencia.

#### 2.4.4.4 Función de Pertenencia Curva de Bell

La función de pertenencia generalizada de Bell se determina por tres parámetros (a, b, c).

La forma gráfica y la expresión matemática quedarían así:



$$Bell(x,a,b,c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2*b}}$$

donde  $b > 0$

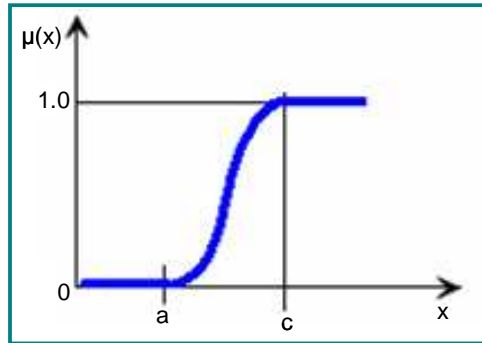
**Figura 9.** Función de Pertenencia Curva de Bell

La función de pertenencia es una generalización directa de la distribución de Cauchy utilizada en la teoría de las probabilidades por eso también se le conoce como función de pertenencia de Cauchy, una función de pertenencia particular de Bell se puede obtener a través de una selección correcta del conjunto de parámetros (a, b, c). Específicamente podemos ajustar  $\mathbf{c}$  y  $\mathbf{a}$  para variar el centro y el ancho de la función de pertenencia respectivamente y utilizamos  $\mathbf{b}$  para controlar la inclinación de los puntos extremos.

#### 2.4.4.5. Función de Pertenencia Tipo Sigmoidal

El parámetro a controla la inclinación del punto de transición  $x = \mathbf{c}$ , esta función de pertenencia se utiliza en las redes neuronales como función de activación; y para simular el comportamiento del sistema de inferencia borroso.

La gráfica de la función de pertenencia tipo Sigmoidal y su formula serian así:



$$\text{Sig}(x; a, c) = \frac{1}{1 + e^{[-a(x-c)]}}$$

**Figura 10.** Función de Pertenencia tipo sigmoidal.

Aparte de las funciones de pertenencia anteriormente mencionadas, también se pueden crear otro tipo de funciones especializadas y de cualquier tipo según la necesidad practica y especifica de cada aplicación.

#### 2.4.5. Operaciones entre conjuntos borrosos.

Al igual que para los conjuntos clásicos o convencionales, están definidas las operaciones básicas para conjuntos borrosos como son la intersección, unión, complemento e inclusión entre otras. Dentro la teoría de los conjuntos borrosos tiene especial relevancia la que hace uso de operaciones conocidas como operaciones estándar, definidas como:

$$\text{Intersección: } \mu_A \cap \mu_B(x) = \text{mín} [\mu_A(x), \mu_B(x)] \quad (2)$$

$$\text{Unión: } \mu_{A \cup B}(x) = \text{max} [\mu_A(x), \mu_B(x)] \quad (3)$$

$$\text{Complemento: } \mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (4)$$

Esta no es la única forma posible de definir estas operaciones, diferentes funciones pueden ser apropiadas para representarlas en diferentes contextos.

En la figura 11 se puede ver la representación gráfica de estas operaciones.

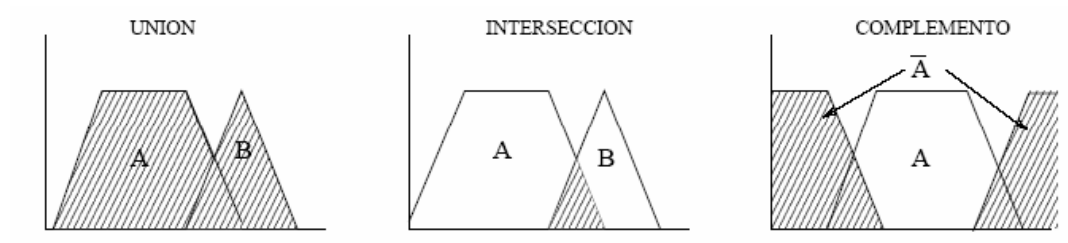


Figura 11. Operaciones borrosas<sup>22</sup>.

### 2.4.5.1. Complemento borroso: negación

Dado un conjunto borroso  $A \subset U$ , se define su complemento como el conjunto

borroso  $\bar{A}$  cuya función de pertenencia viene dada por la expresión:

$$\mu_{\bar{A}}(x) = C(\mu_A(x)) = 1 - \mu_A(x), \forall x \in U \quad (5)$$

donde  $C(x)$  es una función que debe cumplir las siguientes propiedades:

1. Condiciones de contorno:  $C(0) = 1, C(1) = 0$ .
2. Monotonía: para todo  $a; b \in [0; 1]$ , si  $a > b$ , entonces  $C(a) < C(b)$

La función  $C(x)$  también es conocida como **c-norma**.

En la mayoría de los casos, es deseable considerar algunos requerimientos adicionales para estas funciones:

3.  $C(x)$  es una función continua.
4.  $C(x)$  es involutiva, lo que significa que  $C(C(a)) = a \quad \forall a \in [0,1]$ .

<sup>22</sup> Aja F, Santiago. Tesis de doctorado. Un marco matricial para la implementación de la inferencia borrosa aplicados al procesado de información no numérica. Universidad de Valladolid. España. 2003.

Existen muchas funciones que cumplen las propiedades antes descritas, y que por lo tanto pueden ser usadas para representar el complemento borroso. Algunas de ellas son:

$$C(x) = 1 - x \quad \text{Negación} \quad (6)$$

$$C(x) = \frac{1-x}{1-\lambda x}, \lambda \in (0, \infty) \quad \text{Sugeno} \quad (7)$$

$$C(x) = (1-x^w)^{1/w}, w \in (0, \infty) \quad \text{Yager} \quad (8)$$

#### 2.4.5.2. Intersección borrosa: t-norma.

Dados dos conjuntos borrosos A y B, definidos sobre un mismo universo del discurso U, se define su intersección como un conjunto borroso  $A \cap B$  cuya función de pertenencia viene dada por la expresión.

$$\begin{aligned} \mu_{A \cap B}(x) &= T[\mu_A(x), \mu_B(x)] = \text{Mín} [\mu_A(x), \mu_B(x)] \\ &= \mu_A(x) \wedge \mu_B(x) \quad \forall x \in U \end{aligned} \quad (9)$$

donde la función T(x,y) es una norma triangular o t-norma. Una t-norma es una ampliación  $T: [0,1] \times [0,1] \rightarrow [0,1]$  que verifica las siguientes propiedades:

1. Conmutativa:  $T(x, y) = T(y, x), \forall x, y \in [0,1]$ .
2. Asociativa:  $T(T(x, y), z) = T(x, T(y, z)), \forall x, y, z \in [0,1]$ .
3. Monotonía: si  $(x \leq y)$  y  $(w \leq z)$  entonces  $T(x, w) \leq T(y, z), \forall x, y, w, z \in [0,1]$ .
4. Elemento absorbente:  $T(x, 0) = 0, \forall x \in [0,1]$ .
5. Elemento neutro:  $T(x, 1) = x, \forall x \in [0,1]$ .

Existen muchas funciones que cumplen estas propiedades, por lo tanto pueden ser usadas para representar la intersección entre conjuntos borrosos. Algunas de estas son:

$T(x, y) = \min(x, y)$	Mínimo.
$T(x, y) = \max(0, x + y - 1)$	Diferencia acotada.
$T(x, y) = x \cdot y$	Producto algebraico
$T(x, y) = \begin{cases} \min(x, y) & \text{si } \max(x, y) = 1 \\ 0 & \text{en cualquier otro caso} \end{cases}$	Producto drástico

En ocasiones es necesario restringir las posibles t-normas considerando tres requerimientos adicionales que son:

1. Continuidad:  $T()$  es un afición continua.
2. Subidempotencia:  $T(x, x) < x$ .
3. Monotonía estricta:  $a_1 < a_2$  y  $b_1 < b_2$  implica  $T(a_1, b_1) < T(a_2, b_2)$ .

También podemos encontrar la intersección borrosa expresada de la siguiente forma:

$$\mu_A \cap \mu_B(x) = \mu_A(x) \wedge \mu_B(x) \quad (10)$$

donde  $\wedge$  representa al operador mínimo (intersección borrosa).

### 2.4.5.3. Unión borrosa: t-conorma

Dados dos conjuntos borrosos A y B definidos sobre el mismo universo de discurso  $U$ , se define su unión como un conjunto borroso  $A \cup B$  cuya función de pertenencia viene dada por la expresión:

$$\mu_{A \cup B}(x) = S(\mu_A, \mu_B) = \text{Max}(\mu_A(x), \mu_B(x)), \forall x \in U \quad (11)$$

donde la función  $S(x, y)$  es una conorma triangular, también llamada *t-conorma* o *s-norma*. Es una aplicación  $S : [0, 1] \times [0, 1] \longrightarrow [0, 1]$  que satisface los siguientes requisitos:

1. Conmutatividad:  $S(x, y) = S(y, x), \forall x, y \in [0,1]$
2. Asociatividad:  $S(x, S(y, z)) = S(S(y, x), z), \forall x, y, z \in [0,1]$
3. Monotonidad: si  $(x \leq y) \wedge (w \leq z)$  entonces  $S(x, w) \leq S(y, z), \forall x, y, w, z \in [0,1]$
4. Elemento absorbente:  $S(x,1) = 1, \forall x \in [0,1]$
5. Elemento neutro:  $S(x,0) = x, \forall x \in [0,1]$

Al igual que en los casos anteriores, existe un gran número de funciones que cumplen estas propiedades y que pueden ser utilizadas para representar la unión. Algunos ejemplos son:

$S(x; y) = \max(x; y)$	Máximo
$S(x; y) = \min(1; x + y)$	Suma acotada
$S(x; y) = x + y \text{ si } x \notin y$	Suma algebraica
$S(x, y) = \begin{cases} \max(x, y) & \text{si } \min(x, y) = 0 \\ 1 & \text{en cualquier otro caso} \end{cases}$	Suma drástica

En ocasiones es necesario restringir las posibles t-conormas considerando tres requerimientos adicionales, que tengan en cuenta casos especiales, tal y como se hizo para la t-norma:

1. Continuidad:  $S()$  es una función continua.
2. Superidempotencia:  $S(x; x) > x$
3. Monotonía estricta:  $a_1 < a_2$  y  $b_1 < b_2$  implica  $S(a_1; b_1) < S(a_2; b_2)$

Se puede encontrar en la bibliografía la unión de dos conjuntos borrosos expresada como:

$$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) \quad (12)$$

donde  $\vee$  representa al operador máximo (unión borrosa estándar).

## 2.4.6. Modificadores Lingüísticos

Son operadores unitarios que se aplican a conjuntos difusos aplicando una operación que modifica el significado de un término, o de manera más general de un conjunto borroso. Por ejemplo, si un conjunto borroso denota presión débil, entonces presión muy débil, presión más o menos débil, presión extremadamente débil son ejemplos de modificadores aplicados a este conjunto. Los modificadores se pueden ver como operadores que actúan sobre la función de pertenencia de un conjunto borroso para modificarla.

Sea  $A$  un valor lingüístico caracterizado por un conjunto borroso con una función de pertenencia  $\mu_A(x)$  luego  $A^K$  se puede interpretar como una versión modificada del valor lingüístico original expresado como:

$$A^k = \int_x [\mu_A(x)]^k / x \quad (13)$$

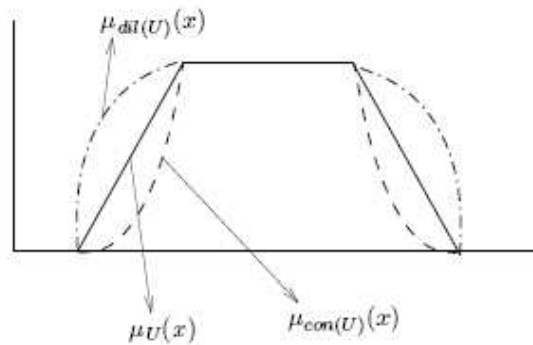
Algunos de estos operadores pueden ser:

1. Concentración:  $\mu_{con(U)}(x) = [\mu_U(x)]^2 \quad (14)$

2. Dilatación:  $\mu_{dil(U)}(x) = [\mu_U(x)]^{1/2} \quad (15)$

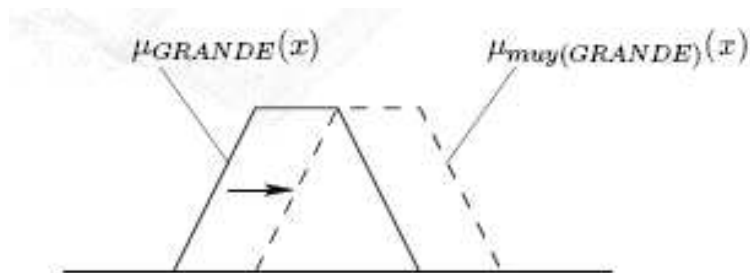
3. Modificadores artificiales:  $\mu_{plus(U)}(x) = [\mu_U(x)]^{1.25} \quad (16)$

$$\mu_{minus(U)}(x) = [\mu_U(x)]^{0.75} \quad (17)$$



**Figura 12.** Modificadores aplicados sobre un conjunto borroso<sup>23</sup>.

En la figura 13 se muestra gráficamente el efecto de los dos primeros modificadores sobre un conjunto borroso con función de pertenencia trapezoidal. Además de cambiar la forma de las funciones de pertenencia, algunos modificadores también cambian la posición de los conjuntos sobre el eje de ordenadas; por ejemplo el modificador muy aplicado sobre el conjunto borroso GRANDE desplaza a esta hacia la derecha como lo muestra la siguiente figura 14:



**Figura 13.** Modificador Lingüístico

#### 2.4.7. Variable Lingüística

Es una variable cuyos valores son palabras o sentencias que se enmarcan en lenguaje natural. Cada una de estas palabras o términos se conoce como

<sup>23</sup> Aja F, Santiago. Tesis de doctorado. Un marco matricial para la implementación de la inferencia borrosa aplicados al procesamiento de información no numérica. Universidad de Valladolid. España. 2003.

etiqueta lingüística y se representa por medio de un conjunto borroso definido sobre el universo de discurso de la variable.

Una variable lingüística se caracteriza por una quintupla  $(\mathbf{A}; \mathbf{T}(\mathbf{A}); \mathbf{U}; \mathbf{G}; \mathbf{M})$  en la que:

$\mathbf{A}$  es el nombre de la variable;

$\mathbf{T}(\mathbf{A})$  (o simplemente  $\mathbf{T}$ ) denota el conjunto de términos de  $\mathbf{A}$ , es decir, el conjunto de nombres de los valores lingüísticos de  $\mathbf{A}$ ,

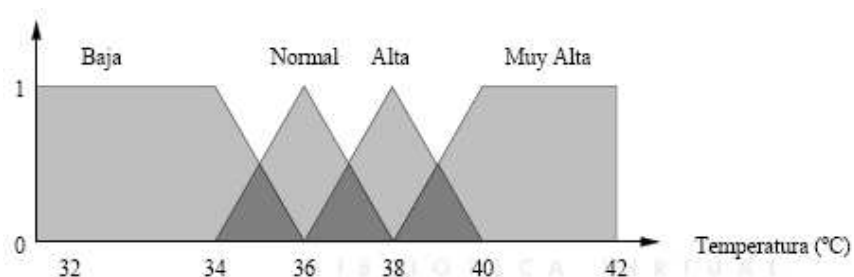
$\mathbf{X}$  es el universo de discurso de la variable  $\mathbf{A}$ , que toma valores en un universo de discurso  $\mathbf{U}$  asociada a una variable base  $\mathbf{u}$ ;

$\mathbf{G}$  es una regla sintáctica para términos lingüísticos (que suele tomar la forma de una gramática) para los nombres de los valores de  $\mathbf{A}$ , y

$\mathbf{M}$  es una regla semántica para asociar su significado a cada  $\mathbf{A}$ ,  $\mathbf{M}(\mathbf{X})$ , que es un subconjunto borroso de  $\mathbf{U}$ .

Por ejemplo, la temperatura del cuerpo humano puede catalogarse como 'baja', 'normal', 'alta' o 'muy alta'. Cada uno de estos términos es una etiqueta lingüística que puede definirse como un conjunto borroso.

Los conjuntos que forman la variable se muestran en la Figura 15.



**Figura 14.** Definición de la variable lingüística Temperatura.

## 2.4.8. Proceso de inferencia borrosa

Al contrario que en Lógica Clásica, en Lógica Borrosa el razonamiento no es preciso, sino que este tiene lugar de una manera aproximada. Esto quiere decir que se puede inferir una conclusión aunque el hecho no verifique la regla plenamente ("Razonamiento Aproximado"). Dicha conclusión se parecerá tanto mas a la conclusión formal de la regla original cuanto mayor sea el grado de cumplimiento de la regla por parte del hecho.

El razonamiento aproximado se resume generalmente, por extensión del razonamiento clásico, en los esquemas de *modus ponens* generalizado y *modus tollens* generalizado, que veremos a continuación.

### 2.4.8.1. Proposiciones borrosas

En una proposición borrosa, a diferencia de lo que ocurre en las proposiciones clásicas, el grado de verdad puede estar en un valor intermedio entre uno y cero. Hay un tipo de proposición borrosa que será particularmente útil, llamada proposición borrosa incondicional y no cualificada. Se define como:

p: Si X es A entonces Y es B

donde X e Y son variables con valores en los universos U y V respectivamente y A y B son conjuntos borrosos sobre U y V respectivamente. Esta expresión se puede escribir más formalmente por medio de la relación:

$\langle X; Y \rangle$  es R

donde R es un conjunto borroso definido en  $U \times V$ , y la función de pertenencia de cada una se puede derivar de cada punto  $(x; y) \in U \times V$  por medio de una implicación borrosa

$$R(x, y) = \tau [A(x), B(y)] \quad (18)$$

Hay muchas definiciones de la implicación superior. Uno de ellos, es el conocido como la implicación de Lukasiewicz, definida por:

$$\tau [A(x),B(y)] = \min(1, 1 - A(x) + B(y)) \quad (19)$$

#### 2.4.8.2. *Modus Ponens* Generalizado (MPG)

En este esquema, se expresa una regla, se da un hecho y a partir de ambos se obtiene una conclusión.

Regla: Si  $X$  es  $A$  entonces  $Y$  es  $B$   
 Hecho:  $X$  es  $A'$   
 Conclusión:  $Y$  es  $B'$  (MPG)

donde  $A$ ,  $B$ ,  $A'$  y  $B'$  son conjuntos borrosos. La conclusión  $B'$  es un conjunto borroso caracterizado por una generalización del *Modus Ponens* propuesto por Zadeh:

$$\mu_{B'}(y) = \max_{x \in U} [\min(\mu_{A'}(x), R(x, y))] \quad (20)$$

siendo  $R(x, y)$  la implicación definida en la ecuación. De hecho, si el conjunto borroso  $A_0$  se reduce al elemento  $x_0$ , entonces:

$$\mu_{A'} = \begin{cases} 1 & \text{para } x = x_0 \\ 0 & \text{para } x \neq x_0 \end{cases}$$

tomando la implicación  $R(x; y)$  por la correspondiente de Mamdani, se obtiene:

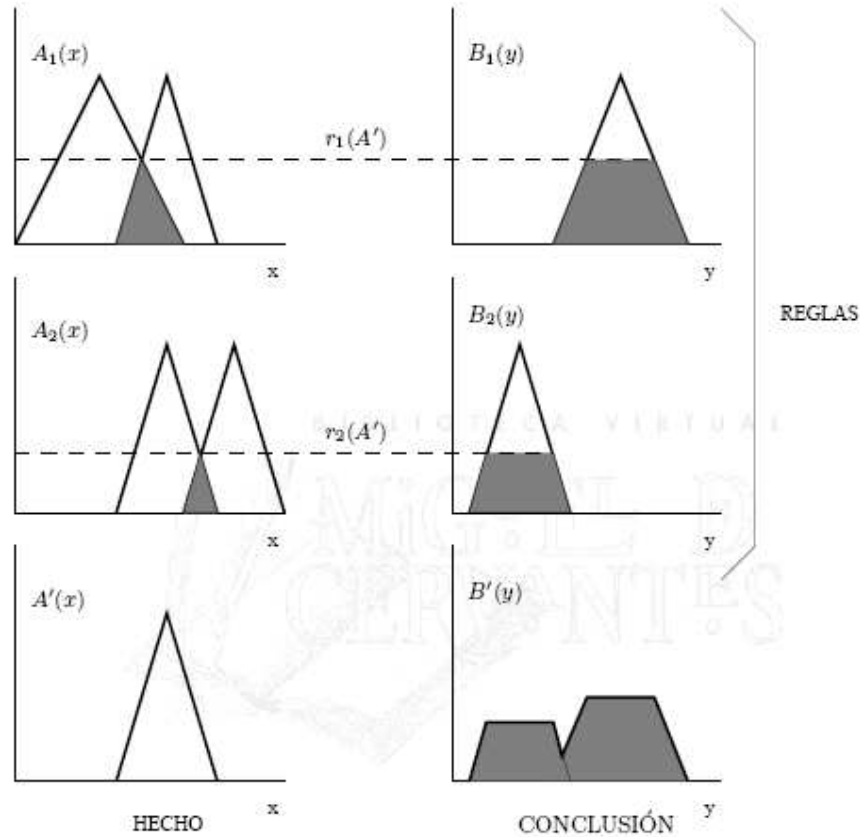
$$\mu_{B'}(y) = \min(\min(\mu_{A'}(x_0), \mu_B(y))) \quad (21)$$

#### 2.4.8.3. *Modus Tollens* Generalizado (MTG)

En el *Modus Tollens* Generalizado (MTG) se cumple que:



$$dc_j(A') = \max_{x \in U} \min[(A'(x), A_j(x))] \quad (25)$$



**Figura 15.** Representación gráfica del método de interpolación.

2. Calcular la conclusión  $B'$  truncando cada conjunto  $B_j$  por el valor  $r_j(A_j)$  y tomar la unión de los conjuntos truncados (en caso de que las reglas sean disyuntivas)

$$B'(y) = \bigcup_j (dc_j(A') \cap B_j(y)) \forall y \in Y \quad (26)$$

que usando las operaciones estándar resulta ser:

$$B'(y) = \max_j \min[dc_j(A'), B_j(y)] \forall y \in Y \quad (27)$$

Para el caso de dos entradas, la base de reglas tendrá dos antecedentes,  $A_1(y)$  y  $B_1(y)$ , por lo que el  $dc$  debe ser calculado de otra forma; se asumirá

que el  $dc$  es el mínimo (o de manera mas general, otra t-norma) de los dos niveles de activación de cada antecedente por cada entrada. Específicamente

$$dc_j(X, Y) = dc_j(X) \cap dc_j(Y) \quad (28)$$

que usando el mínimo resulta:

$$dc_j(X, Y) = \min(dc_j(X), dc_j(Y)) \quad (29)$$

Esta expresión puede extenderse para el caso de múltiples antecedentes, que para una t-norma genérica resulta

$$(dc_i)_j = \bigcap_i (dc_j(Y_i)) \quad (30)$$

donde  $Y_i$  son las entradas al sistema.

## 2.5. SISTEMA DE INFERENCIA BORROSA (FIS)

Un sistema de inferencia borrosa es la aplicación de la inferencia borrosa a la automatización de procesos, esto es, al proceso de extraer una conclusión a partir de ciertas premisas y un conjunto de reglas **El Sistema de Inferencia Borroso** es una plataforma popular de computo basada en los conceptos de la teoría de los conjuntos borrosos, las reglas borrosas IF – THEN, y el razonamiento borroso.

La estructura básica de un sistema de inferencia borroso consta de unos componentes conceptuales, **La base de conocimiento o diccionario**, la cual define las funciones de pertenecía que se utilizan en las reglas borrosas, **la Base de Reglas**, es la que contiene el conjunto de acciones a realizar en función del estado, y **el Mecanismo de Inferencia**, el cual realiza el procedimiento de inferencia sobre las reglas borrosas y los hechos o datos suministrados sobre las reglas borrosas para producir una respuesta o conclusión.

El sistema de inferencia difuso se conoce por muchos nombres como: sistema difuso de reglas, sistema experto difuso, modelo difuso, lógica asociativa difusa, controlador difuso, sistema de lógica borrosa (SLB). Ellos trabajan de una forma bastante diferente a los controladores convencionales; el conocimiento experto se usa en vez de ecuaciones diferenciales para describir un sistema. Este conocimiento puede expresarse de una manera muy natural, empleando las variables lingüísticas que son descritas mediante conjuntos borrosos.

### **2.5.1. Tipos de sistemas basados en reglas borrosas**

Existen tres clases de sistemas basados en reglas borrosas (SBRB), de acuerdo con la forma de las reglas y del tipo de entradas y salidas:

#### **2.5.1.1. Sistemas puros:**

Estos sistemas tienen como entrada y como salida conjuntos borrosos. Al no realizar ninguna transformación sobre las entradas o sobre las salidas, tienen sólo dos componentes principales: **una base de conocimiento y un motor de inferencia.**

Las reglas lingüísticas empleadas son de la forma:

**SI  $X_1$  es  $A_1$  y.... $X_n$  es  $A_n$  ENTONCES  $T$  ES  $B$**

Donde  $X_i$  e  $Y$  son variables lingüísticas, y los  $A_i$  y  $B$  son etiquetas lingüísticas asociadas a conjuntos borrosos.

#### **2.5.1.2. Sistemas borrosos tipo Mamdani**

Este tipo de sistemas fue propuesto por Mamdani, quien fue capaz de traducir las teorías borrosas propuestas por Zadeh en el primer sistema borroso

aplicado a un problema de control. En esta clase de sistemas, la más usada dentro de los sistemas borrosos, se conoce también con el nombre de *controladores borrosos*.

Un sistema tipo Mamdani se corresponde con la noción más conocida de *sistema borroso*; está compuesto por una base de conocimiento, un motor de inferencias, y unos interfaces de borrosificación y desborrosificación (Fig. 16). Estos sistemas tienen una serie de características interesantes :

- Pueden ser usados en aplicaciones del mundo real, ya que manejan con facilidad entradas y salidas reales.
- Proporcionan un marco natural para la inclusión de conocimiento experto en forma de las reglas lingüísticas.
- Tienen gran libertad a la hora de elegir los interfaces de borrosificación y desborrosificación.

Por el contrario presentan también una serie de limitaciones:

- Falta de flexibilidad en el SBRB debido a la forma tan rígida en la que se particionan los espacios de entrada y salida.
- No existe una distinción clara entre el conocimiento experto y la definición de las variables lingüísticas incluidas en las reglas borrosas.
- Cuando las variables de entrada al sistema dependen unas de otras, es muy complicado obtener una partición borrosa adecuada de los espacios de entrada.
- El tamaño de la base de conocimiento depende directamente del número de variables y términos lingüísticos que existan en el sistema.

Las reglas que manejan esta clase de sistemas son de la forma:

**SI  $X_1$  es  $A_1$  y ...  $X_n$  es  $A_n$  es  $A_n$  ENTONCES  $Y$  es  $B$**

donde las entradas  $X_i$  y la salida  $Y$  son ahora números (no borrosos), en lugar de términos lingüísticos (como en el caso anterior), y por lo tanto, los  $A_i$  y  $B$  son conjuntos borrosos sin interpretación directa, en lugar de etiquetas lingüísticas.

### 2.5.1.3. Sistemas borrosos tipo Takagi-Sugeno-Kang

En lugar de trabajar con reglas lingüísticas como las de la sección anterior, Tak-agi, Sugeno y Kang propusieron un nuevo modelo basado en reglas donde el antecedente estaba compuesto de variables lingüísticas y el consecuente se representaba como una función de las variables de entrada. La forma mas habitual de esta clase de reglas es la siguiente:

$$\text{SI } X_1 \text{ es } A_1 \text{ y } \dots X_n \dots \text{ ENTONCES } Y = p_1 X_1 + p_2 X_2 + \dots p_n X_n + p_0$$

siendo  $X_{1i}$  las variables de entrada,  $Y$  la variable de salida, y  $p_i$  parámetros reales. Esta clase de reglas se conocen como reglas TSK en referencia a sus creadores.

La salida de un sistema borroso TSK que usa una base de conocimiento con  $m$  reglas se obtiene como la media ponderada de las salidas individuales proporcionadas por cada regla,  $Y_i$ , ( $i = 1, \dots, m$ ), como sigue

$$\text{Salida} = \frac{\sum_{i=1}^m h_i Y_i}{\sum_{i=1}^m h_i} \quad (31)$$

siendo  $h_i = T(A_1^i(x_1), \dots, A_n^i(x_n))$  el grado de emparejamiento entre la parte antecedente de la regla  $i$  y las entradas actuales al sistema  $x = (x_1, \dots, x_n)$ .  $T$  es un operador de conjunción que se modela mediante una t-norma.

## 2.5.2 Elementos de un Sistema de Inferencia Borrosa.

En la figura 17 aparecen los elementos principales de un Sistema de Lógica Borrosa (SLB). Se ha escogido como modelo un sistema tipo Mamdani por ser este el más usado en la literatura. Contiene cuatro componentes fundamentales:

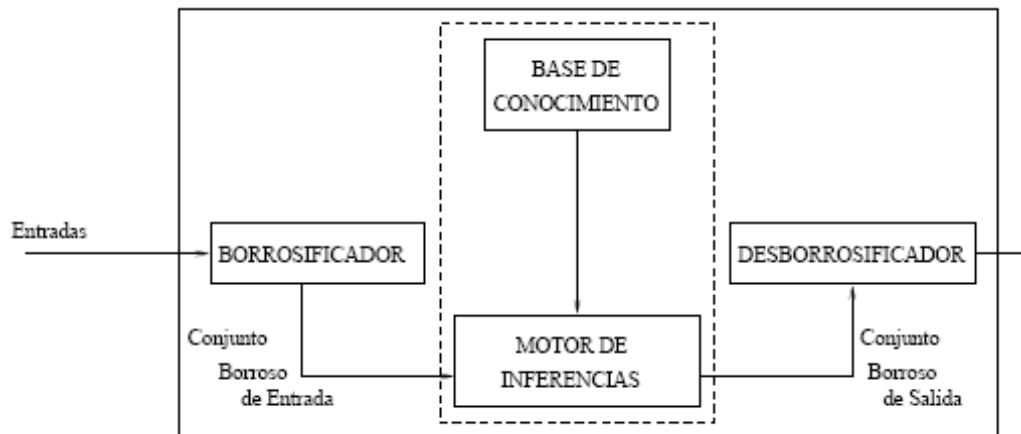


Figura 16. Sistema de Lógica Borrosa

### Interface de Borrosificación

Que realiza un escalado de los valores de las entradas para adecuarlos a los valores típicos para los que se define el sistema, y una "borrosificación" que convierte los datos de entrada en valores lingüísticos adecuados para la manipulación de estos como entidades borrosas.

### Base de Conocimiento

Formada por una "base de datos", que recoge la de definición de las funciones de pertenencia de las entradas y el sistema, y una "base de reglas", que caracteriza y resume la política y objetivos del control de un experto por medio de un conjunto de reglas lingüísticas de control.

## **Motor de Inferencias**

Que inferirá las acciones del sistema empleando alguna representación de la implicación borrosa, así como de los procedimientos de inferencia en Lógica Borrosa.

**Interface de Desborrosificación** (o de concreción) que convertirá la acción "borrosa" actualmente inferida en una acción concreta susceptible de aplicación sobre el proceso, y realizara un escalado para adecuar los rangos de salida para los que se ha definido el sistema con las entradas del proceso.

### **2.5.3. Interface de Borrosificación o Borrosificador**

En mucho casos, especialmente dentro de la literatura de control, las entradas a un sistema borroso no son conjuntos borrosos sino valores numéricos concretos. Por eso es necesario establecer algún tipo de interface entre estos valores numéricos y el motor de inferencia borrosa. Por un lado habría que elaborar conjuntos borrosos a partir de las entradas no borrosas (borrosificación) y por otro se habrá que calcular un valor numérico de salida a partir del conjunto borroso obtenido en el proceso de inferencia (desborrosificación).

La primera parte del problema tiene una solución muy sencilla. Si  $x_0$  es el valor concreto de una entrada del sistema borroso, parece lógico definir el correspondiente conjunto borroso de entrada mediante la siguiente función de pertenencia:

$$\mu_A(x) = \begin{cases} 1 & \text{si } x = x_0 \\ 0 & \text{en otro caso} \end{cases}$$

#### **2.5.4. Base de conocimiento**

Formada por:

##### **Base de reglas**

Consideremos como ejemplo ilustrativo un problema genérico que puede ser resuelto en términos de razonamiento aproximado por un ser humano experto en la materia, generalmente siguiendo el esquema de *modus ponens generalizado*. Esto quiere decir que el experto es capaz, mediante un conjunto de reglas o pautas de actuación que constituyen su experiencia sobre el tema de elaborar unas conclusiones o consecuentes a partir de unos hechos observados o antecedentes. El experto podrá transmitir sus conocimientos, al menos parcialmente, mediante un conjunto de N reglas del tipo si-entonces, con antecedentes relacionados por el conectivo "Y" en la mayoría de los casos.

La experiencia del ser humano, recogida y almacenada de esta manera, es lo que podemos denominar base de reglas (o base de conocimiento) por similitud con los métodos de trabajo y la terminología propios de los Sistemas Expertos.

Existen varios modos de derivación de las reglas, siendo este el punto fuerte de numerosas investigaciones actualmente. Cabe destacar:

- Basados en los conocimientos de un experto, generalmente a través de una verbalización introspectivo de esta, o a partir de cuestionarios cuidadosamente organizados.
- Basados en las acciones de control de un operador, en función de los datos de entrada-salida observados.
- Basados en un modelo borroso del proceso.
- Basados en aprendizaje (controladores auto-organizados).

##### **Funciones de pertenencia. Base de datos**

Los conceptos asociados con la Base de Datos se usan para caracterizar las reglas borrosas y la manipulación de los datos en un sistema borroso. Estos

conceptos son definidos subjetivamente y están basados en la experiencia y juicio de un experto sobre el proceso. Algunos de estos aspectos serían la cuantificación y normalización de los universos de discurso, número de conjuntos borrosos o categorías lingüísticas de entradas y control, y la elección de las funciones de pertenencia asociadas a estas últimas.

La elección de niveles de cuantificación tiene una influencia esencial en la precisión y "finura" del sistema obtenido. El número de categorías lingüísticas determina la granularidad del sistema final.

Existen dos métodos para definir conjuntos borrosos (mediante sus funciones de pertenencia), dependiendo de si los universos de discurso son discretos o continuos:

- Definición numérica: En el caso de universos discretos, la función de pertenencia de un conjunto borroso se representa como un vector de números cuya dimensión depende del grado de discretización.
- Definición funcional: En el caso de universos continuos, las funciones de pertenencia se expresan por medio de funciones regulares, típicamente triangulares o trapezoidales.

La elección de la forma de las funciones de pertenencia se basa en criterios subjetivos o característicos de cada proceso en cuestión. Por ejemplo, si los datos medibles van a estar afectados por ruido, las funciones de pertenencia deberían ser suficientemente anchas para reducir la sensibilidad del sistema frente a estos.

#### **2.5.5. Motor de inferencias**

El motor de inferencias representa el núcleo del SBRB, y agrupa toda la lógica de inferencia borrosa del sistema, de barrido de las reglas durante esta, elección refinada de reglas a utilizar, etc.

La inferencia borrosa es el proceso mediante el cual se obtiene como consecuente un conjunto borroso a partir de unos antecedentes también borrosos.

### 2.5.6. Interface de Desemborronado o Desborrosificación.

Habrá que calcular un valor numérico de salida a partir del conjunto borroso obtenido en el proceso de inferencia (desemborronado).

Para calcular la señal de salida que se debe aplicar al proceso se puede recurrir a distintos métodos. Desafortunadamente, no existe ningún procedimiento sistemático para la elección de la estrategia de desemborronado, siendo las mas comúnmente usadas en la actualidad las siguientes:

1. **Método del centroide o centro de gravedad.** Es sin duda el más usado de todos los métodos, ya que proporciona variaciones suaves y continuas de los valores de salida. Esta se calcula como el centro de gravedad de la función de pertenencia del conjunto de salida  $D(y)$ . Esto es:

$$y_0 = \frac{\int y u_D(y) dy}{\int u_D(y) dy} \quad (32)$$

Para el caso de trabajar con valores discretos la expresión queda:

$$y = \frac{\sum_{K=1}^N y_k \mu(y_k)}{\sum_{K=1}^N \mu(y_k)} \quad (33)$$

2. **Método de máxima pertenencia:** el valor de salida es aquel cuyo grado de pertenencia al conjunto de salida  $D(y)$  inferido sea máximo. Es decir:

$$\max_{y \in D(y)} u_D(y) = u_D(y_0) \quad (34)$$

- 3. Método del máximo indexado:** Este método calcula el centro de gravedad del subconjunto borroso del consecuente inferido, correspondiente a los puntos cuyo valor de pertenencia al consecuente es superior a un umbral dado.

### 3. METODOLOGÍA Y PLAN DE TRABAJO

#### 3.1. METODOLOGÍA

Se realizó un estudio preeliminar de la literatura para analizar los modelos de ciclo de vida más utilizados en el desarrollo de proyectos software, finalmente se eligió el modelo mas pertinente con los requerimientos del proyecto.

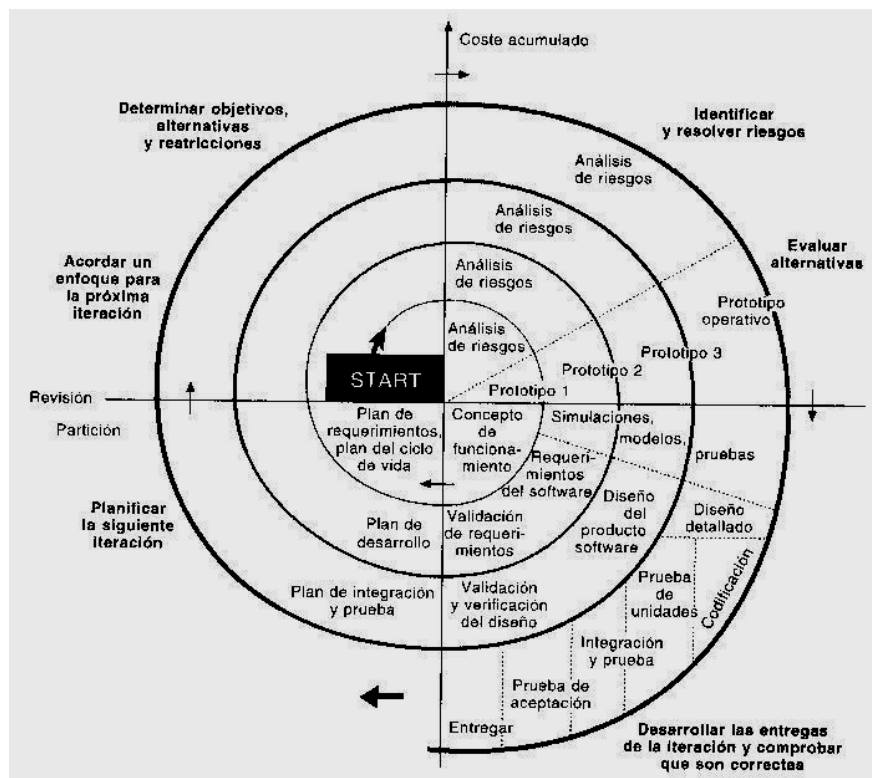
Los modelos evolutivos son modelos iterativos que permiten desarrollar versiones de software cada vez más completas y funcionales. Dentro de estas metodologías podemos encontrar el modelo incremental que combina el modelo en cascada pura y el de construcción de prototipos a través de la entrega de incrementos a través del tiempo de desarrollo del proyecto. Otro modelo que se encuentra dentro de esta categoría es el Modelo en Espiral que combina la construcción de prototipos con el modelo orientado a riesgos que proporciona el modelo lineal secuencial.

Para llevar a cabo este proyecto se selecciono como modelo de desarrollo, ***el modelo en espiral***, para lo que se tuvieron en cuenta inicialmente las siguientes características del proyecto:

- ✓ Este proyecto se divide en miniproyectos software.
- ✓ Las especificaciones deben ser flexibles a las modificaciones o adaptaciones que se lleven a cabo en el desarrollo del proyecto.
- ✓ Se localizan los requisitos, se genera un plan para manejar los requisitos y a continuación se establece una aproximación a la siguiente iteración.
- ✓ Cada iteración supone que el proyecto pasa a una escala superior.

El modelo de espiral es un modelo de ciclo de vida orientado a riesgos que divide un proyecto software en miniproyectos. Cada miniproyecto se centra en uno o más riesgos importantes hasta que todos éstos estén controlados. El concepto “riesgo” puede referirse a requerimientos poco comprensibles, arquitecturas poco comprensibles, problemas de ejecución importantes, y demás.

En la **Figura 17**, se detalla un esquema de flujo de trabajo, desde su análisis y especificaciones iniciales, hasta la obtención del producto deseado, incluyendo las fases que se deben ejecutar.



**Figura 17.** Esquema de flujo de trabajo

Se puede adaptar cada iteración de la espiral a las necesidades que demanda el proyecto. El modelo se puede combinar con otros modelos de ciclo de vida de dos maneras. Se puede comenzar un proyecto con una serie de iteraciones para reducir los riesgos a un nivel aceptable; después, se puede finalizar el

esfuerzo de desarrollo con un ciclo de vida en cascada u otro modelo no basado en riesgos. También se puede incorporar otros modelos de ciclo de vida como iteraciones dentro del modelo en espiral.

Algunas de las ventajas que ofrece el modelo en espiral son:

- Trabaja con poca identificación de los requerimientos;
- Trabaja con poca comprensión sobre la arquitectura;
- Genera un sistema altamente fiable;
- Genera un sistema con amplio desarrollo;
- Gestiona riesgos durante la planificación, para el producto y otro tipo de riesgos;
- Permite modificaciones a medio camino;
- Ofrece signos visibles de progreso.

### **3.2. PROGRAMACIÓN ORIENTADA A OBJETOS**

Un programa tradicional se compone de procedimientos y de datos. Un programa orientado a objetos consiste solamente en objetos, entendiendo por objeto una entidad que tiene unos atributos particulares, los datos, y unas formas de operar sobre ellos, los métodos o procedimientos.

La programación orientada a objetos es una de las técnicas más modernas que trata de disminuir el costo del software, aumentando la eficiencia en la programación y reduciendo el tiempo necesario para el desarrollo de una aplicación. Con la programación orientada a objetos, los programas tienen menos líneas de código, menos sentencias de bifurcación y módulos que son más comprensibles porque reflejan de una forma clara la relación existente entre cada concepto a desarrollar y cada objeto que interviene en dicho desarrollo. Donde la programación orientada a objetos toma verdadera ventaja es en la manera de compartir y reutilizar el código.

No obstante, no debe pensarse que la programación orientada a objetos resuelve todos los problemas de una forma sencilla y rápida. Para conseguir buenos resultados, es preciso dedicar un tiempo significativamente superior al análisis y al diseño. Pero, éste no es un tiempo perdido, ya que simplificará enormemente la realización de aplicaciones futuras.

Según lo expuesto, las ventajas de la programación orientada a objetos son sustanciales. No obstante, también presenta inconvenientes; por ejemplo, la ejecución de un programa no gana en velocidad y obliga al usuario a aprenderse una amplia biblioteca de clases antes de empezar a manipular un lenguaje orientado a objetos.

Sin embargo, la programación orientada a objetos se ha convertido en un paradigma importante en todos los campos de la Ciencia de la Computación y, por ello, es importante utilizar esta técnica para el desarrollo de este proyecto; la principal razón que conduce a utilizarla, es porque permite la reutilización del código del lenguaje y, a su vez, facilita la actualización en versiones mejoradas. Adicionalmente tiene otras ventajas que son importantes para el proyecto, como la flexibilidad que permite en la programación, en cuanto si el código de una clase no funciona, esto no obstruye la funcionalidad de todo el prototipo del software.

### **3.3. LENGUAJE UNIFICADO DE MODELADO UML**

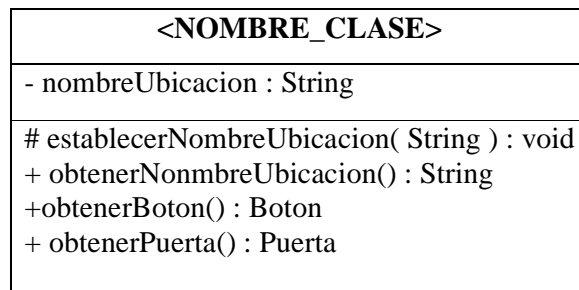
Se ha escogido UML, porque sus diagramas facilitan el entendimiento de la funcionalidad del diseño. Además, debido a que existe un interés, en que la primera versión sea actualizada con facilidad en otras versiones más robustas. Ya que esto traería mayor acceso a la investigación, en lo que se refiere a los temas de la lógica difusa y la contaminación del aire.

Con la utilización de UML, un lenguaje de modelado estándar, se incentivará el modelaje de sistemas de software antes y durante la construcción. Se

utilizarán las partes claves de la notación, la semántica y el proceso dirigido a personas que ya conocen la tecnología orientada a objetos, para utilizar estos elementos.

UML es un lenguaje gráfico creado para ofrecer una solución a la hora de construir o planificar un sistema software o cualquier otro modelo de negocio.

Ejemplo para un elevador:



**Figura 18.** Diagrama de UML.

NOTACION para diagramas:

- si privado

+ si público

# si protegido

La relación también puede ser de herencia:

↖ de cual extends (herencia).

Es un lenguaje de modelado y no un método. Los métodos consisten en su mayoría en un lenguaje y un proceso para modelar. El lenguaje de modelado es la notación principalmente gráfica, de que se valen los métodos para expresar los diseños. El proceso es la orientación que nos dan sobre los pasos a seguir para hacer el diseño. El lenguaje de modelado es la parte más importante del método, es la clave para la comunicación.

Lo que UML permite es crear un modelo a partir de ciertos elementos y ciertas técnicas, que representan el sistema. Los modelos creados hacen uso de notación gráfica que representa principalmente la información, los procesos y el comportamiento del sistema.

También permite una abstracción del sistema y sus componentes; UML se usa para especificar, visualizar, construir y documentar los componentes de un sistema software, también para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

### **3.4. PLAN DE TABAJO**

Teniendo en cuenta la metodología seleccionada para el desarrollo del proyecto, y realizando una adaptación para que el modelo se ajuste mejor a las necesidades, el plan de trabajo contará con las siguientes etapas principales:

#### **3.4.1. FASE I: Concepto Inicial**

En esta fase se realizan las siguientes etapas:

**3.4.1.1. Recopilación de Información:** Esta etapa tiene en cuenta las siguientes actividades:

- *Búsqueda Bibliográfica:* Recolección del material necesario para la ambientación y desarrollo del proyecto;
- *Análisis Bibliográfico:* Clasificar y examinar el material recolectado para enmarcar el contexto del proyecto.

**3.4.1.2. Entrenamiento Previo:** En esta etapa se lleva a cabo la familiarización con los ambientes de programación a utilizar en el desarrollo de la herramienta software.

**3.4.1.3. Especificaciones:** En esta etapa se definen los recursos necesarios para la elaboración de este proyecto, además se realiza una recopilación de los requerimientos generales del sistema.

### **3.4.2. FASE II: Construcción**

Haciendo uso de la metodología de desarrollo seleccionada, se desea llevar a cabo el proyecto en dos iteraciones. En la primera iteración se desarrollará un prototipo que cumpla con un conjunto de requerimientos generales planteados en la primera fase del proyecto. Posteriormente se realizará un refinamiento del prototipo basado en la realimentación, con esto se actualizan los requerimientos y se cumple con el objeto de desarrollo del proyecto. En esta fase se realizarán las siguientes etapas:

#### **3.4.2.1. Revisión de Requerimientos y Funcionalidades del Producto:**

En esta etapa se determina el grado en que el componente a producir cumplirá con los requisitos que fueron contemplados en la primera fase. Así mismo, se recopilan y analizan los requisitos específicos que surjan durante esta fase.

#### **3.4.2.2. Diseño:**

En esta etapa se realiza un análisis detallado del prototipo que se va a producir en el ciclo actual, haciendo uso de UML tal y como se definió en la metodología a utilizar para la construcción del proyecto.

#### **3.4.2.3. Construcción:**

En esta etapa se realizó la implementación del prototipo definido para el ciclo actual y, se realizaron algunas modificaciones al diseño de alto nivel.

#### **3.4.2.4. Pruebas y Evaluación:**

Una vez construido el prototipo, se deben realizar las respectivas pruebas comprobando su funcionamiento y además que se cumplan los requisitos especificados en esta fase. Además, se realiza una evaluación de los

resultados obtenidos, produciendo información que servirá de realimentación para refinar los requerimientos y especificaciones.

#### **3.4.3. FASE III: Prueba del Sistema:**

En esta fase se realizan pruebas completas de funcionamiento del sistema con diferentes usuarios.

#### **3.4.4. FASE IV: Documentación:**

Esta actividad se realiza en paralelo a las otras.

## **4. FASE DE CONSTRUCCIÓN**

Los prototipos modelados en esta fase fueron dos. El primero para verificar la funcionalidad de lo desarrollado, calcular los datos de salida y buscar errores y el segundo para corregir los errores, se hace la integración con el módulo “Interface Gráfica de Usuario”, establecer la veracidad de los datos de salida y alcanzar una mayor facilidad de manejo para los usuarios.

### **4.1. PRIMER PROTOTIPO.**

Este primer prototipo implementa todas las funcionalidades básicas del sistema de Lógica Difusa, la parte gráfica del sistema y una primera aplicación desde la cual se pueden llamar estas gráficas para capturar sus parámetros y ser visualizados.

En esta fase no fue necesario arrancar de cero y fue de gran ayuda que la directora de este proyecto, que al mismo tiempo es la investigadora principal del trabajo de grado de maestría del cual hace parte este proyecto; aportó el diseño de alto nivel; además de los requerimientos descritos, a través de los casos de uso.

En esta fase se definen los requerimientos del sistema:

#### **4.1.1. Requisitos:**

En general visualizar un Frame implementado como una instancia de la clase JFrame, es el contenedor de los componentes, esto es, crear una ventana, que nos será muy útil ya que una vez implementada podremos ir agregándole elementos gráficos como tabs, bordes, textos, botones, etc; que contenga los

métodos y constructores para crear un sistema de inferencia basado en Lógica difusa que permita tomar los valores de entrada, aplicar el procedimiento de emborronado, desemborronado e inferir el conjunto de valores de salida.

En los subpaquetes para (parámetros) y cvar (variables) para el sistema debe poder crear, modificar, borrar o graficar la variable lingüística por lo tanto debe realizar las siguientes tareas:

Variable Lingüística:

- Nombre
- Clase de variable: de entrada o de salida
- Unidades.
- Rango: valor mínimo, valor máximo

Que puede contener uno o varios conjuntos borrosos.

Conjunto borroso:

- Nombre
- Función: Gaussiana, Bell, Sigmoidal, Trapezoidal, Triangular; estas serian las funciones de membresía.
- Color de la función de membresía.
- Parámetros: varían según la función; pueden ser a,b,c,d o sigma.

Modificar variable lingüística: esta tarea cambia cualquiera o cuales quiera de las características de la variable lingüística y los conjuntos borrosos que esta contiene.

Eliminar variable lingüística: esta tarea permite eliminar cualquier variable lingüística seleccionada y los conjuntos borrosos que esta contiene.

Modificar conjunto borroso: esta tarea cambia cualquiera o cualesquiera de los parámetros y características del o de los conjunto borrosos seleccionados.

Eliminar conjunto borroso: esta tarea permite eliminar cualquier conjunto borroso de la variable lingüística seleccionada.

Estas tareas deben ir vinculadas a la parte de reglas.

En el subpaquete rules (reglas) para el sistema debe poder crear, modificar, borrar y tener la opción seleccionar el conector “AND” u “OR”, por lo tanto debe realizar las siguientes tareas:

- Mostrar Variables Lingüísticas de entrada y de salida con sus respectivas etiquetas lingüísticas.
- Mostrar conector: AND, OR.
- Escoger conector: AND, OR.
- Escoger etiqueta lingüística de cada variable Lingüística de entrada
- Escoger etiqueta lingüística de cada variable Lingüística de salida
- Crear regla
- Modificar regla
- Borrar regla

Resultados:

- Mostrar gráficamente las Variables Lingüísticas de entrada
- Seleccionar el valor de activación para cada variable Lingüística de entrada
- Realizar el proceso de desemborronado para determinar el valor o los valores de salida.
- Mostrar gráficamente el correspondiente conjunto borroso, de cada variable lingüísticas entrada, activado por la regla.
- Mostrar gráficamente las Variables Lingüísticas de salida con su correspondiente conjunto borroso de salida activado por la regla.
- Mostrar gráficamente el resultado de la activación de la regla.
- Mostrar numéricamente el resultado.
- Mostrar gráficamente el resultado.

## **Métodos**

Para poder desarrollar y trabajar con los requisitos expuesto anteriormente el módulo debe realizar unas tareas implícitas:

### **Crear y agregar variables lingüísticas:**

Debe tener la opción de crear variables lingüísticas con sus respectivas características, lo cual implica:

1. La creación de una o varias variables lingüísticas.
2. Nombre de la función de membresía.
3. Unidades
4. Rango:
  - Valor máximo
  - Valor mínimo
5. Tipo: Entrada o salida.

### **Modificar las variables lingüísticas:**

1. Escoger la variable lingüística a modificar.
2. modificar sus características: (modificarlas en reglas y resultados)
  - Nombre
  - Unidades
  - Rango: Valor máximo
  - Valor mínimo
  - Tipo: Entrada o Salida
3. Actualizar en la parte numérica y gráfica.

### **Eliminar variables lingüísticas:**

1. Escoger la variable lingüística a eliminar.

2. Borrar la variable lingüística seleccionada.

### **Crear y agregar conjuntos borrosos:**

Debe tener la opción de crear conjuntos borrosos con sus respectivas características, lo cual implica:

1. La creación de uno o varios conjuntos borrosos.
2. Nombre de la función de membresía.
3. Función de membresía:
  - Gaussiana
  - Bell
  - Sigmoidal
  - Trapezoidal
  - Triangular.
4. Color de la función de membresía.
5. Parámetros: a, b, c, d ó  $\sigma$ . Depende de la función de membresía.

### **Modificar los conjuntos borrosos:**

1. Escoger el conjunto borroso a modificar.
2. Modificar sus características: (modificarla en reglas y resultados)
  - Nombre de la función de membresía.
  - Función de membresía:
    - Gaussiana
    - Bell
    - Sigmoidal
    - Trapezoidal
    - Triangular.
3. Color de la función de membresía.
4. Parámetros: a,b,c,d o sigma. Depende de la función de membresía.
5. Actualizar en la parte numérica y gráfica.

**Eliminar conjuntos borrosos:**

1. Escoger el conjunto borroso a borrar.
2. Borrar el conjunto borroso seleccionado.

**Crear base de reglas:**

1. Mostrar Variables Lingüísticas de entrada y sus conjuntos borrosos.
2. Mostrar Variables Lingüísticas de salida y sus conjuntos borrosos.
3. Mostrar conector: AND, OR.
4. Escoger conector: AND, OR.
5. Escoger variables lingüísticas de salida y su conjunto borroso.
6. Escoger variables lingüísticas de entrada y su conjunto borroso.
7. Crear y agregar regla a la base de reglas: adhiere una nueva regla a la base de reglas.

**Modificar la base de reglas:**

1. Escoger la regla a modificar.
2. Modificar su contenido: (modificarla en reglas y resultados)  
Conector: AND, OR.
3. Escoger variables lingüísticas de salida y su conjunto borroso.
4. Escoger variables lingüísticas de entrada y su conjunto borroso.
5. Buscar y actualizar regla en la base de reglas

**Eliminar reglas de la base de reglas:**

1. Escoger la regla eliminar.
2. Borrar la regla.

## Resultados del proceso de Emborronado

1. Mostrar gráficamente las Variables Lingüísticas de entrada
2. Leer el valor de activación: Leer el valor crisp, suministrado por el usuario en la “Interface Gráfica de Usuario”, el cual es un valor que activa las etiquetas lingüísticas de cada variable lingüística.
3. Para cada variable lingüística se calcula el grado de pertenencia de cada conjunto borroso activado.
4. Identificar las reglas, que son activadas por los valores de entradas.
5. Identificar los conjuntos borrosos que son afectados por las reglas activadas y de cuales variables lingüísticas de salida proviene.
6. Realizar el proceso de Desemborronado, para determinar el valor o los valores de salida.

Analizando estos requisitos y por recomendación del codirector de este proyecto se determina la utilización de Java para desarrollar este Prototipo por lo tanto se utilizará al máquina virtual de Java: **J2SE 1.4.2\_13** el cual su aplicación final puede ser utilizada en diferentes plataformas:

- ✓ Windows 32 bits Platform
- ✓ Windows x64 bits Platform
- ✓ Linux Platform
- ✓ Linux x64 Platform
- ✓ Mac OS Platform
- ✓ Solaris SPARC Platform
- ✓ Solaris x86 Platform

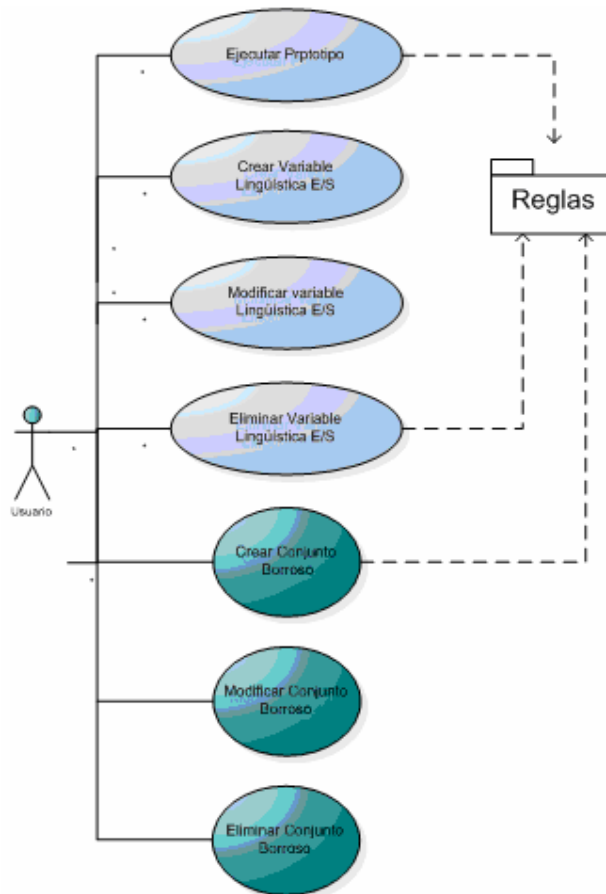
Este se puede obtener de forma gratuita en la página de SUM: <http://www.sun.com/> o de la página de NetBeans: <http://www.netbeans.org/>

Y el IDE utilizado será **NetBeans 5.0 IDE** también puede ser bajado de la pagina de SUM.

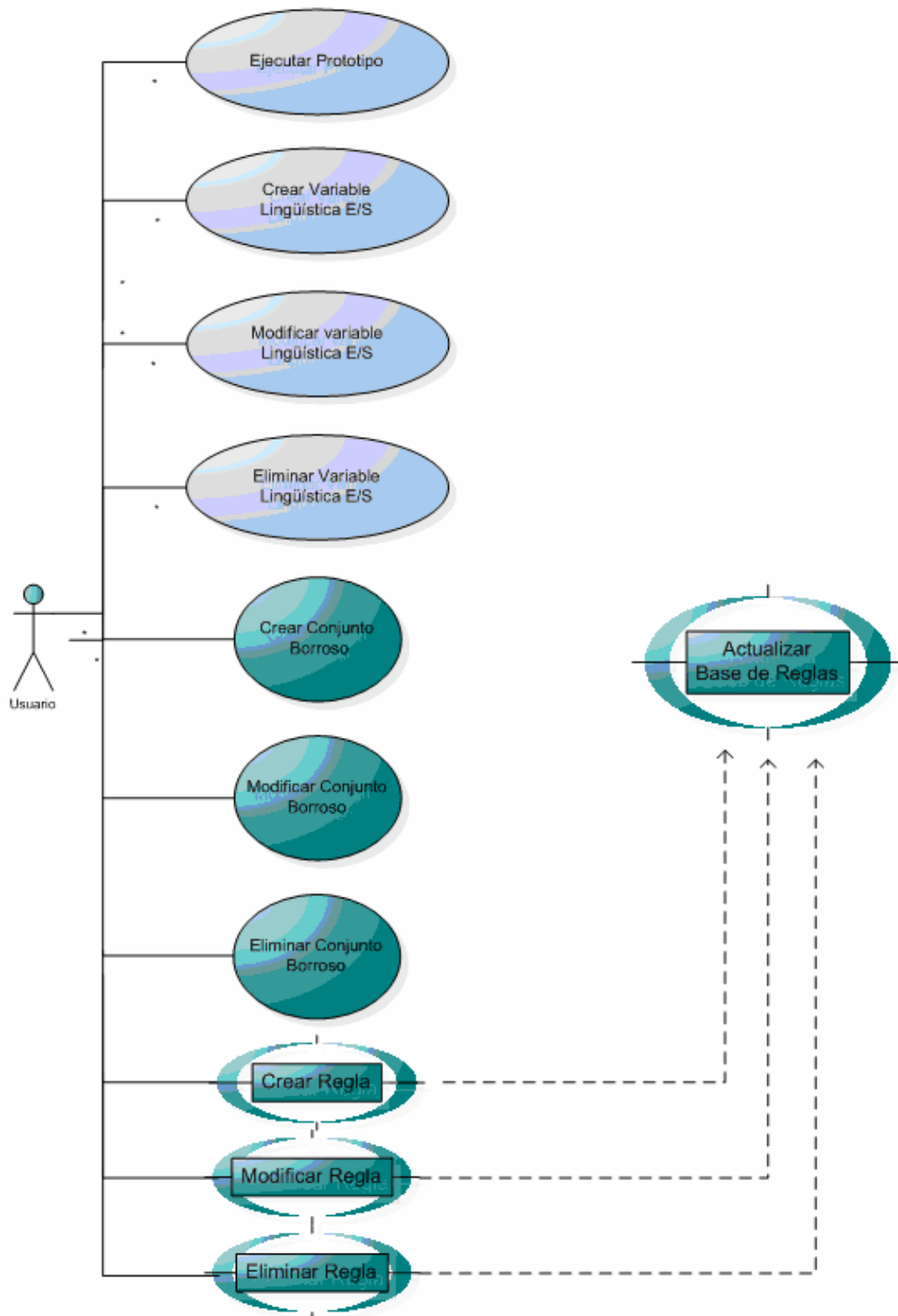
Utilizando Java, se pueden eliminar los inconvenientes de la interface CGI y también se pueden añadir aplicaciones que vayan desde experimentos científicos interactivos de propósito educativo a juegos o aplicaciones especializadas.

### Casos de uso.

En el siguiente diagrama de casos de uso se representa la forma en como un usuario (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso). En la figura 19 se muestran los casos de usos para el primer prototipo.



**Figura 19.** Casos de uso cvar y oper



**Figura 20.** Casos de uso para el primer prototipo.

El actor representa cualquier usuario que utilice el módulo, en este caso serán miembros de el GIEMA o el CEIAM.

- Capturar entrada: El módulo recibe las entradas o datos suministrados por el cliente.
- Crear variable lingüística de E/S: el módulo recibe y captura los datos de entrada suministrados por el usuario y crea las variables lingüísticas de entrada o salida, según especifique el usuario, con sus respectivas características.
- Modificar variable lingüísticas de E/S: le permite al usuario cambiar una o varias características de las variables lingüísticas de entrada o de salida.
- Eliminar variables lingüísticas de E/S: le permite al usuario eliminar la variable lingüística de entrada o de salida escogida.
- Crear conjunto borroso: le permite al usuario crear un conjunto borrosos y secuencialmente crear varios en la variable lingüística de entrada o de salida escogida.
- Modificar los conjuntos borrosos: le permite al usuario modificar las características individualmente de un conjunto borroso en la variable lingüística de entrada o de salida escogida.
- Eliminar conjunto borroso: le permite al usuario eliminar un conjunto de la variable lingüística de entrada o de salida seleccionado.
- Crear regla: le permite al usuario armar la regla de inferencia definiéndola de cada variable lingüística ya sea de entrada o de salida determinando sus características y agregándola a la base de reglas.
- Modificar regla: le permite cambiar las características de cada regla y actualizar la base de reglas esto lo hace automáticamente.
- Eliminar regla: le permite al usuario eliminar una regla de la base de reglas.
- Resultado: le permite al usuario ingresar los valores crisp para computar las reglas que generan los datos de salida y mostrar estos en forma numérica y gráfica.

#### **4.1.2. Diseño**

Como una fase inicial de la construcción se realizó una revisión exhaustiva del diseño de alto nivel suministrado por el trabajo de grado de Maestría del cual hace parte este proyecto.

Luego, mientras se llevó a cabo el desarrollo, se realizaron algunos cambios de ese diseño inicial. Con el objeto de facilitar el mismo desarrollo y más que todo la integración.

Y como resultado se obtuvo el siguiente diseño (Figura 21) .

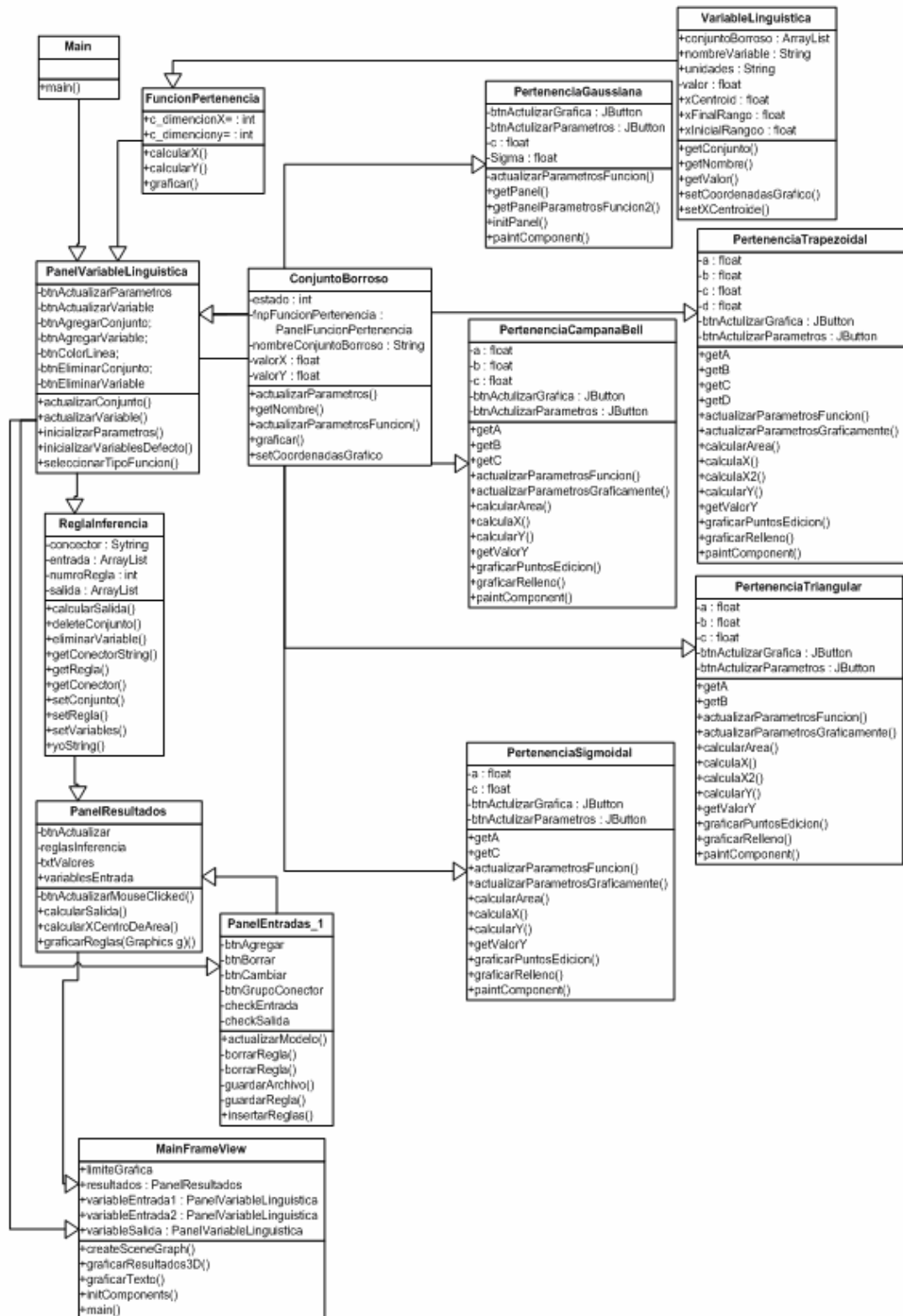


Figura 21. Diagramas de UML para el primer prototipo.

### **4.1.3. Implementación.**

Luego de haber realizado el desarrollo en NetBeans el módulo cumple con los requerimientos especificados encontrados en la etapa de análisis en este primer prototipo, los cuales llamamos métodos:

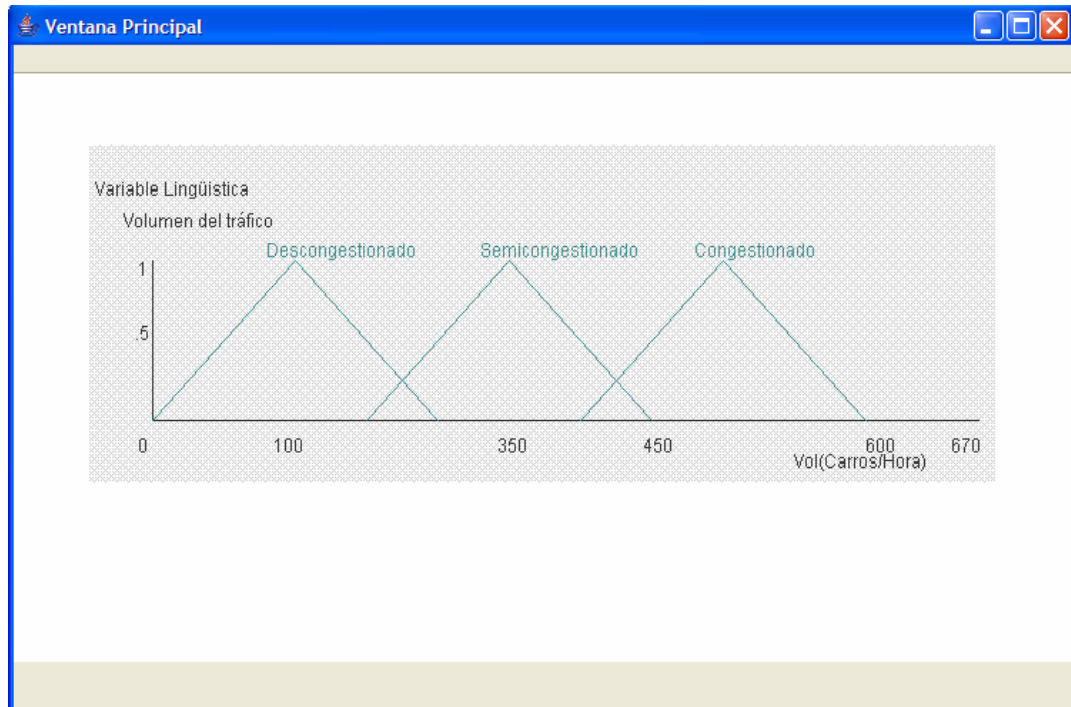
Estos métodos son:

- ✓ Crear variable lingüística de E/S
- ✓ Modificar variable lingüísticas de E/S
- ✓ Eliminar variables lingüísticas de E/S
- ✓ Crear conjunto borroso
- ✓ Modificar conjunto borroso
- ✓ Eliminar conjunto borroso
- ✓ Crear base de Reglas
- ✓ Crear regla
- ✓ Modificar regla
- ✓ Eliminar regla
- ✓ Mostrar resultados.

### **4.1.4. Evaluación:**

Siendo este el primer prototipo las pruebas van encaminadas a verificar las funciones básicas, calcular datos de salida, veracidad y registrar posibles errores y modificaciones necesarias.

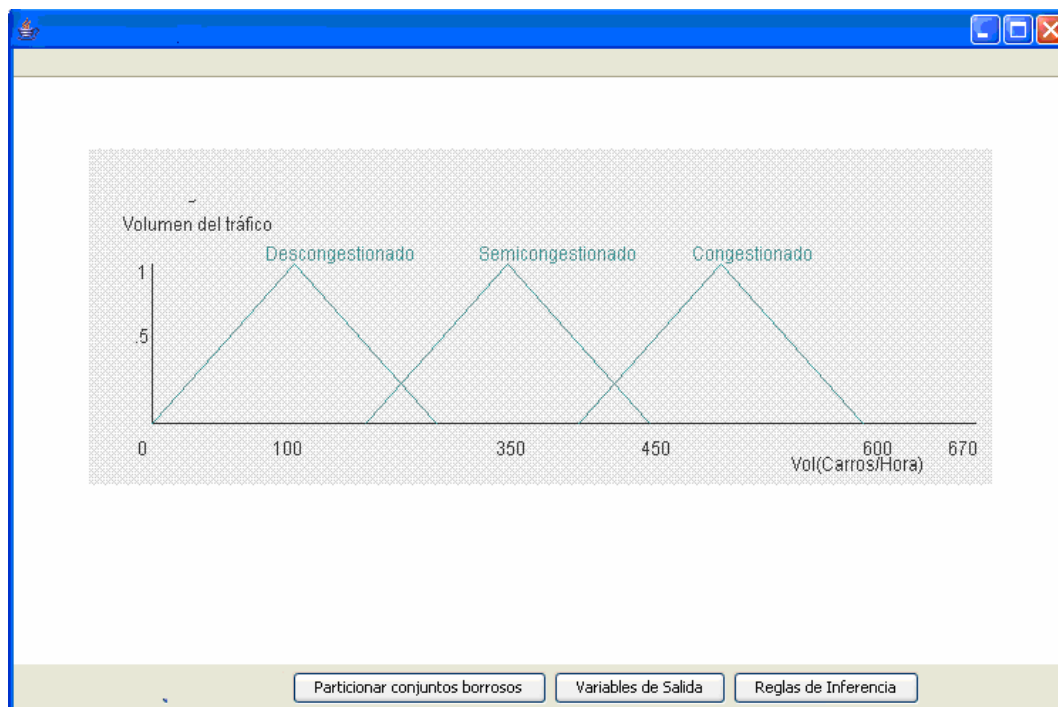
- ✓ Crea las variables lingüísticas con las características parámetros introducidos por el usuario.
- ✓ Muestra gráficamente estos datos: dibuja un espacio en un plano coordinado que contiene las variables de entrada y de salida y sus respectivos conjuntos borrosos.



**Figura 22.** Variable Lingüística en forma gráfica.

- ✓ Crea los conjuntos borrosos
- ✓ Modifica los conjuntos borrosos
- ✓ Cree la base de reglas
- ✓ Crea reglas
- ✓ Modifica reglas
- ✓ Elimina reglas
- ✓ Permite introducir los valores de entrada: crips
- ✓ Y presenta el resultado.

En el cual implementamos y sujetamos los módulos que contienen los métodos para los anteriores procesos, (**figura 23**).



**Figura 23.** Variable Lingüística e inserción de métodos.

#### 4.2. SEGUNDO PROTOTIPO.

Como establecimos anteriormente en el segundo prototipo se corrigen los errores, se hace la integración con el módulo Interface de Usuario, se establece la veracidad de los datos de salida y se aumenta la facilidad de manejo para los usuarios además de esto se introduce una nueva forma gráfica de mostrar los resultados, la cual consiste en gráficos 3D lo que le da un valor agregado.

Para la Integración con la interface gráfica, debido a la programación de los autores se hicieron cambios y reajustes en el código fuente: se introdujo el código desarrollado en este módulo (variables y parámetros) en el código de la "Interface Gráfica de usuario" con el fin de que los datos suministrados por el usuario los tome como datos de entrada, puedan ser guardados, posteriormente abrirlos y trabajar sobre ellos.

Para implementar los resultados en gráficos 3D se utilizó una librería de java llamada **Java 3D 1.5.0**. Java 3D es un proyecto que permite crear entornos tridimensionales en el lenguaje Java, Es una **API** (del inglés **Application Programming Interface - Interface de Programación de Aplicaciones**, es el conjunto de funciones, procedimientos o métodos en este caso que se refiere a programación orientada a objetos que ofrece esta librería para ser utilizado por otro software como una capa de abstracción.) para gráficos 3D para este lenguaje de programación la cual corre sobre OpenGL o Direct3D. Desde la versión 1.2 Java 3D es desarrollado bajo las especificaciones Java Community Process; es distribuida por Sun de forma gratuita y esta disponible en la pagina de SUM: <http://www.sun.com/> o de la página de NetBeans: <http://www.netbeans.org/>. Y puede ser aplicado para multiplataforma:

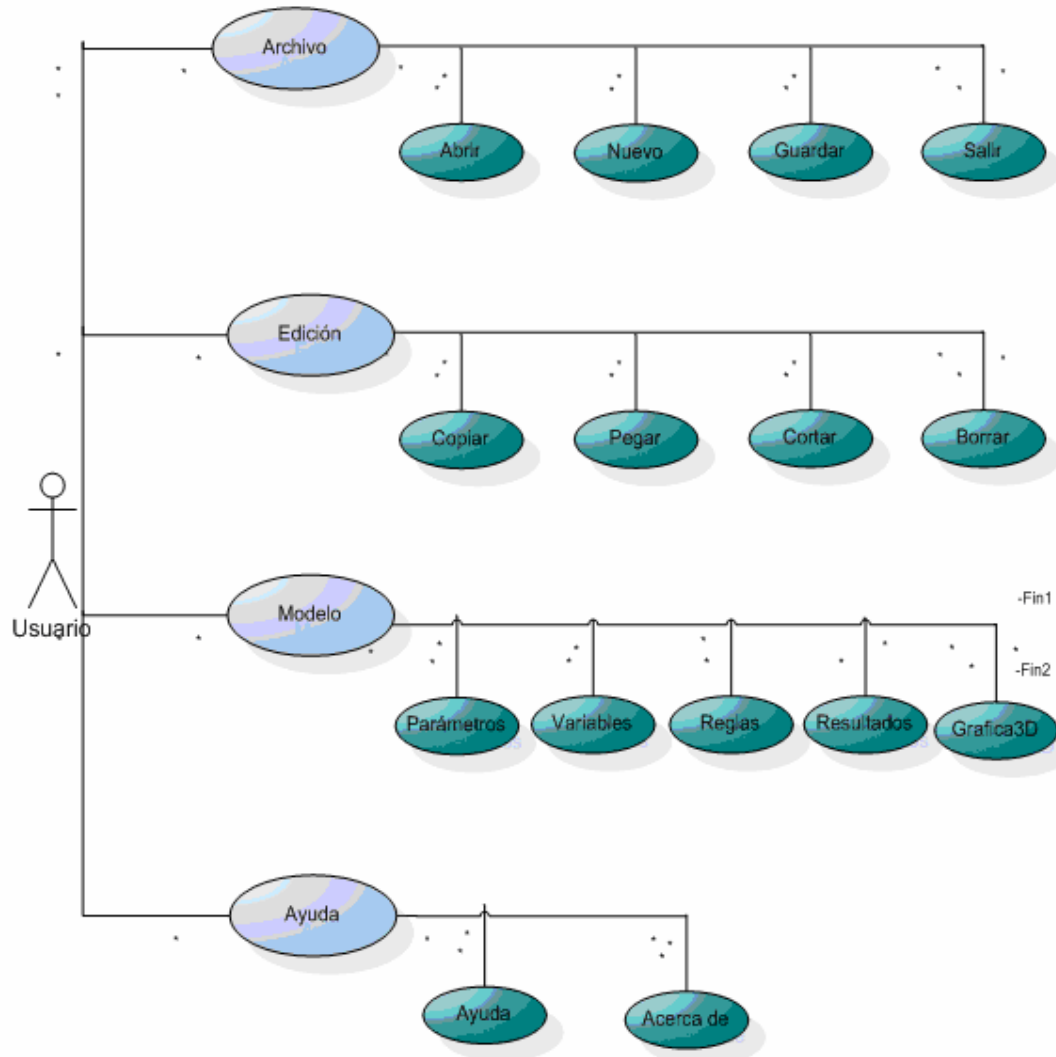
- ✓ Windows Platform 32 bits
- ✓ Windows x64 Platform
- ✓ Linux Platform
- ✓ Linux x64 Platform
- ✓ Mac OS Platform
- ✓ Solaris SPARC Platform
- ✓ Solaris x86 Platform

#### **4.2.1. Requisitos**

Del análisis realizado en el primer prototipo se seguirán tomando sus requerimientos, se conservan todas las funciones o métodos definidos en el primer prototipo, se sumaran los necesarios para la integración con la interface de usuario y se agregaran mejoras agregando requerimientos notados para este segundo prototipo.

Además de los requerimientos mencionados en el primer prototipo para este segundo prototipo se agregan los pertenecientes al módulo Interface Gráfica de

Usuario, que fue desarrollado en otro proyecto, y para una mejor perspectiva del producto final se muestra a continuación el diagrama de casos de uso:



**Figura 24.** Diagrama de casos de uso de la Interface Gráfica de Usuario.

Se agregan varios menús con los cuales el usuario podrá interactuar para crear sus aplicaciones, como: Menú Archivo, Menú Modelo y Menú Ayuda.

El Menú modelo es uno de los más importantes, este nos permite interactuar con los demás módulos que están creados como tabs en el software.

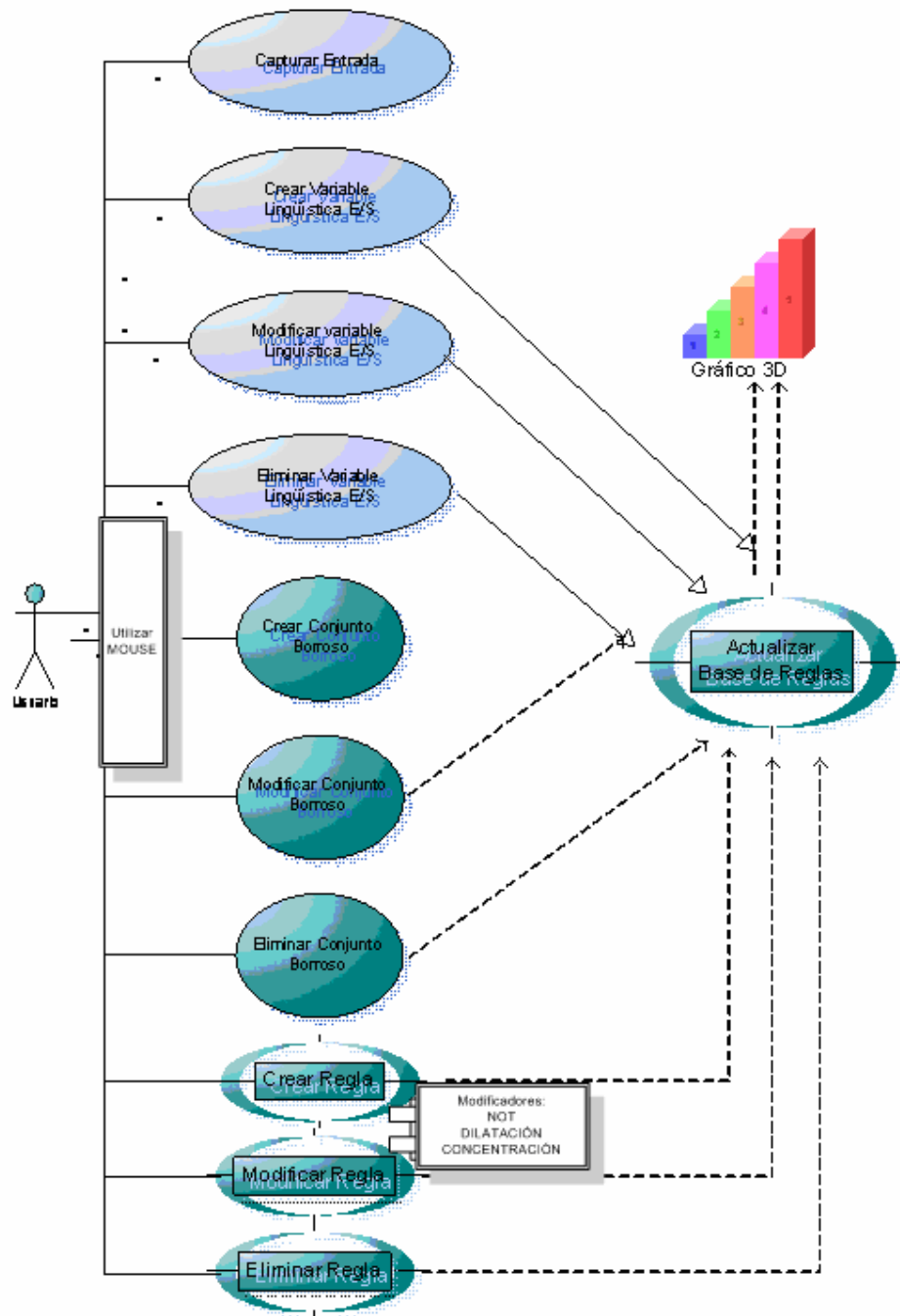
Una **pestaña, lengüeta** o **tab** es una manera de navegar que permite cambiar rápidamente lo que se está viendo sin cambiar de ventana. Desempeñar una tarea a través de pestañas permite cargar varios elementos separados, dentro de una misma ventana y así es posible alternar entre ellos con una mayor comodidad.

Además de estas inclusiones se toman las sugerencias hechas por el director y codirector del proyecto:

- ✓ Crear un nuevo tab que visualice la configuración inicial del modelo.
  - Nombre del modelo
  - Tipo del modelo
  - Método AND
  - Método OR
  - Método de Defusificación.
  - Variables de entrada
  - Variable se salida.
- ✓ Darle más interacción al mouse para utilizar el prototipo.
- ✓ Introducir el mouse para desplazar las gráficas creadas y cambiar sus propiedades.
- ✓ Introducir una gama de colores para los diferentes conjuntos borrosos
- ✓ Introducir los modificadores:
  - NOT
  - DILATACIÓN
  - CONCENTRACIÓNCon sus respectivas modificaciones en tab reglas y tab resultados.
- ✓ Crear un nuevo método para ver los resultados gráficamente en forma tridimensional.
- ✓ Generar un archivo de salida para poder ser guardado y modificado.

Debido a la interacción que hay entre el subpaquete Interface Gráfica del Usuario, el paquete Motor de Inferencia y otros supaquetes como: para

(parámetros), cvar (variables), oper (operadores) y rule (reglas); estos se unificaron dando como resultado una interface para estos paquetes y subpaquetes. En términos de diseño y visualización el módulo para y cvar quedarían unidos; los cuales corresponden a los parámetros y las propiedades de las Variables Lingüísticas. Este subpaquete provee al usuario las funciones administrativas para la o las variables lingüísticas creadas y a usar.



**Figura 25.** Casos de uso para el segundo prototipo.

Entre estas funciones están: crear, nombrar, definir unidades, rango de valores que serán valor mínimo y máximo, eliminar variable, modificar esta opción vinculada al botón actualizar, (figura 26).

The image shows a software interface with two main sections: 'VARIABLE LINGUISTICA' and 'CONJUNTO BORROSO'. The 'VARIABLE LINGUISTICA' section includes radio buttons for 'Entrada' and 'Salida', a 'Nombre' dropdown menu set to 'Item 1', input fields for 'Unidades' (°C), 'Valor mínimo' (0), and 'Valor máximo' (1), and buttons for 'Nueva variable', 'Eliminar', and 'Actualizar'. The 'CONJUNTO BORROSO' section includes a 'Nombre' dropdown menu set to 'Item 1', a 'Función' dropdown menu set to 'Item 1', a 'Color de línea' input field, and buttons for 'Nuevo conjunto', 'Eliminar', and 'Actualizar parámetros'. A 'Parámetros de la función' area is present but empty.

Figura 26. Módulos para y cvar.

The image shows a screenshot of a software application window titled 'Archivo Ayuda'. The window has a menu bar with 'Archivo' and 'Ayuda'. Below the menu bar are tabs for 'Variable Linguistica', 'Reglas', 'Resultados', and 'Grafica 3D'. The main content area is a large empty space. At the bottom of the window, there are two panels: 'VARIABLE LINGUISTICA' and 'CONJUNTO BORROSO'. The 'VARIABLE LINGUISTICA' panel includes radio buttons for 'Entrada' and 'Salida', a 'Nombre' dropdown menu, input fields for 'Unidades' (°C), 'Valor mínimo' (0), and 'Valor máximo' (1), and buttons for 'Nueva variable', 'Eliminar', and 'Actualizar'. The 'CONJUNTO BORROSO' panel includes a 'Nombre' dropdown menu, a 'Función' dropdown menu set to 'Gaussiana', a 'Color de línea' input field with a blue color swatch, and buttons for 'Nuevo conjunto', 'Eliminar', and 'Actualizar parámetros'. A 'Parámetros de la función' area is present but empty.

Figura 27. Módulos para y cvar integrados a la interface principal.

## 4.2.2. Diseño.

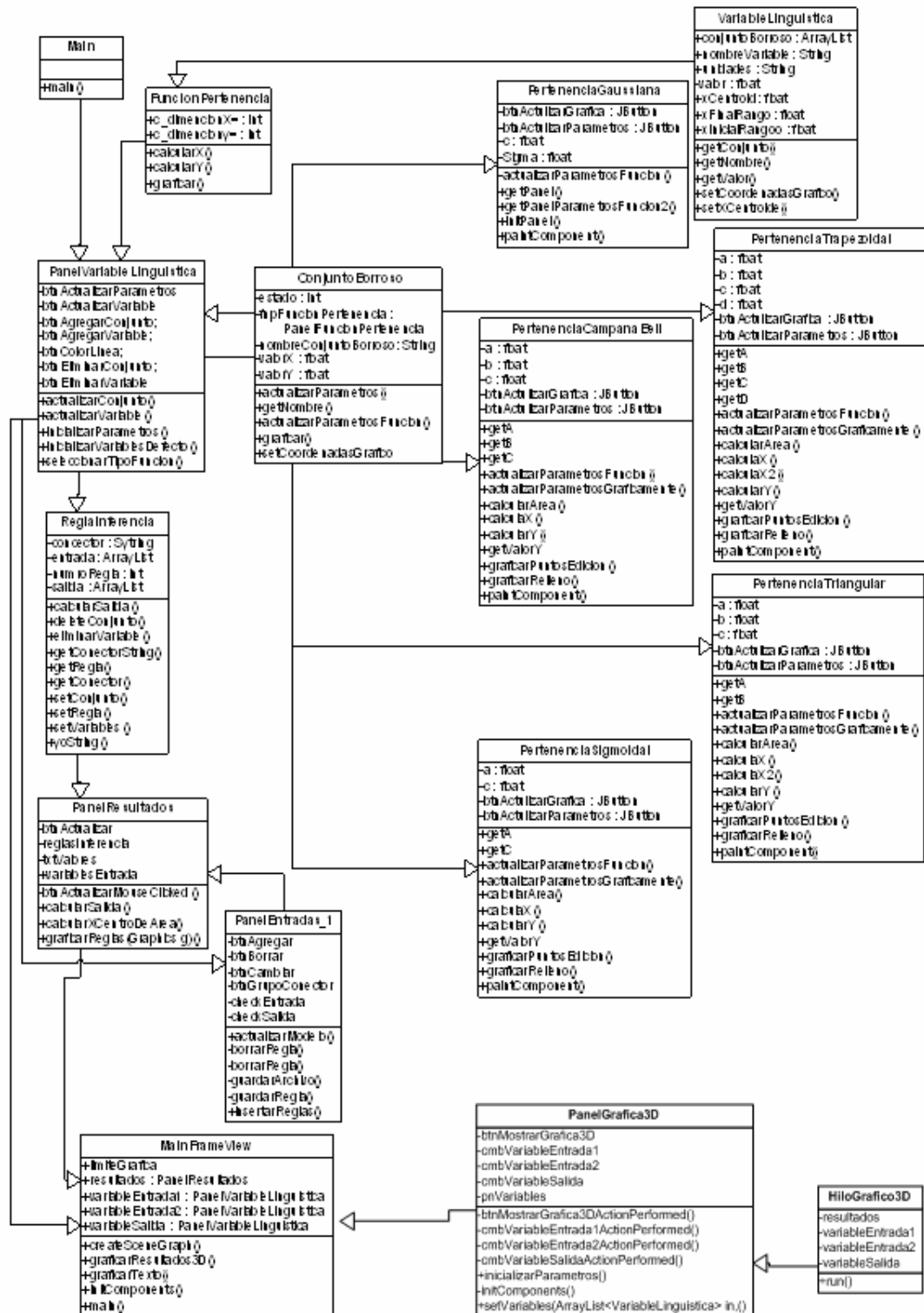
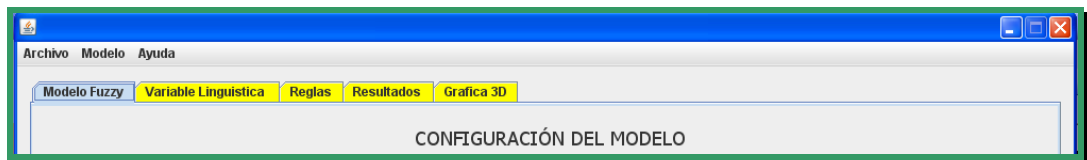


Figura 28. Diagramas de UML para el segundo prototipo.

### 4.2.3. Implementación.

En este segundo prototipo el subpaquete Interface Gráfica de Usuario ha sido integrado al paquete Motor de Inferencia. Por esto, a partir de esta sección se muestran los gráficos correspondientes a esta interfase, además, ya están integrados los métodos y procedimientos para crear, modificar y eliminar variables lingüísticas, conjuntos borrosos y demás tareas que debe realizar el motor de inferencia.

La posibilidad de agrupar los diferentes editores y resultados en TABs que son pestañas o lengüetas implementadas dentro del frame principal para establecer y ordenar en forma secuencial las ventanas dentro de la aplicación, (**figura 29**).



**Figura 29.** Tabs, pestañas o lengüetas.

Tab Modelo Fuzzy

Tab Editor de Variables Lingüísticas y conjuntos borrosos

Tab Editor de Reglas

Tab Resultados: editor de Valores Crips y Visualizador de resultados

Tab Grafico 3D.

Para que el usuario implemente paso a paso sus aplicaciones de una manera además de secuencial lógica y con la opción de devolverse al paso o pasos anteriores en el caso de desear corregir o modificar para corregir un error o generar una mejora.

Luego de haber realizado los cambios el módulo cumple con los requerimientos especificados encontrados en la etapa de análisis en este segundo prototipo, los métodos quedarían así:

Estos métodos son:

✓ Tab Inicial de configuración

✓ Menús de operación:

Menú Archivo:

Abrir

Guardar

Salir

Menú Modelo:

Parámetros

Variable Lingüística

Reglas

Resultados

Gráfico 3D

Menú Ayuda:

Acerca de

Ayuda

✓ Tab Modelo Fuzzy

✓ Crear variable lingüística de E/S:

Nombre

Unidades

Valor Mínimo

Valor Máximo

✓ Modificar variable lingüísticas de E/S

✓ Eliminar variables lingüísticas de E/S

✓ Crear conjunto borroso:

Nombre

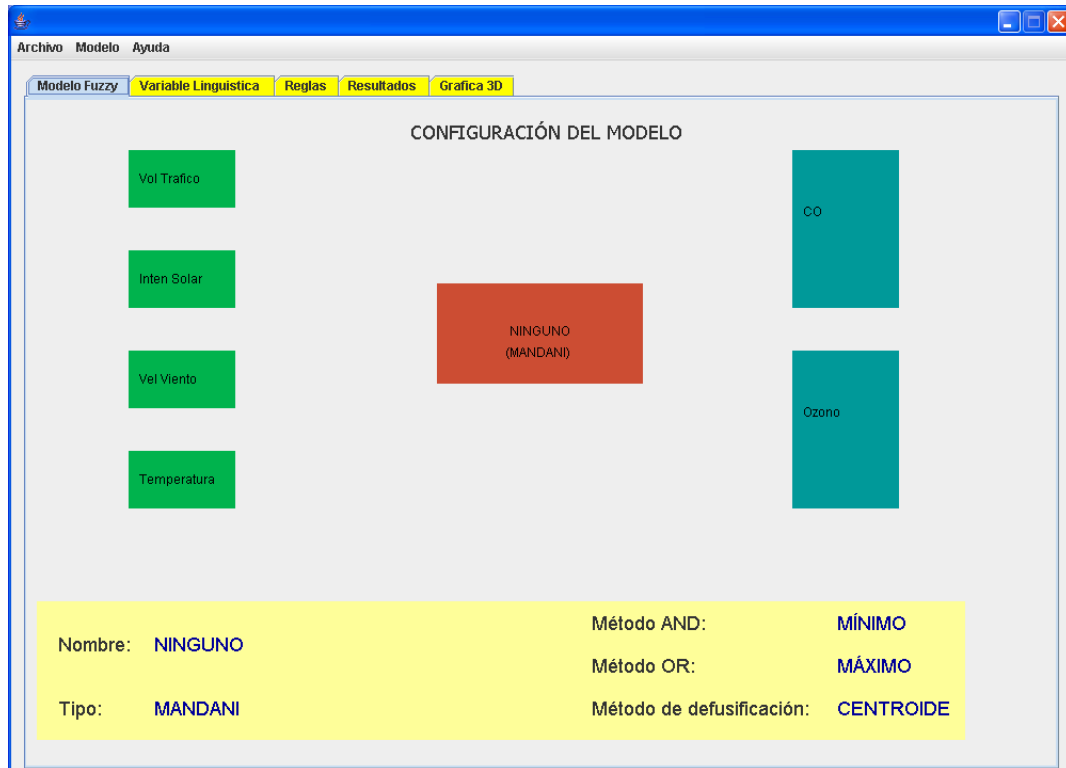
Función

- Color
- ✓ Modificar conjunto borroso
- ✓ Eliminar conjunto borroso
- ✓ Crear base de Reglas
- ✓ Definir un conector AND u OR
- ✓ Crear regla
  - Escoger Variable Lingüística de E/S
  - Conector
- ✓ Aplicar Modificadores:
  - NOT
  - DILATACION
  - CONCENTRACION
- ✓ Modificar regla
- ✓ Eliminar regla
- ✓ Introducir valores Crips
- ✓ Mostrar resultados en forma numérica y gráfica 2D.
- ✓ Mostrar resultados en forma 3D.
  - Escoger primera Variable Lingüística de entrada, escoger segunda Variable Lingüística de entrada y escoger variable Lingüística de Salida.
- ✓ Generar archivo de salida

Los cuales están contenidos en Tab para su fácil creación, uso y edición; como describimos a continuación:

### **Tab Modelo Fuzzy**

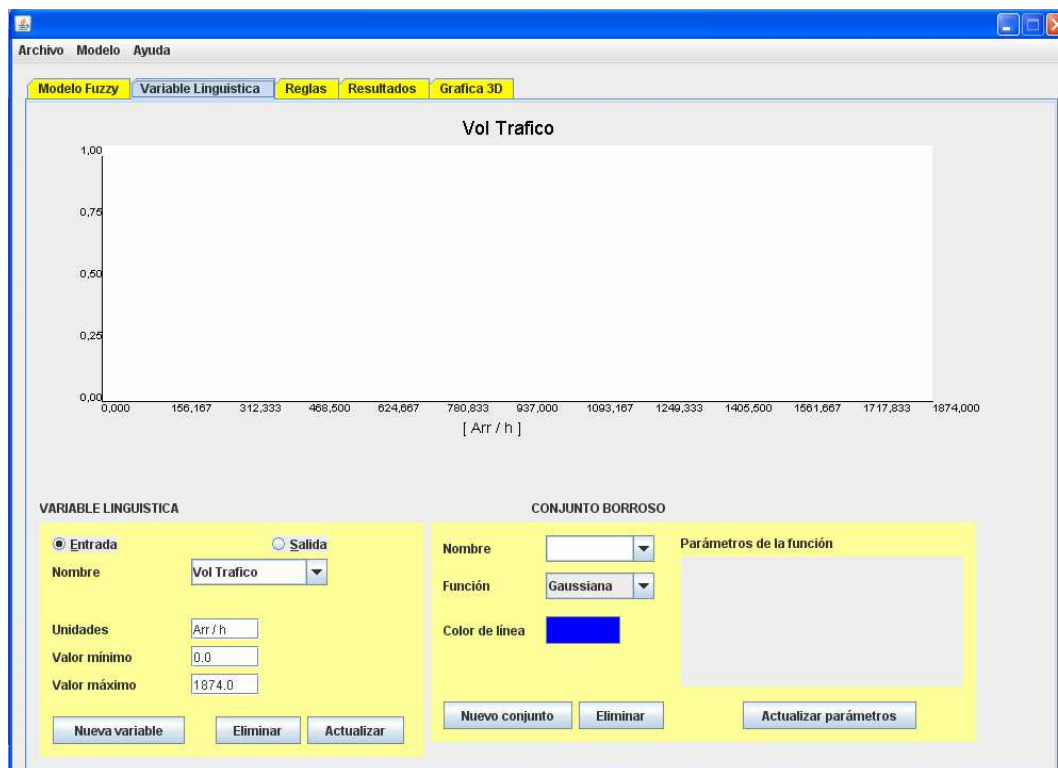
Contiene las características iniciales de un modelo de Lógica difusa como son Nombre del archivo, el método de defusificación tipo, muestra gráficamente las variables iniciales de entrada y de salida, el tipo de conexión



**Figura 30.** Tab Modelo Fuzzy.

### **Tab Editor de Variables Lingüísticas.**

Este Tab contiene de forma gráfica la posibilidad de crear, modificar y eliminar las variables lingüísticas seleccionadas y sus características o parámetros. Además permite crear, modificar y eliminar los conjuntos borrosos de estas variables lingüísticas ya sea de entrada y salida como vemos en la figura. Esto lo podemos hacer efectuar tanto dando valores desde el teclado como escogiendo gráficamente las opciones con el mouse haciendo click o arrastrando las gráficas habilitadas para ello.



**Figura 31.** Interface gráfica integrada a los subpaquetes.

Los métodos implementados quedarían de la siguiente manera:

### **Editor de Conjuntos Borrosos.**

Está implícito en el Tab editor de variables lingüísticas. Permite crear, modificar y eliminar los conjuntos borrosos, sus características o parámetros de estas variables lingüísticas ya sea de entrada y salida como vemos en la figura 32. Esto lo podemos hacer efectuar tanto dando valores desde el teclado como escogiendo gráficamente las opciones con el mouse haciendo click o arrastrando las gráficas habilitadas para ello.

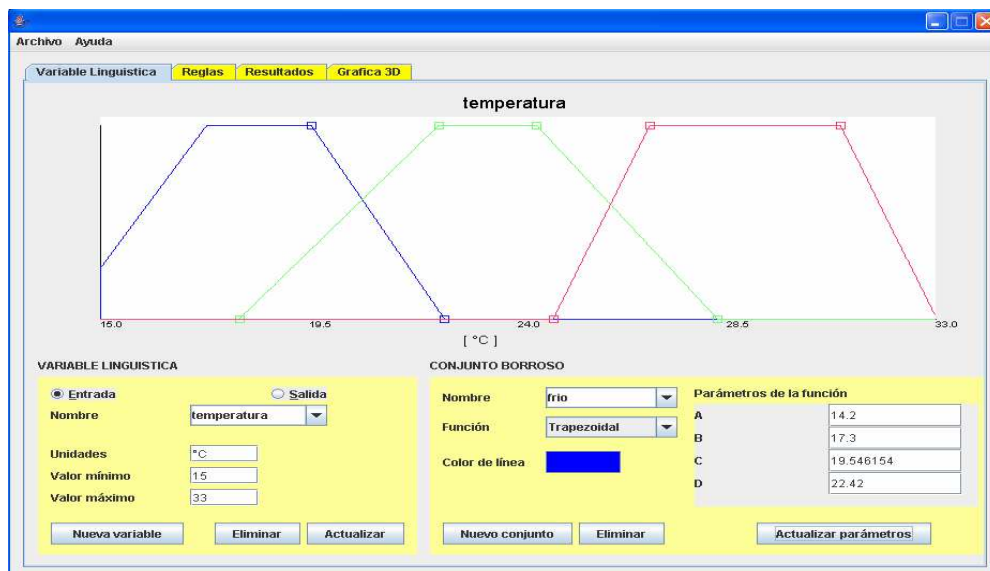


Figura 32. Módulo de parámetros y variables lingüísticas.

### Tab Editor de Reglas.

Este Tab permite crear, modificar y eliminar las reglas de la base de reglas.

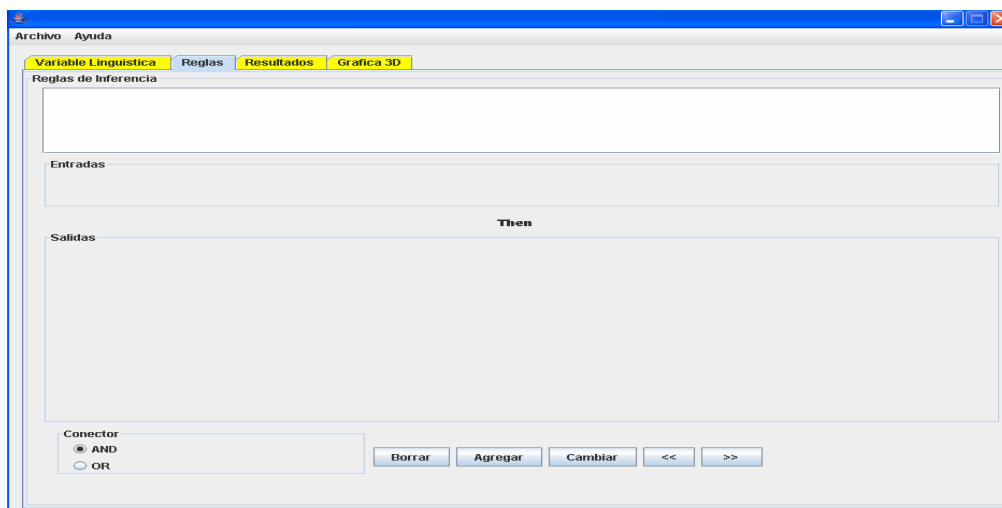
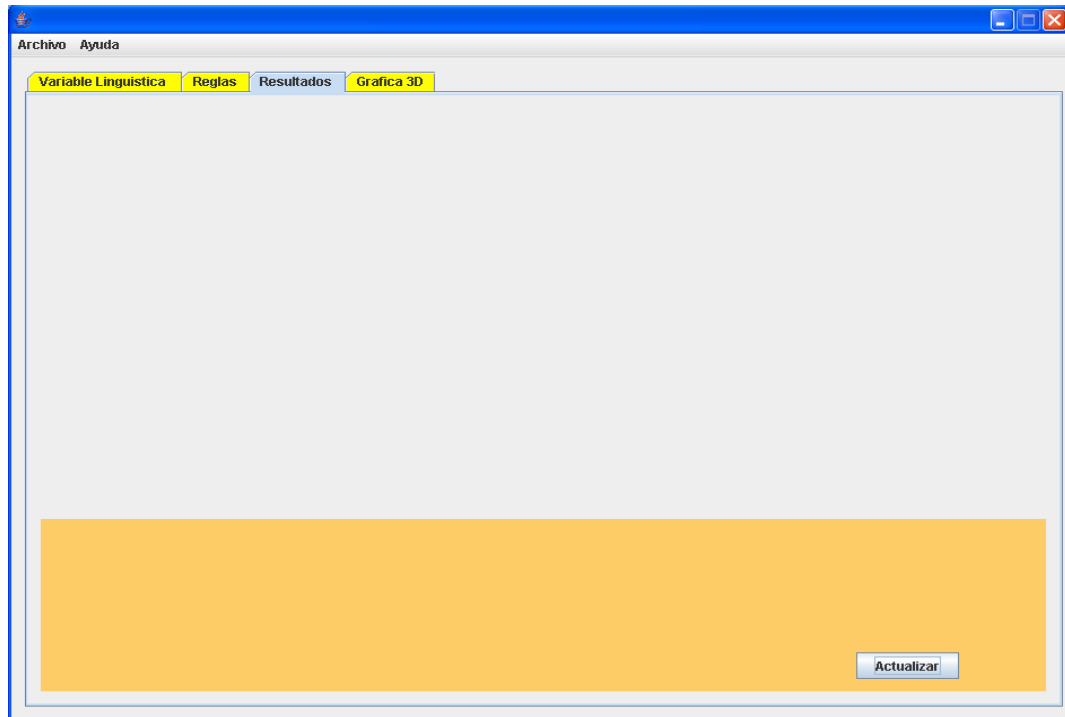


Figura 33. Tab editor de reglas.

### Tab Resultados.

Este Tab nos permite observar en forma gráfica las reglas. Es el que nos muestra las reglas creadas en el Tab reglas, nos permite introducir los valores crisp ya sean de forma numérica por teclado o utilizando el mouse para

arrastrar el valor crips, representado como una barra vertical sobre la variable lingüística activando los conjuntos borrosos correspondientes. (Figura 34).

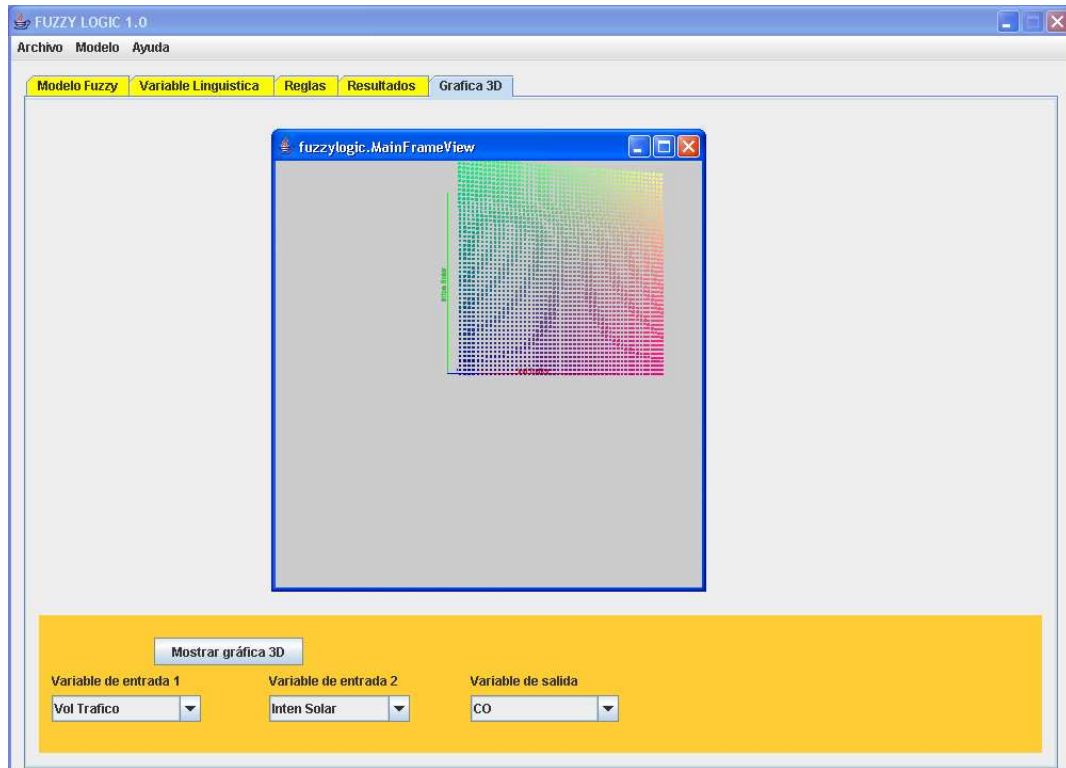


**Figura 34.** Tab de resultados.

### **Tab Gráfica 3D.**

Nos permite observar también en forma gráfica los resultados del modelo desarrollado pero en una gráfica 3D.

Esta gráfica 3D se desarrolló implementando un hilo debido a que debe tomar los valores a graficar y mostrar esos valores representados como líneas de un alto grado de visualización y perfil, por lo tanto, queda en una gráfica fuera del frame principal.



**Figura 35.** Tab Gráfica 3D.

#### **4.2.4. Validación final.**

Se realizaron las pruebas con el codirector del proyecto, director del grupo GIEMA, y catedrático de la materia Sistemas Expertos para la maestría en informática. Para ello utilizaremos un modelo sencillo con dos variables lingüísticas de entrada y una de salida y que mostraremos extensamente en la sección siguiente.

## 5. FASE DE PRUEBAS

### 5.1. Preliminar

Esta prueba se realizó a partir de un modelo básico que fue suministrado por el trabajo de grado de Maestría del cual hace parte el presente proyecto de grado.

Se trata de un modelo de la contaminación del aire, en el cual intervienen las variables que tienen una mayor incidencia para zona representativa (la calle 36 con carrera 15).

Para llevar a cabo estas pruebas, fué necesario hacer una revisión bibliográfica del mismo<sup>24</sup>. Como se describe a continuación.

Esta prueba será también el modelo final de muestra.

Aunque describiremos un modelo de cuatro variables de entrada y cinco de salida, para su mayor comprensión manipularemos un modelo con cuatro variables lingüísticas de entrada y dos de salida.

VARIABLES DE ENTRADA:

- Volumen del tráfico
- Intensidad solar
- Velocidad del viento
- Temperatura

---

<sup>24</sup> Jaimes, M. Castillo, E. F. Cachón, J. L. Artículo. Modelo difuso de la dispersión del aire en Bucaramanga. UIS. Bucaramanga.

Variables de salida:

- Material Particulado (PM<sub>10</sub>)
- Ozono (O<sub>3</sub>)
- Monóxido de carbono (CO)
- Óxido de nitrógeno (NO<sub>2</sub>)
- Dióxido de sulfuro (SO<sub>2</sub>)

## **5.2. MODELO DE LA CONTAMINACIÓN DEL AIRE.**

### **5.2.1. Datos de entrada**

Ahora necesitamos conocer los datos de entrada los cuales son los factores que generan o los que emiten u originan estos contaminantes criterio.

- **Volumen del tráfico**
- **Intensidad solar**
- **Velocidad del viento**
- **Temperatura**

### **5.2.2. Descripción de los datos de entrada.**

- **Volumen del tráfico**

Los datos utilizados para esta variable fueron suministrados por La Dirección de Transito y Transporte de Bucaramanga, pertenecientes al año 2005. Se consideró que en el mejor de los casos no pasa ningún carro, y que en el peor de los casos pasan más o menos 465 carros en un cuarto de hora, lo que resulta en un total de 1860 carros que transitan por esta intersección en una hora. Esta variable de entrada se partió en 3 conjuntos borrosos: Descongestionado, Semicongestionado y Congestionado. Descongestionado con un mayor grado de pertenencia en un intervalo de flujo entre 0 y 60 arribos en 15 minutos, Semicongestionado con mayor grado de pertenencia en un

intervalo de flujo entre 150 y 250 arribos en 15 minutos y Congestionado con un mayor de grado de pertenencia en un intervalo de flujo entre 400 y 475 arribos en 15 minutos.

#### • **Intensidad solar**

Los datos utilizados para definir esta variable de entrada fueron suministrados por la CDMB<sup>25</sup>, las unidades utilizadas para expresarla son  $W/m^2$  y fue dividida en 3 conjuntos borrosos: Mediodía, Aurora y Noche. Se escogió para todos los subconjuntos una función de pertenencia del tipo curva de Bell. Para el subconjunto mediodía se estableció un máximo grado de pertenencia con una irradiación entre 700 y 1000  $W/m^2$ , mientras que para el subconjunto noche se estableció un mínimo grado de pertenencia, pues la radiación solar es nula en la noche. Asimismo, para el subconjunto aurora se estableció el menor grado de pertenencia en irradiaciones altas, entre 6:00 a 8:00 a.m. y 5:30 a 7:00 p.m.

#### • **Velocidad del viento**

Los datos utilizados fueron suministrados por la CDMB y tomados en 2003, por la estación de monitoreo centro, ubicada en la carrera 15, en la terraza de Mercadefam<sup>26</sup>. Los datos suministrados indican la velocidad del viento a 2 m de altura (recomendada para estudios en calidad del aire) y tienen las modificaciones necesarias en los cálculos para tener en cuenta las disminuciones en la velocidad del viento que provocan los elementos altos (e.g., edificios).

Esta variable de entrada se partió en 3 conjuntos borrosos: lento (con un mayor grado de pertenencia en un intervalo entre 0 y 0,5 m/s), moderado (con mayor grado de pertenencia en un intervalo entre 1,6 y 2,1 m/s) y rápido (con un mayor de grado de pertenencia en un intervalo 3,3 y 3,6 m/s).

---

<sup>25</sup> CDMB: <http://www.cdm.gov.co/conozca/painas/ibuca.htm>

<sup>26</sup> Amado, J. P. Castro, J. C. y otros (2004). Trabajo de grado. Estudio Comparativo de Contaminación atmosférica por la operación de un S.I.T.M en Bucaramanga. UIS. Bucaramanga

**Tabla 1.** Velocidad del viento estación de monitoreo Terraza Mercadefam

<b>HORA</b>	<b>m/s</b>
6:00	0.8
7:00	1.0
8:00	1.3
9:00	1.5
10:00	1.8
11:00	1.0
12:00	1.3
13:00	1.3
14:00	1.4
15:00	1.5
16:00	1.5
17:00	1.3
18:00	1.1

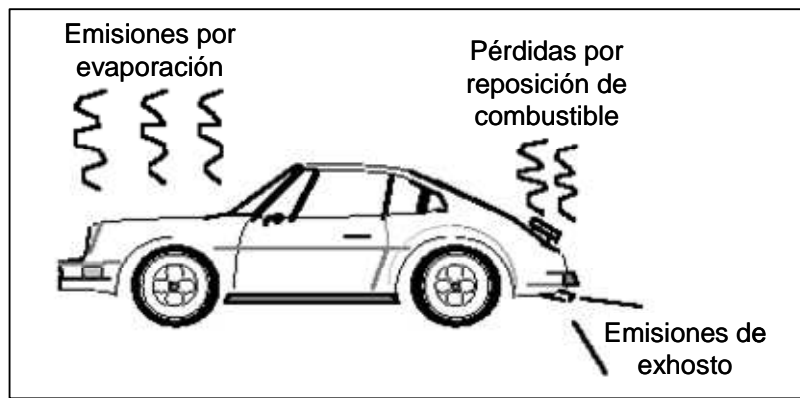
• **Temperatura**

Para esta variable de entrada los datos fueron suministrados por la CDMB, las unidades utilizadas son °C y se dividió en 3 conjuntos borrosos, para los cuales se escogió una función de pertenencia del tipo Gauss. Para el subconjunto Frío se estableció un mayor grado de pertenencia en un intervalo entre 18,5 y 19,5 °C; para el subconjunto Templado un mayor grado de pertenencia en un intervalo entre 20 y 21° C y para el subconjunto Calor un valor de máxima pertenencia en el intervalo entre 24,5 y 25,5° C.

### 5.2.3. Contaminantes criterio.

La contaminación que proviene de las fuentes móviles (e.g., automóviles, **Figura 36**) se debe a los productos secundarios del proceso de combustión (que son emitidos por el tubo de escape) y de la evaporación del combustible. Los principales contaminantes emitidos por las fuentes móviles incluidos en la normatividad Colombiana<sup>27</sup> son:

- Material Particulado (PM<sub>10</sub>)
- Ozono (O<sub>3</sub>)
- Monóxido de Carbono (CO)
- Óxidos de Nitrógeno (NO<sub>2</sub>)
- Dióxido de Azufre (SO<sub>2</sub>)



**Figura 36.** Contaminantes emitidos por los automóviles.

### 5.2.4. Gases contaminantes.

A continuación, para entender mejor los modelos que plantearemos, se da una breve explicación sobre los contaminantes criterio mencionados anteriormente.

<sup>27</sup> Quiroga, L. Y. Téllez, Z. C. Trabajo de grado. Valoración estadística y comparativa del cálculo del Índice de la calidad del aire. UIS. Bucaramanga. (2005).

<b>CONTAMINANTES DEL AIRE</b>	<b>Material Particulado (PM<sub>10</sub>)</b>
	<p>Corresponden a una amplia clase de sustancias de distinta naturaleza química y física que existen como partículas sólidas o líquidas de una variada distribución de tamaños. Son emitidas directamente o pueden resultar de la transformación en la atmósfera de sustancias gaseosas como óxidos de azufre, óxidos de nitrógeno y compuestos orgánicos volátiles.</p> <p>El material particulado puede ser: PTS (partículas totales en suspensión), PM<sub>10</sub> (Partículas menores a 10 micrómetros, ≤ 10 μm) y PM<sub>2.5</sub> (Partículas menores a 2,5 micrómetros, ≤ 2,5 μm). A partir de un tamaño menor a 10 μm (PM<sub>10</sub>), la fracción comienza a ser significativa en términos de salud para las personas. En este caso, estas partículas entran en el aparato respiratorio y están asociadas con muchas enfermedades respiratorias como bronquitis, asma, enfisema, pulmonía y enfermedades cardíacas.</p>
	<b>Ozono (O<sub>3</sub>)</b>
	<p>El O<sub>3</sub> es un contaminante secundario muy reactivo; reacciona con las plantas y los tejidos animales de forma perjudicial, irrita los ojos, puede reducir la función pulmonar en los humanos (a partir de 150 μg/m<sup>3</sup>) y detiene el crecimiento de las plantas hasta reducir los campos agrícolas y las cosechas en horticuivos (a partir de 30 ppb). Si se respira en grandes cantidades, es tóxico.</p> <p>A temperatura y presión ambientales el ozono es un gas de olor acre e incoloro, que en grandes concentraciones puede volverse azulado. El ozono troposférico se forma a partir de NO<sub>x</sub> (óxidos de nitrógeno) y VOCs (compuestos orgánicos volátiles, como el formaldehído), los cuales se disocian formando radicales que reaccionan con el oxígeno molecular, dando lugar a la formación del O<sub>3</sub>.</p>
	<b>Dióxido de Nitrógeno (NO<sub>2</sub>)</b>
	<p>Bajo las condiciones de alta temperatura y presión que imperan en el motor, los átomos de nitrógeno y oxígeno del aire reaccionan para formar varios óxidos de nitrógeno, que se conocen de manera colectiva como NO<sub>x</sub>. Los óxidos de nitrógeno, como los hidrocarburos, son precursores de la formación de ozono. El NO<sub>2</sub> sirve como indicador de las concentraciones de varios compuestos químicamente similares (óxidos de nitrógeno, NO<sub>x</sub>).</p> <p>El NO<sub>2</sub> es un agente sumamente oxidante, soluble en agua, de color café-rojizo, forma partículas en la atmósfera, contribuyendo al problema del material particulado y al deterioro de la visibilidad, además es uno de los principales causantes de la lluvia ácida.</p> <p>El NO<sub>2</sub> irrita los pulmones, causando bronquitis, pulmonía y una reducción significativa de la resistencia respiratoria a las infecciones. La exposición frecuente a concentraciones mayores a las encontradas normalmente en el aire, puede causar un incremento en la incidencia de enfermedades respiratorias en los niños, agravamiento de afecciones en individuos asmáticos y con enfermedades respiratorias crónicas.</p>

<b>CONTAMINANTES DEL AIRE</b>	<b>Dióxido de Azufre (SO<sub>2</sub>)</b>
	<p>El SO<sub>2</sub> es un gas incoloro de olor característico, es una especie de vida corta en la atmósfera y se reconoce como un contaminante porque reacciona con el agua ( H<sub>2</sub>O ) para formar ácido sulfúrico. Todos los combustibles usados por el hombre contienen azufre en pequeñas o medianas cantidades; al quemarse estos combustibles, una gran cantidad del azufre contenido forma SO<sub>2</sub>, que al ser liberado a la atmósfera contribuye en la formación de la lluvia ácida. El SO<sub>2</sub> sirve como indicador de las concentraciones de varios compuestos químicamente similares (óxidos de azufre, SOx).</p> <p>En altas concentraciones el SO<sub>2</sub> puede ocasionar dificultad para respirar, humedad excesiva en las mucosas conjuntivas, irritación severa en vías respiratorias y los pulmones por formación de partículas de ácido sulfúrico, ocasionando vulnerabilidad en las defensas; los grupos de la población más sensibles al SO<sub>2</sub> son los niños, ancianos, asmáticos y personas con enfermedades pulmonares crónicas como bronquitis y enfisema.</p>
	<b>Monóxido de Carbono (CO)</b>
	<p>El CO es un gas incoloro e inodoro, producto de la combustión incompleta y ocurre cuando el carbono en el combustible se oxida parcialmente en lugar de oxidarse por completo y formar dióxido de carbono (CO<sub>2</sub>). El CO reduce el flujo de oxígeno en el torrente sanguíneo y es particularmente peligroso para las personas con padecimientos cardíacos, en altas concentraciones afecta el sistema respiratorio y puede causar la muerte</p>

### 5.2.5. Unidades de los contaminantes criterio

**Tabla 2.** Unidades de los contaminantes criterio

<b>CONTAMINANTE</b>	<b>ABREVIATURA</b>	<b>UNIDAD</b>
Material Particulado	PM <sub>10</sub>	µg/m <sup>3</sup>
Ozono	O <sub>3</sub>	ppb
Monóxido de Carbono	CO	ppb
Óxido de Nitrógeno	NO <sub>2</sub>	ppm
Dióxido de Azufre	SO <sub>2</sub>	ppb

### 5.3. EJECUCIÓN DE PRUEBAS

Para una mejor y mas fácil implementación del modelo utilizaremos las siguientes abreviaturas (Sigla utilizada para el compuesto) como:

**Tabla 3.** Abreviaturas para Variables Lingüísticas de entrada

<b>Variable lingüística</b>	<b>Abreviatura</b>
Volumen del tráfico	VT
Intensidad solar	IS
Velocidad del viento	VV
Temperatura	T

**Tabla 4.** Abreviaturas para Variables Lingüísticas de salida

<b>Variable lingüística</b>	<b>Abreviatura</b>
Material Particulado	PM <sub>10</sub>
Ozono	O <sub>3</sub>
Monóxido de Carbono	CO
Óxido de Nitrógeno	NO <sub>2</sub>
Dióxido de Azufre	SO <sub>2</sub>

#### 5.3.1. Características del archivo generado

Recibe el nombre de "PrototipoContaminación.txt"

Variables de entrada = 4

Variables de salida = 2

Tamaño del archivo = 556 Kb

#### 5.3.2. Características del prototipo

Parámetros utilizados para las Variables lingüísticas y sus respectivos conjunto borrosos.

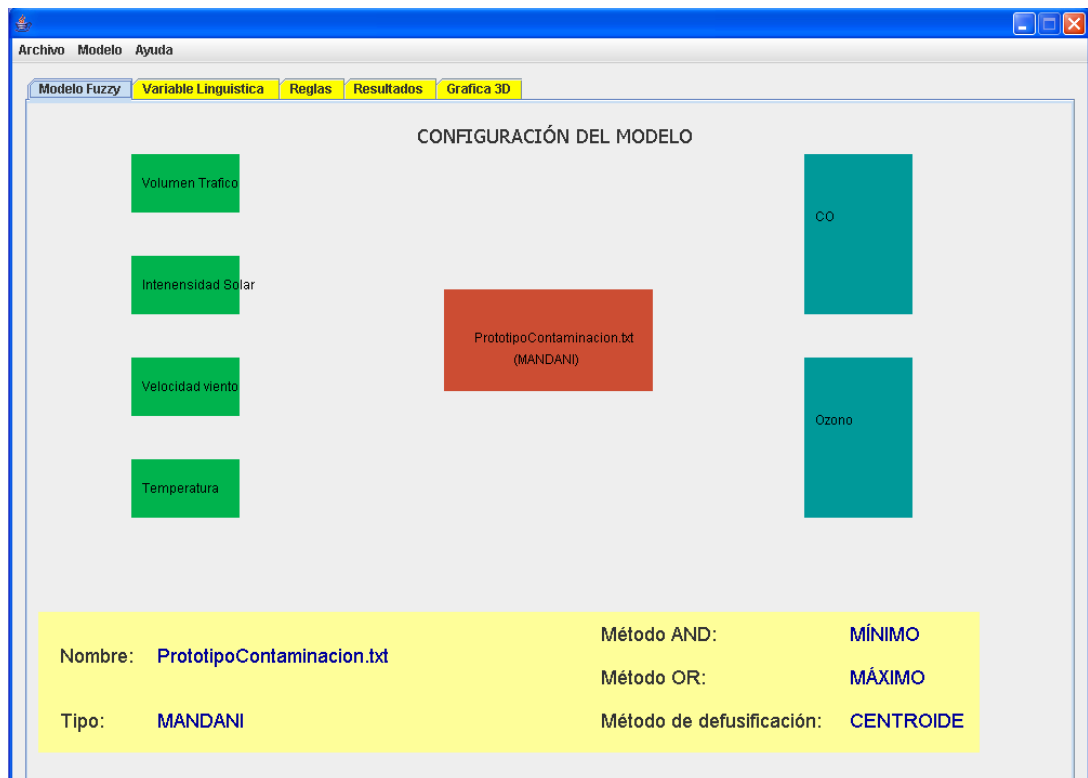
<b>Volumen del Tráfico</b>	
Tipo variable	Entrada
Unidades	Arribos/hora
Valor mínimo	0
Valor máximo	1874
Conjuntos borrosos	Parámetros (Función de pertenencia Gauss)
Descongestionado	C = 0.8    Sigma = 255
Semicongestionado	C = 800    Sigma = 380
Congestionado	C = 1836    Sigma = 344
<b>Intensidad Solar</b>	
Tipo variable	Entrada
Unidades	Wattios/m <sup>2</sup>
Valor mínimo	16
Valor máximo	920
Conjuntos borrosos	Parámetros (Función de pertenencia Bell)
Descongestionado	A = 83    B = 2    C = 17
Semicongestionado	A = 326    B = 5    C = 21
Congestionado	A = 303    B = 3    C = 812
<b>Velocidad del viento</b>	
Tipo variable	Entrada
Unidades	m/s
Valor mínimo	0.21
Valor máximo	3.5
Conjuntos borrosos	Parámetros (Función de pertenencia Gauss)
Lento	C = 0.2    Sigma = 0.6
Moderado	C = 1.9    Sigma = 0.7
Rápido	C = 3.5    Sigma = 0.5
<b>Temperatura</b>	
Tipo variable	Entrada
Unidades	°C
Valor mínimo	19
Valor máximo	26
Conjuntos borrosos	Parámetros (Función de pertenencia Trapecio)
Frió	A=17 B=18 C=20 D=21
Templado	A=20 B=21 C=23 D=24
Calor	A=23 B=24 C=26 D=27
<b>CO</b>	
Tipo variable	Salida
Unidades	ppb
Valor mínimo	1
Valor máximo	4
Conjuntos borrosos	Parámetros (Función de pertenencia Gauss)
Baja	C = 0.9    Sigma = 0.5
Media	C = 2.6    Sigma = 0.5
Alta	C = 4    Sigma = 0.6
<b>O<sub>3</sub></b>	
Tipo variable	Salida
Unidades	ppb
Valor mínimo	0
Valor máximo	38
Conjuntos borrosos	Parámetros (Función de pertenencia Gauss)
Baja	C = 0    Sigma = 7
Media	C = 17    Sigma = 7.6
Alta	C = 38    Sigma = 7

Variable Lingüística	Abreviatura	Tipo	Unidades	Rango [Valor Max.-V. Min]	Conjuntos Borrosos		
					Etiqueta lingüística	Función	Parámetros
Volumen Del Tráfico	VT	ENTRADA	Arribos / hora	[0 1874]	Descongestionado Semicongestionado Congestionado	Gaussiana	C=0.8, Sigma=255 C=800, Sigma=380 C=1836, Sigma=344
Intensidad Solar	IS		Wattios/m <sup>2</sup>	[16 920]	Noche Mediodía Noche	Curva de Bell	A=83, B=2, C=17 A=326, B=5, C=21 A=303, B=3, C=812
Velocidad del Viento	VV		m/s	[0.21 3.5]	Lenta Moderada Rápida	Gaussiana	C= 0.2, Sigma = 0.6 C= 1.9, Sigma = 0.7 C= 3.5, Sigma = 0.5
Temperatura	T		° C	[19 26]	Frío Templado Calor	Trapezio	A=17 B=18 C=20 D=21 A=20 B=21 C=23 D=24 A=23 B=24 C=26 D=27
Monóxido de carbono	CO	SALA	ppb	[1 4]	Baja Media Alta	Gaussiana	C = 0.9 Sigma = 0.5 C = 2.6 Sigma = 0.5 C = 4 Sigma = 0.6
Ozono	O <sub>3</sub>	IDA	ppb	[0 38]	Baja Media Alta	Gaussiana	C= 0 Sigma = 7 C=17 Sigma = 7.6 C=38 Sigma = 7

**Tabla 5.** Sinopsis de parámetros del modelo.

## Tab Inicial

Configuración del modelo fuzzy; contiene el nombre del archivo: PrototipoContaminacion.txt, tipo de proceso en este caso Mandani, las variables lingüísticas de retarda y salida y sus respectivos nombres, método AND = mínimo y OR = máximo, método de desemborronado = centroide.



**Figura 37.** Configuración del modelo.

## Variables Lingüísticas

Variables Lingüísticas de entrada:

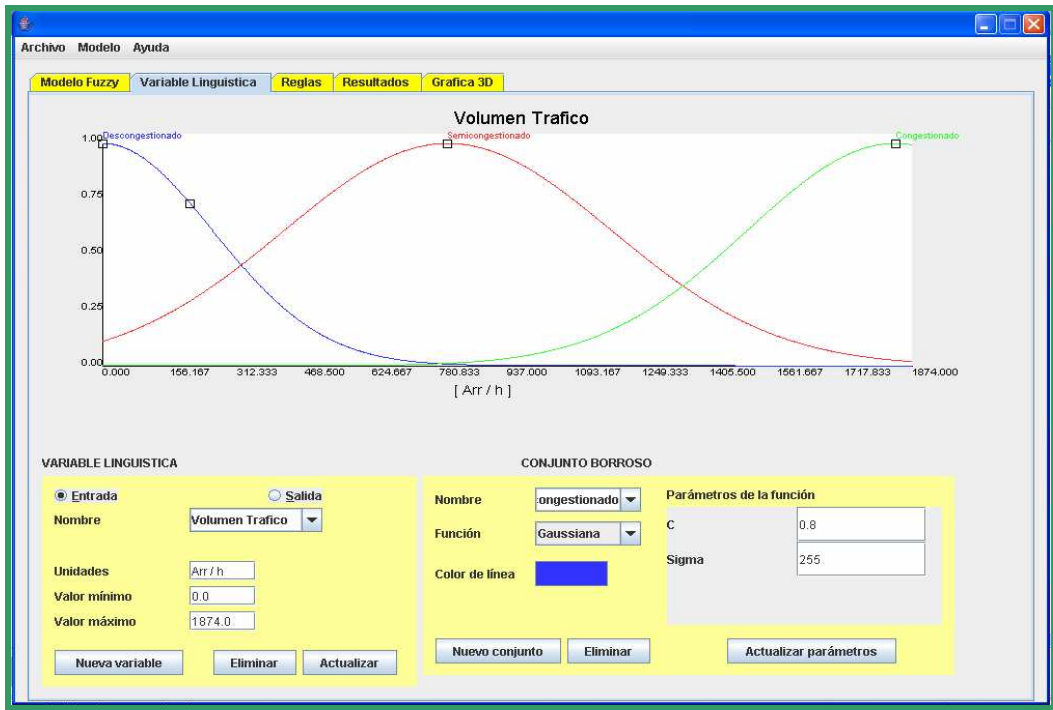


Figura 38. Variable de entrada Volumen Tráfico.

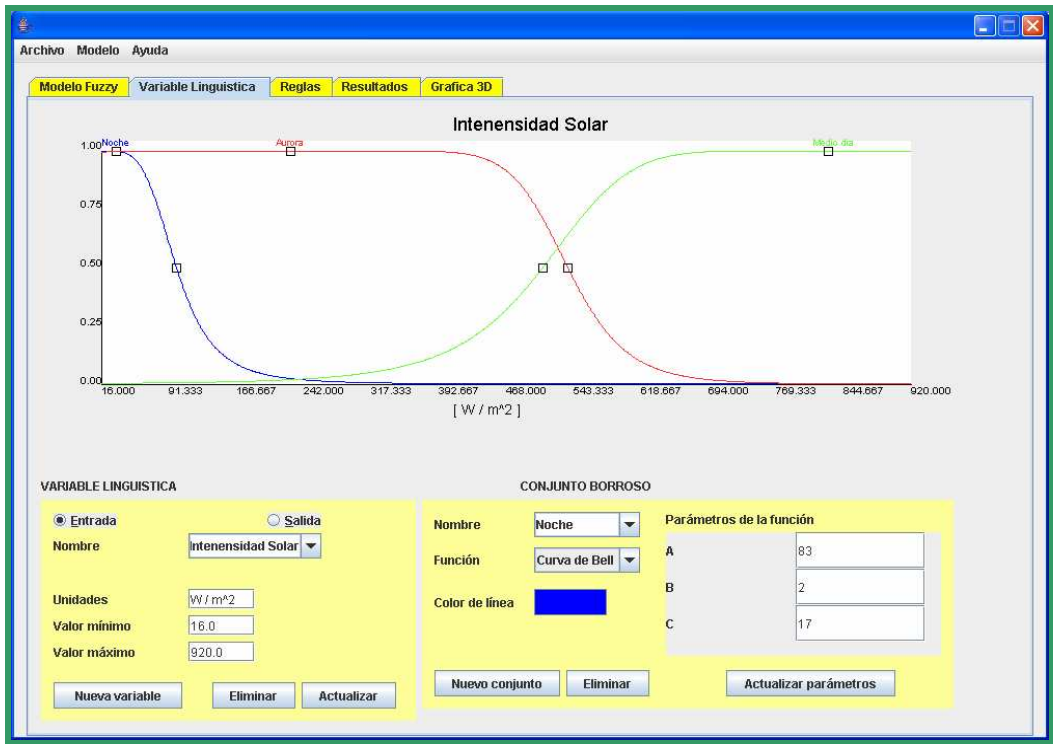


Figura 39. Variable de entrada Intensidad Solar

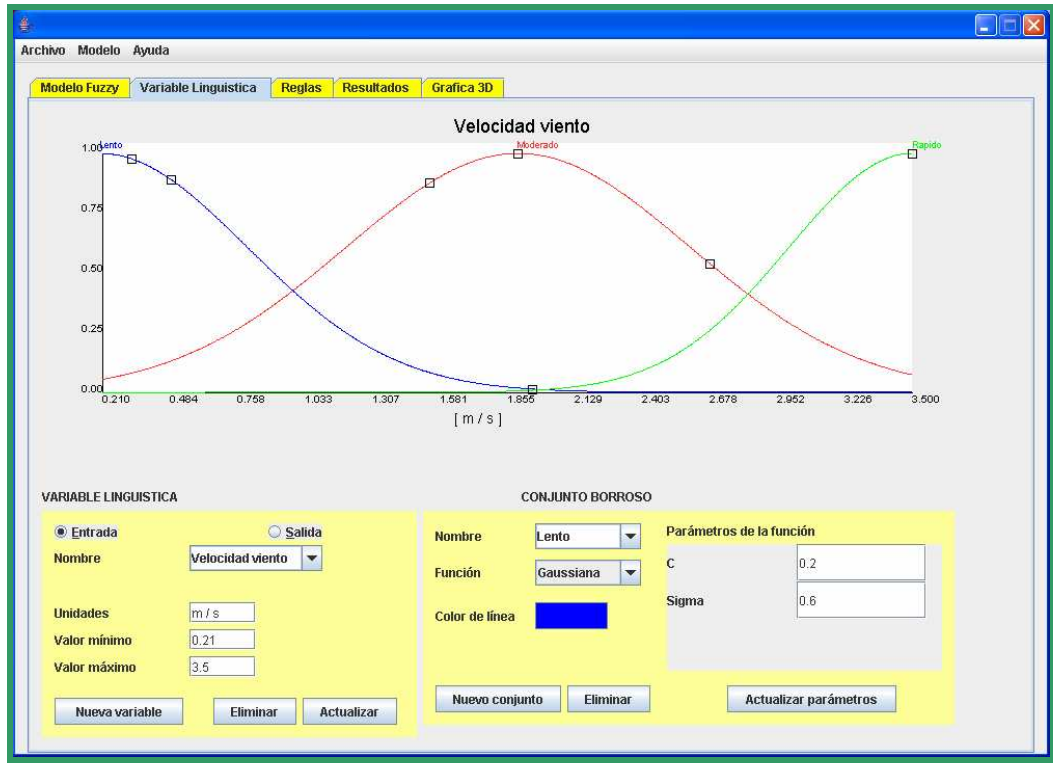


Figura 40. Variable de entrada Velocidad viento

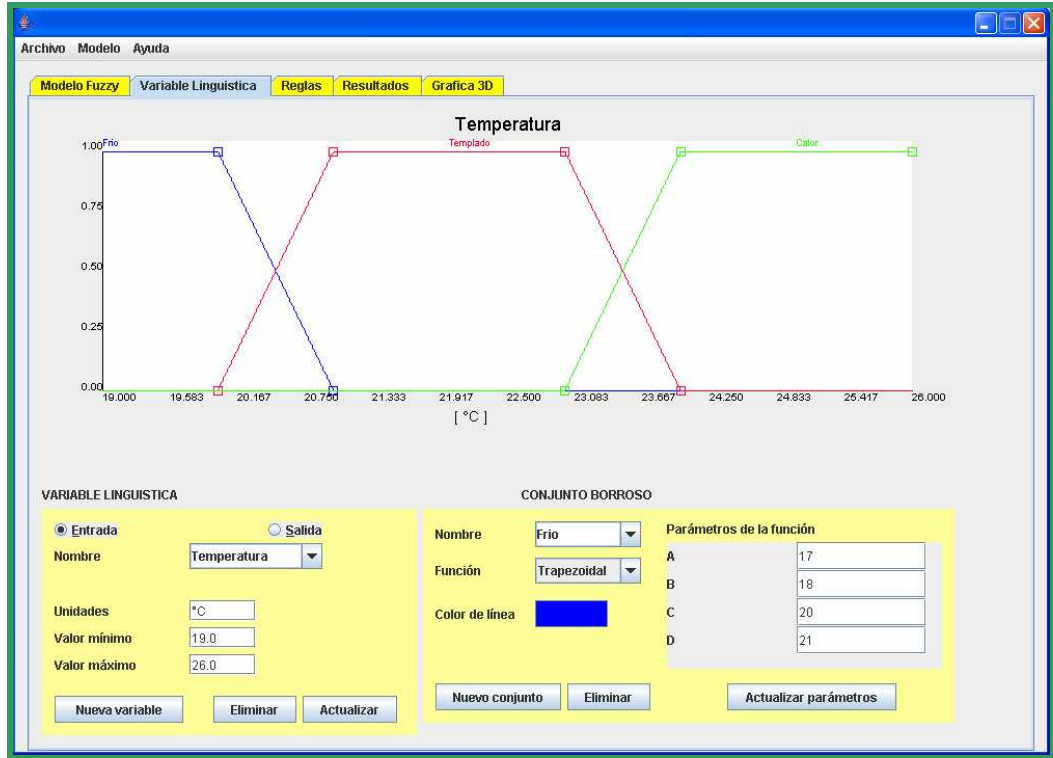


Figura 41. Variable de entrada Temperatura

Variables Lingüísticas de salida:



Figura 42. Variable de salida concentración de CO



Figura 43. Variable de salida concentración de Ozono

## Base de Reglas:

Para esta prueba se creo una base de reglas teniendo en cuenta un resultado lógico para ellas y la supervisión del codirector de este proyecto.

1. If (VT is Semicongestionado) AND (IS is Aurora) AND (VV is Moderado) AND (T is Templado) then (CO is Media) (Oz is Media)
2. If (VT is Congestionado) AND (IS is Medio día) AND (VV is Lento) AND (T is Calor) then (CO is Alta) (Oz is Alta)
3. If (VT is Descongestionado) AND (IS is Noche) AND (VV is Rápido) AND (T is Calor) then (CO is Baja) (Oz is Baja)
4. If (VT is Congestionado) AND (IS is Noche) AND (VV is Lento) AND (T is Frió) then (CO is Alta)
5. If (VT is Congestionado) AND (IS is Noche) AND (VV is Rápido) AND (T is Calor) then (CO is Alta) (Oz is Baja)
6. If (VT is Semicongestionado) AND (IS is Medio día) AND (VV is Rápido) AND (T is Frió) then (CO is Medio) (Oz is Alta)

The screenshot shows a software window titled "Modelo Fuzzy" with a menu bar (Archivo, Modelo, Ayuda) and tabs (Modelo Fuzzy, Variable Linguistica, Reglas, Resultados, Grafica 3D). The "Reglas" tab is active, displaying a list of five inference rules under "Reglas de Inferencia".

**Reglas de Inferencia**

1. If (Volumen Tráfico is Descongestionado ) AND ( Intensidad Solar is Noche ) AND ( Velocidad viento is Lento ) AND ( Temperatura is Frio ) then ( CO is Baja ) ( Ozono is Baja )
2. If ( Volumen Tráfico is Semicongestionado ) AND ( Intensidad Solar is Aurora ) AND ( Velocidad viento is Moderado ) AND ( Temperatura is Templado ) then ( CO is Media ) ( Ozono is Media )
3. If ( Volumen Tráfico is Congestionado ) AND ( Intensidad Solar is Medio día ) AND ( Velocidad viento is Rápido ) AND ( Temperatura is Calor ) then ( CO is Alta ) ( Ozono is Alta )
4. If ( Volumen Tráfico is Congestionado ) AND ( Intensidad Solar is Medio día ) AND ( Velocidad viento is Moderado ) AND ( Temperatura is Calor ) then ( CO is Alta ) ( Ozono is Alta )
5. If ( Volumen Tráfico is Congestionado ) AND ( Intensidad Solar is Medio día ) AND ( Velocidad viento is Moderado ) AND ( Temperatura is Calor ) then ( CO is Alta ) ( Ozono is Alta )

**Entradas**

Volumen Tráfico	Intensidad Solar	Velocidad viento	Temperature
Descongestionado	Noche	Lento	Frio
Semicongestionado	Aurora	Moderado	Templado
Congestionado	Medio día	Rápido	Calor
Ninguno	Ninguno	Ninguno	Ninguno

Below the table are four dropdown menus, each currently set to "NINGUNO".

**Then**

CO	Ozono
Baja	Baja
Media	Media
Alta	Alta
Ninguno	Ninguno

Below the table are two dropdown menus, each currently set to "NINGUNO".

**Conector**

AND  
 OR

Buttons: Borrar, Agregar, Cambiar, <<, >>

Figura 44. Base de reglas

## Resultados

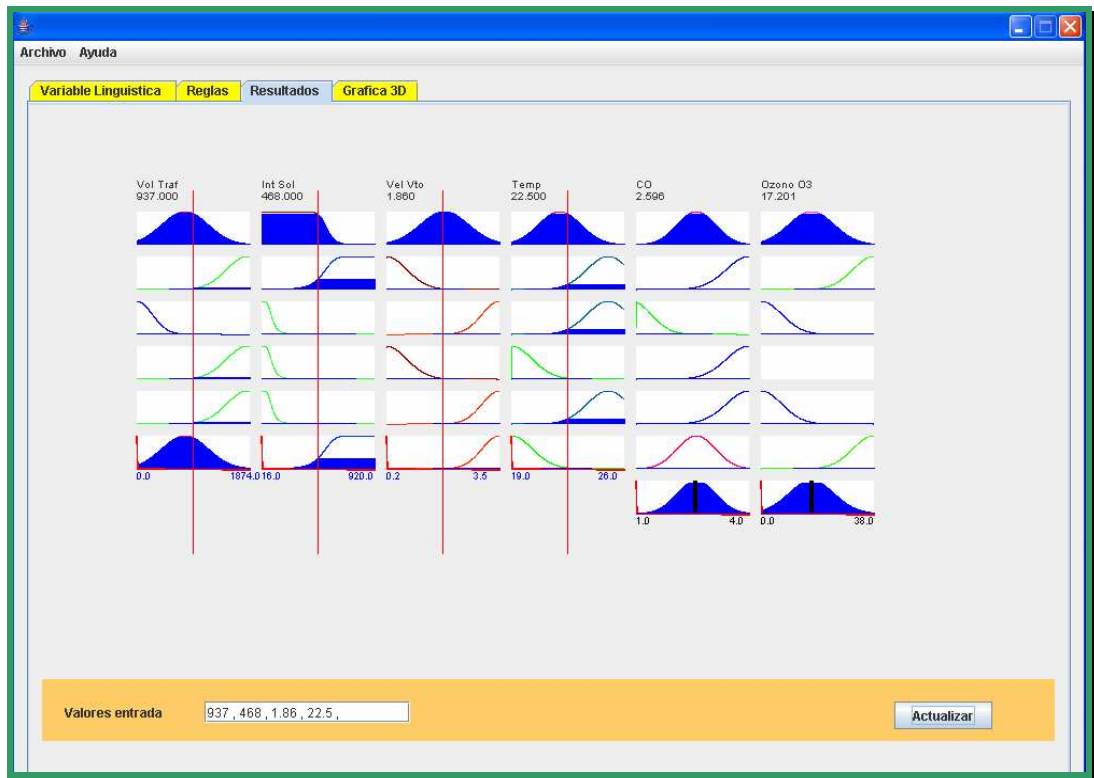


Figura 45. Resultados gráficos y numéricos.

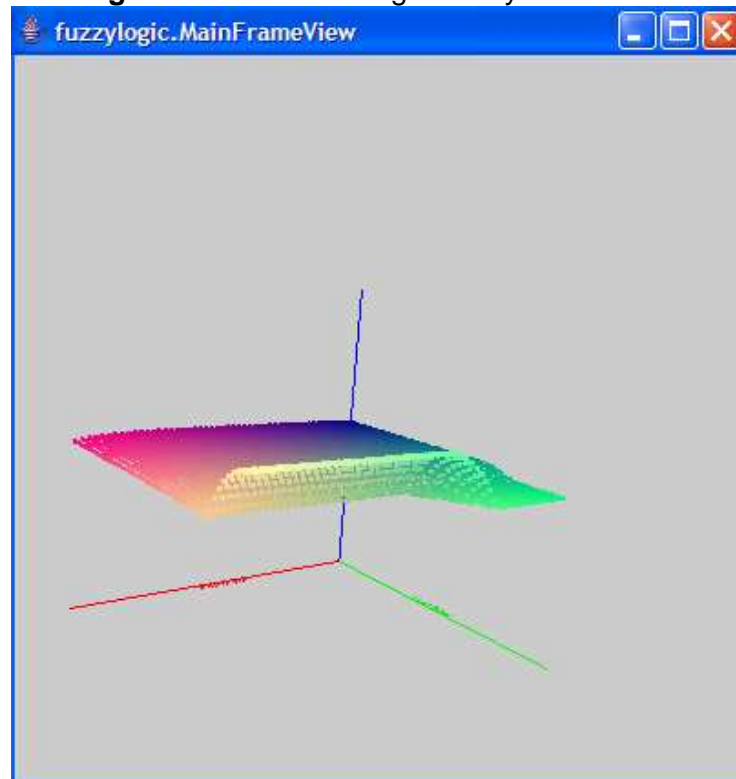
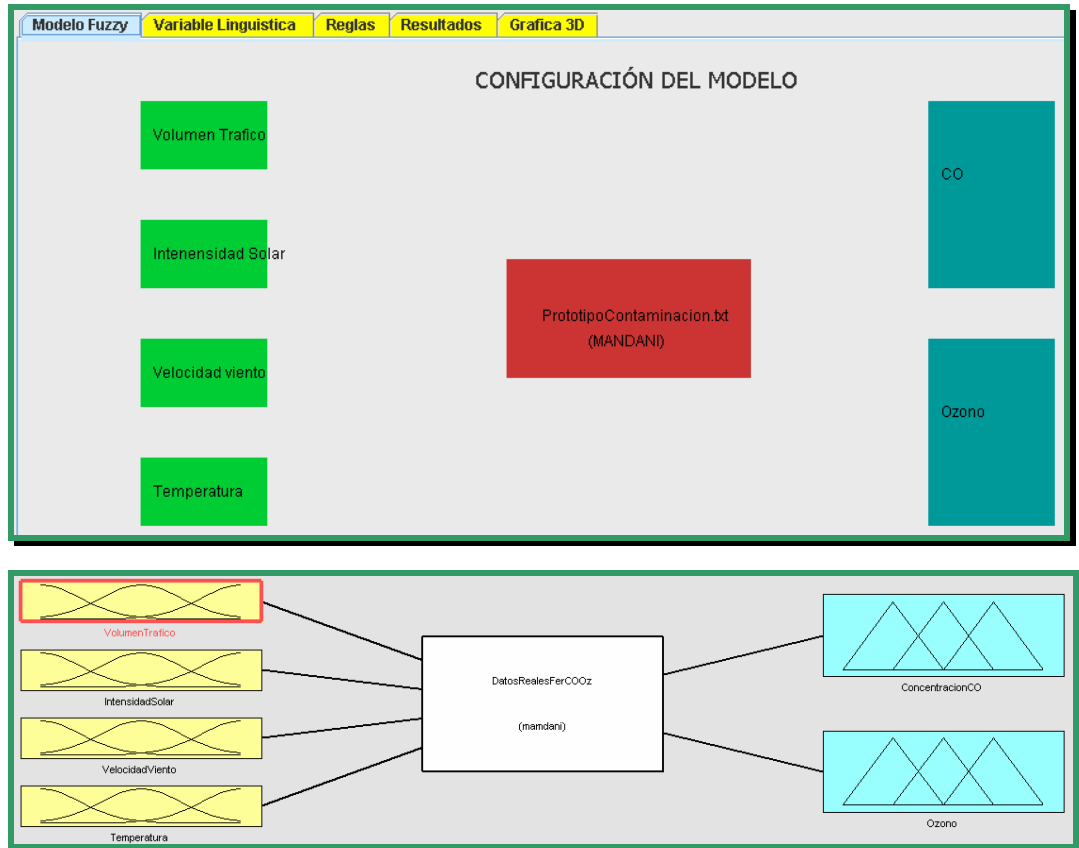


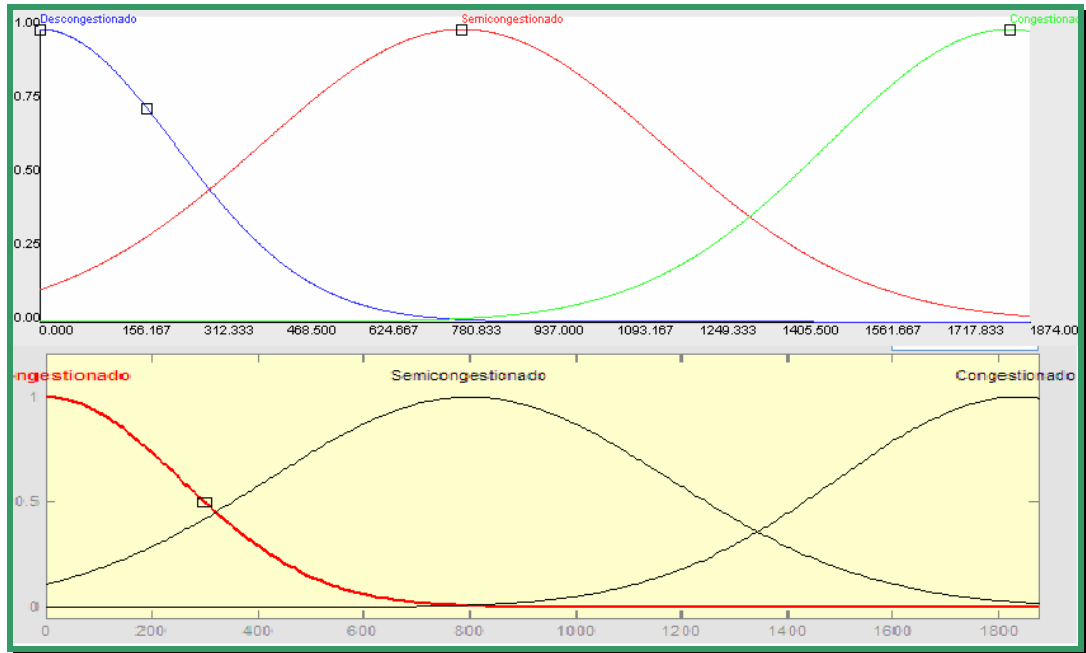
Figura 46. Gráfico tridimensional.

## 5.4. RESULTAODS

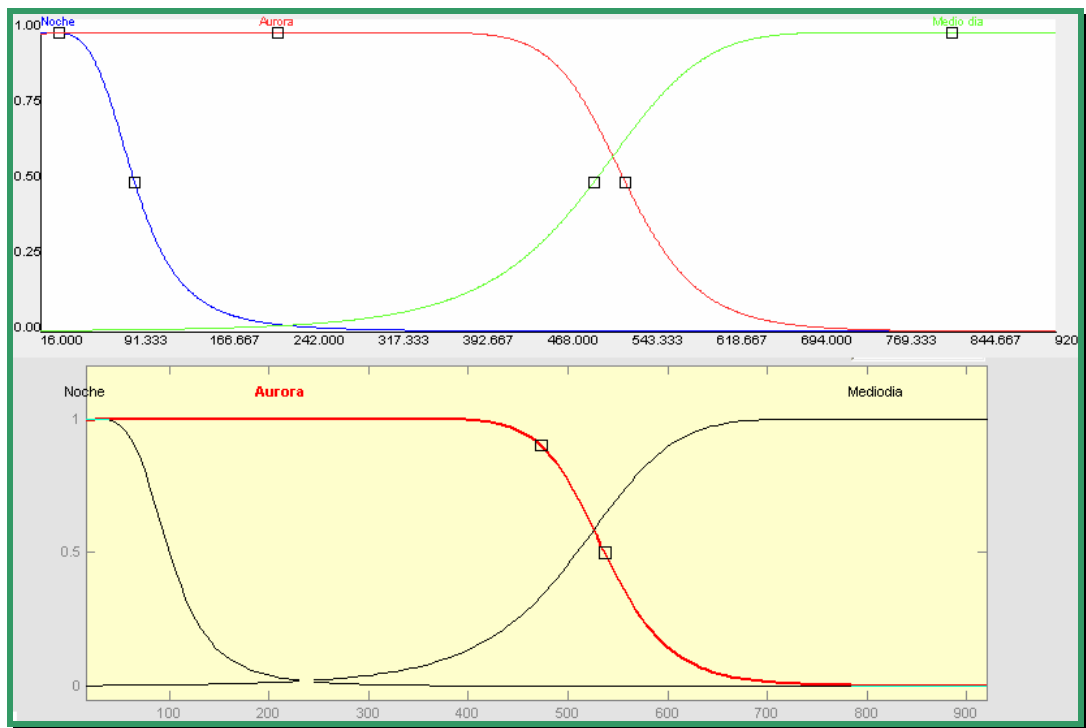
Para verificar los resultados hacemos una comparación con el Toolbox Fuzzy Logic de Matlab.



**Figura 47.** Configuración inicial: arriba Prototipo vs abajo Matlab



**Figura 48.** Volumen tráfico: arriba prototipo vs abajo Matlab



**Figura 49.** Intensidad solar: arriba Prototipo vs abajo Matlab

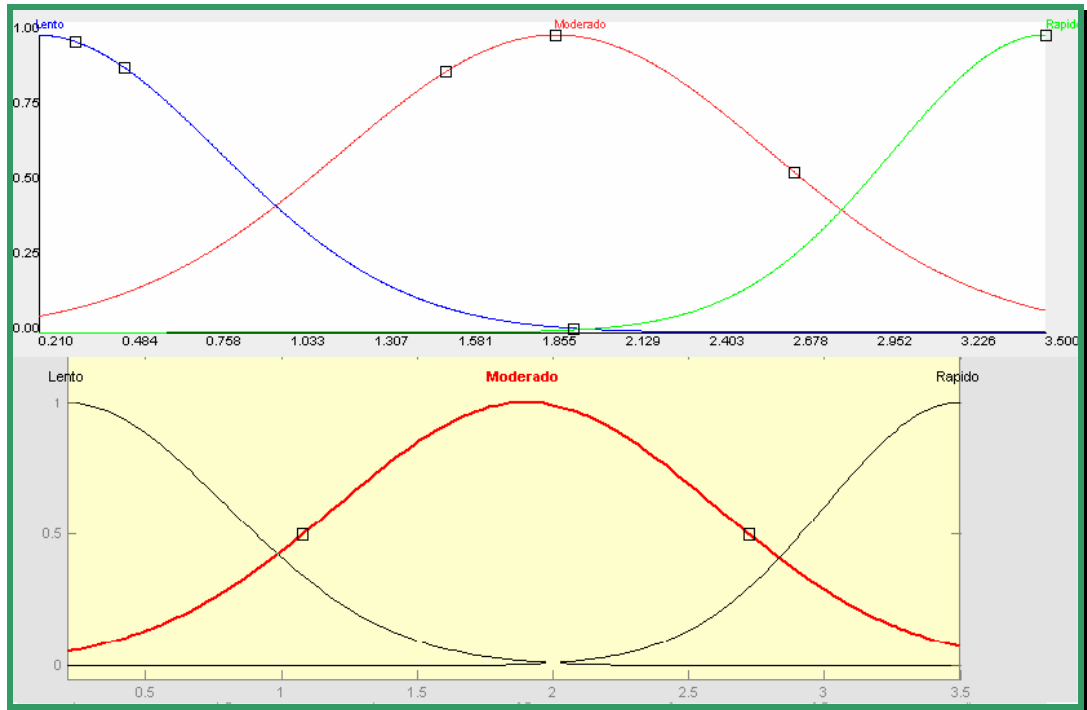


Figura 50. Velocidad Viento: arriba Prototipo vs abajo Matlab

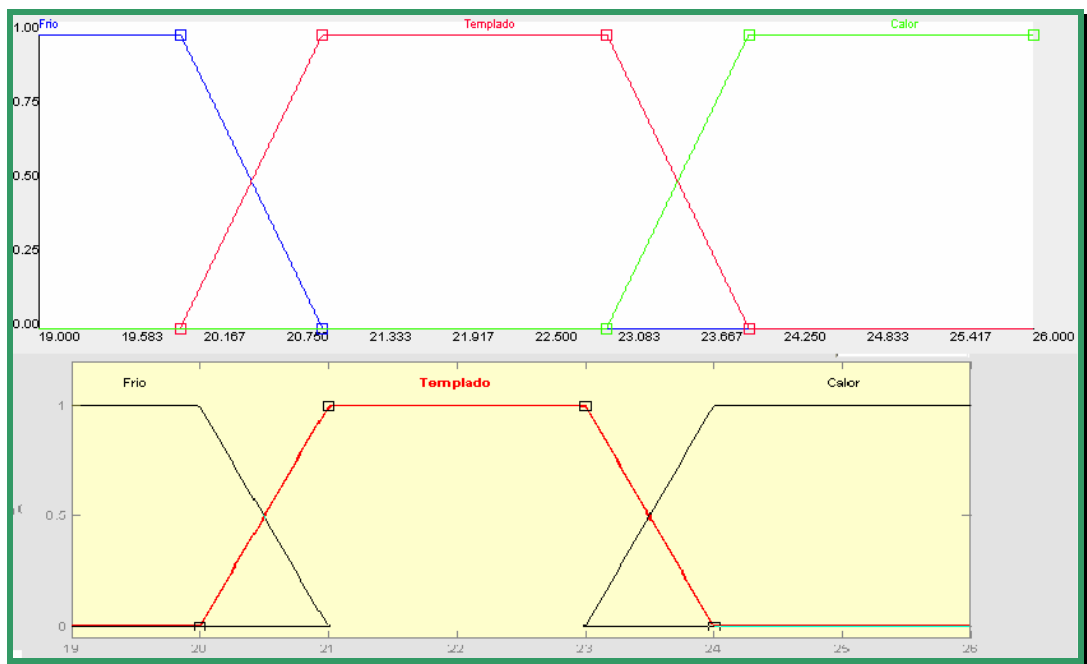


Figura 51. Temperatura: arriba Prototipo vs abajo Matlab

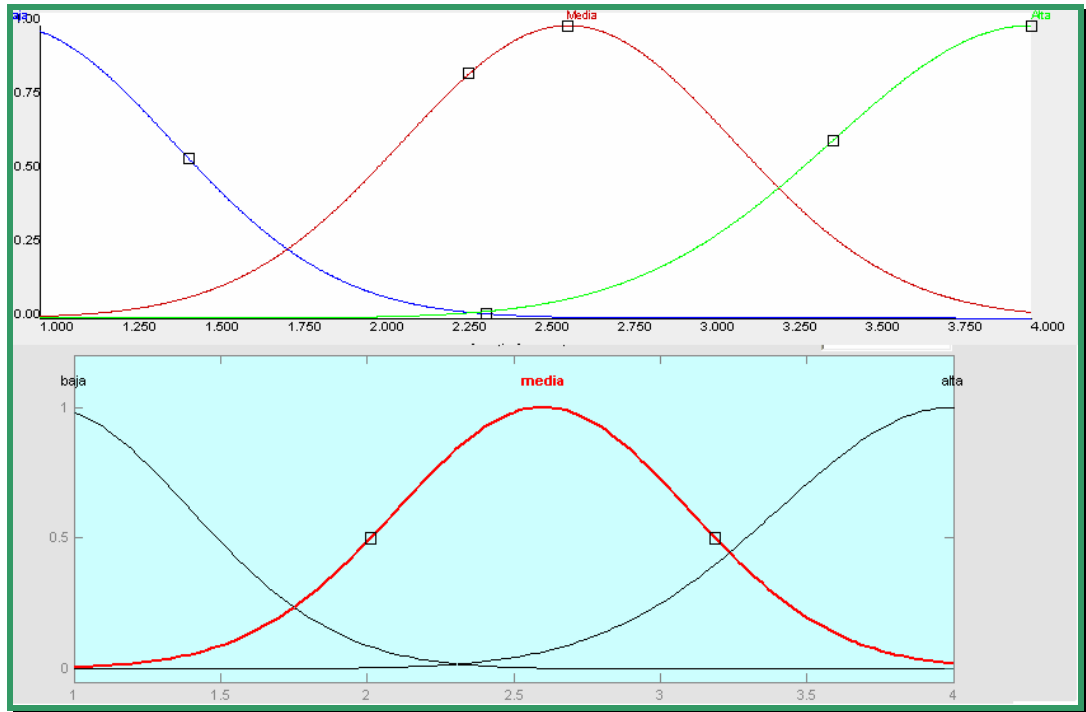


Figura 52. CO: arriba Prototipo vs abajo Matlab

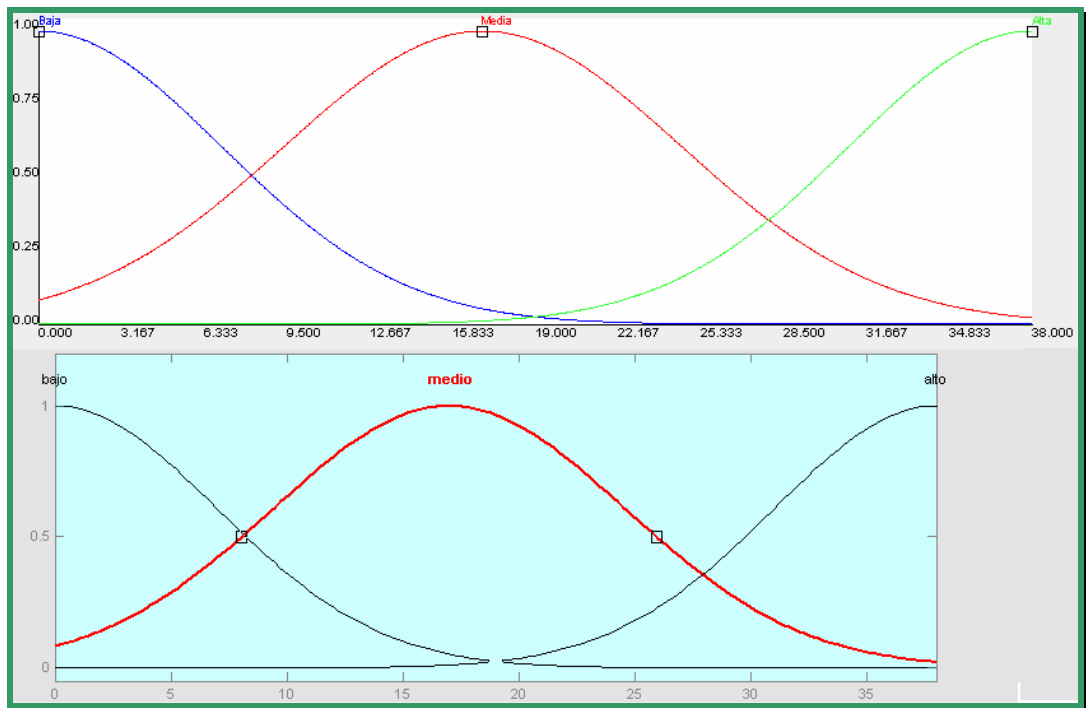


Figura 53. O<sub>3</sub>: arriba Prototipo vs abajo Matlab

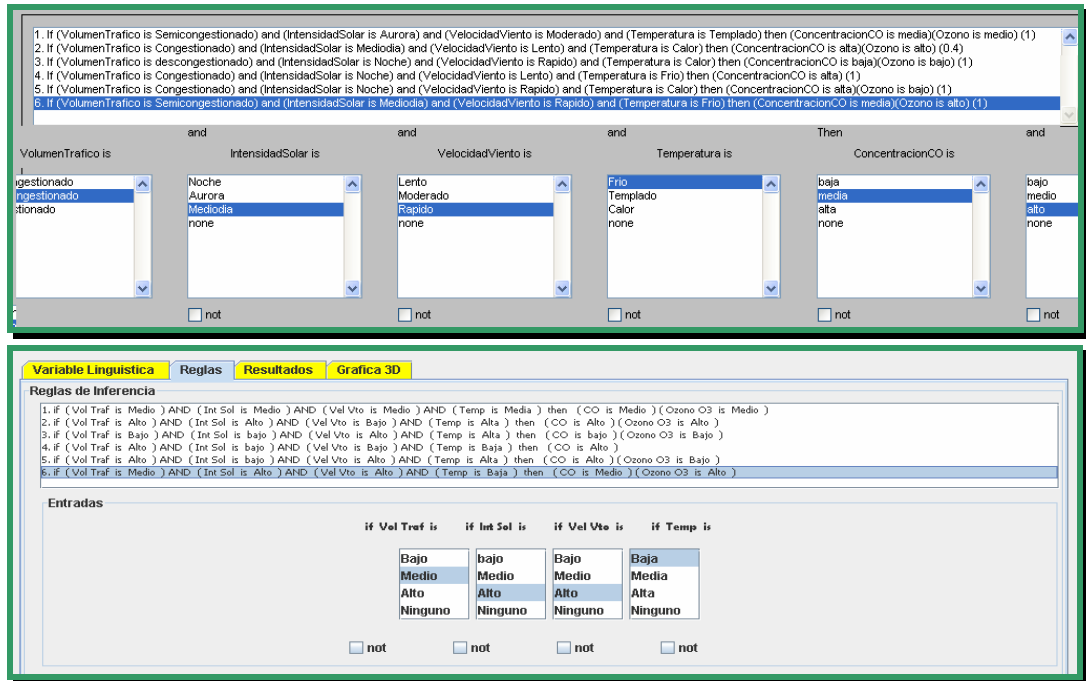


Figura 54. Reglas: arriba Prototipo vs abajo Matlab

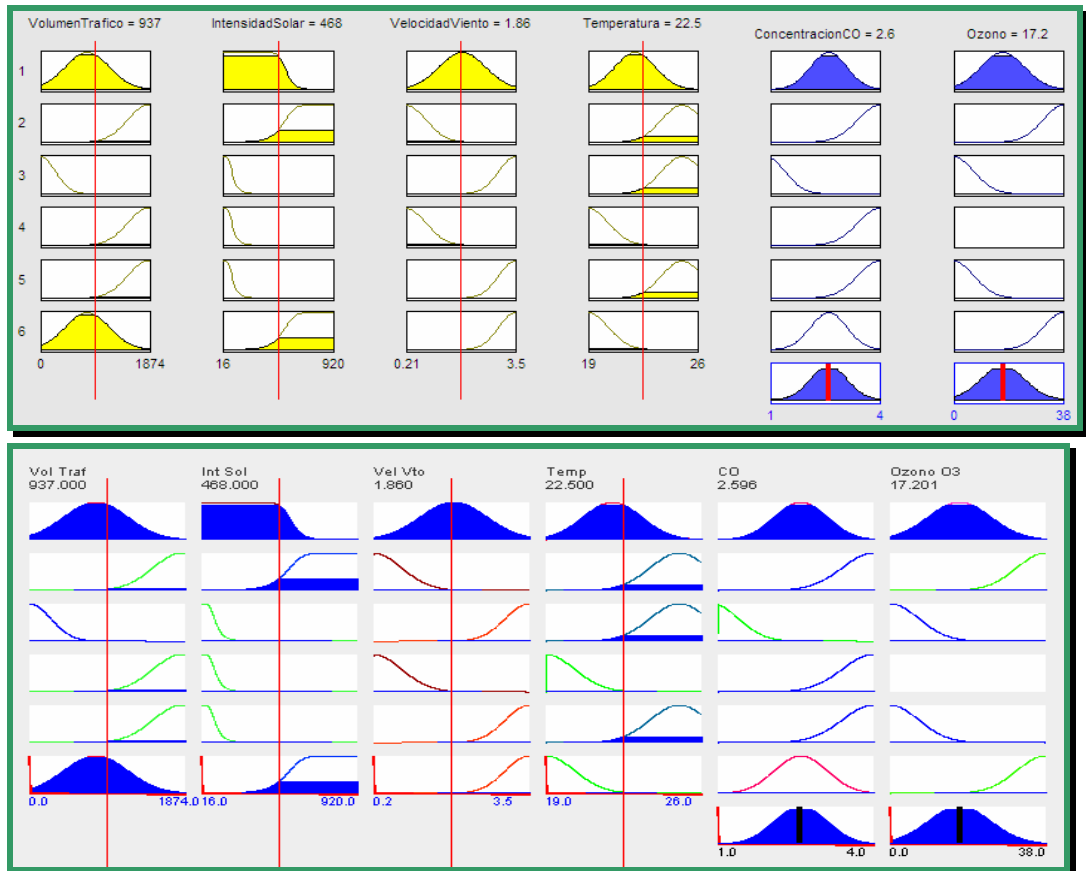
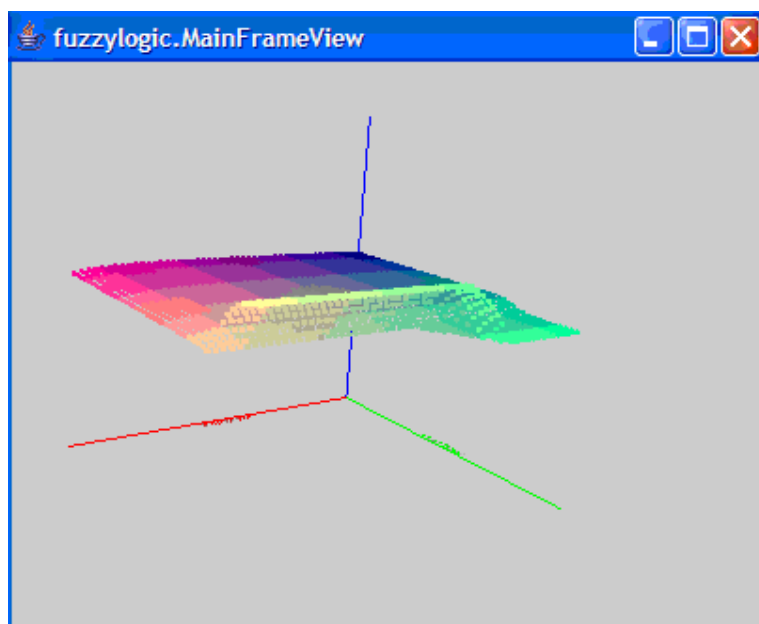
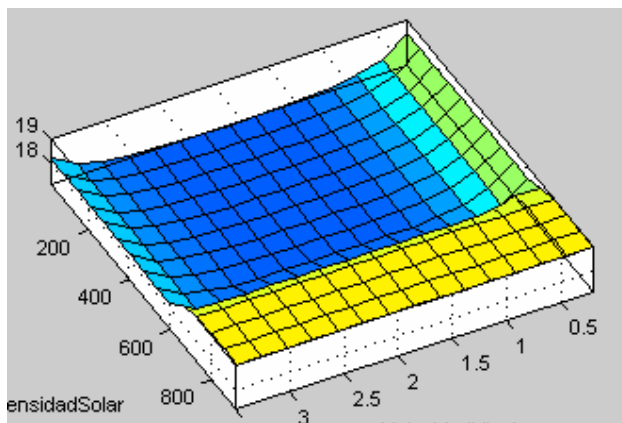


Figura 55. Resultados: arriba Prototipo vs abajo Matlab



**Figura 56.** Gráfico 3D: arriba Matlab vs abajo Prototipo

La mejor forma de verificar la veracidad de los resultados es comparándolo con uno de los software mas utilizados para crear modelos de lógica difusa que en nuestro caso es lo que estamos trabajando. Para ello presentamos la tabla de resultados:

<b>Variable de salida</b>	<b>Prototipo</b>	<b>Matlab</b>
CO	2.596	2.6
O <sub>3</sub>	17.201	17

**Tabla 6:** Sinopsis de resultados

De estas pruebas podemos observar que el prototipo presenta una gran similitud con el Toolbox Fuzzy Logic de Matlab tanto en su utilización como en la forma de manejar las diferentes ventanas.

Fue desarrollado en una metodología orientada da objetos lo cual facilita su futura modificación para mejoras.

Por último comparemos como quedaría distribuido en el escritorio de nuestro PC un proyecto de Toolbox Fuzzy Logic de Matlab y otro en el Prototipo Fuzzy Logic:

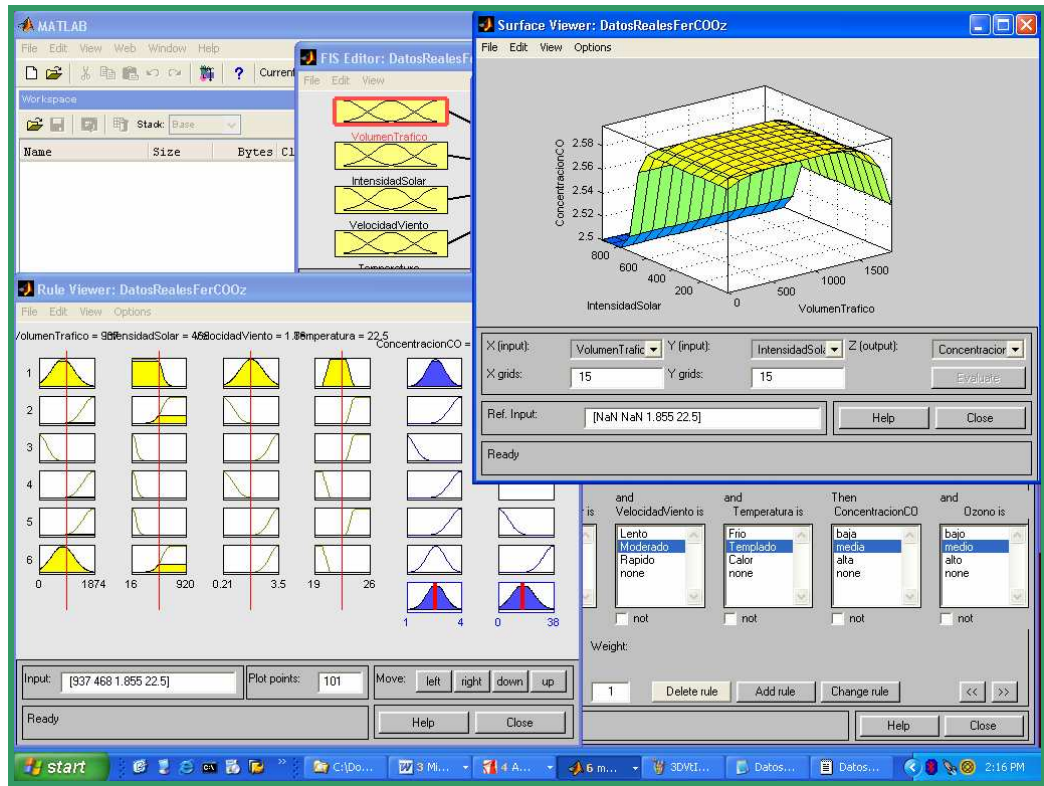
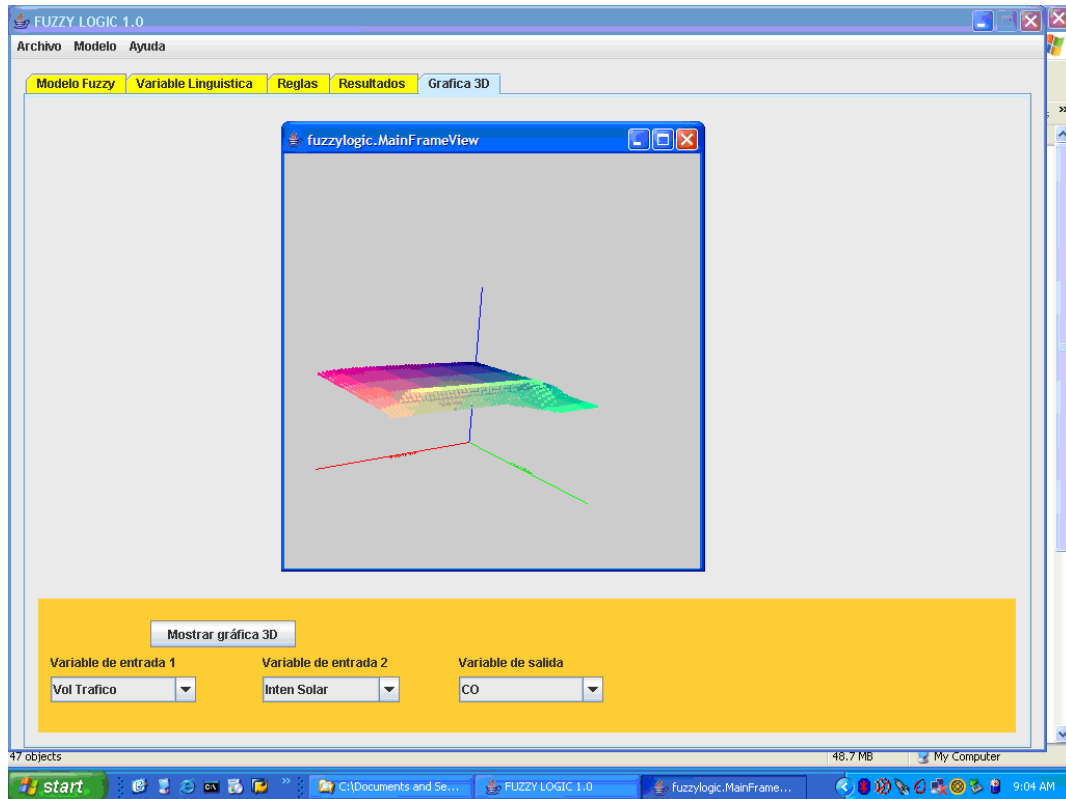


Figura 57. Proyecto distribuido en el escritorio (Toolbox Fuzzy Logic de Matlab)



**Figura 58.** Proyecto distribuido en el escritorio (Prototipo Fuzzy Logic)

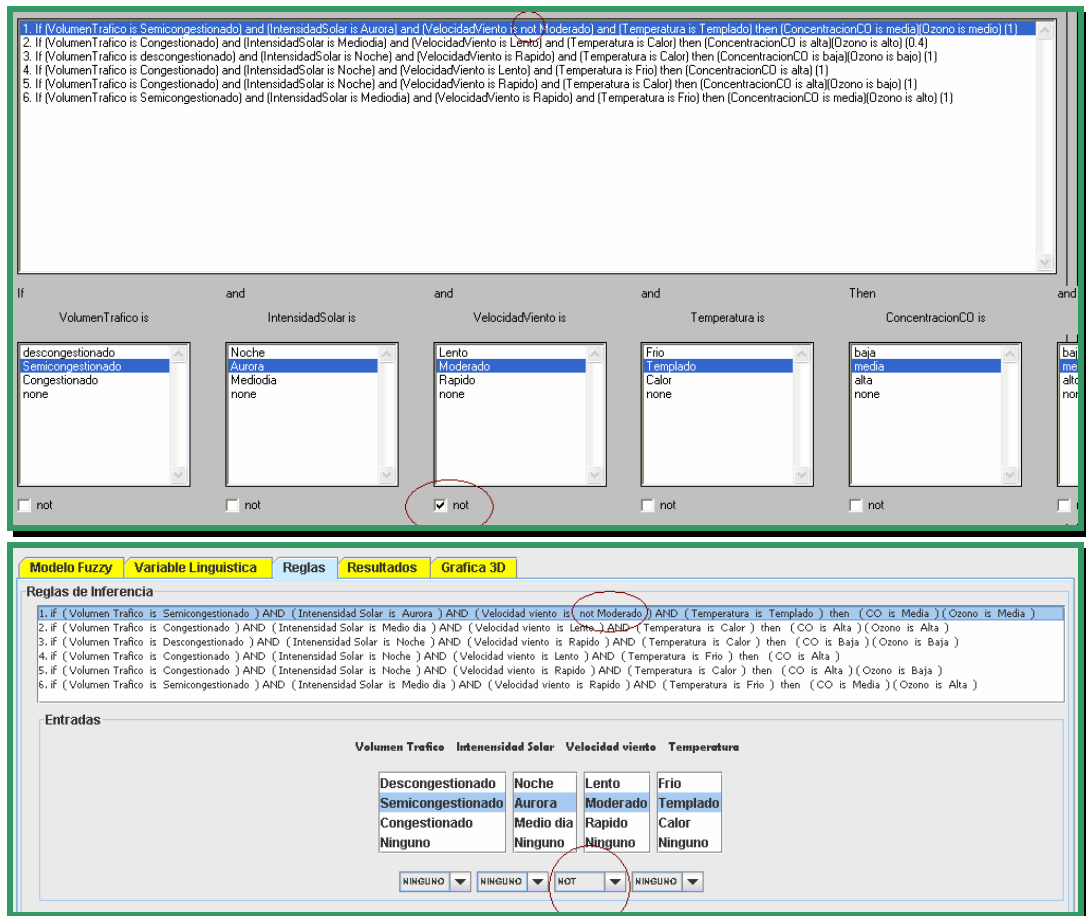
## 5.5. EJECUCIÓN DE LOS MODIFICADORES.

Esta prueba se realizó aparte porque, consiste en modificar el modelo anteriormente descrito, ya creado y los modificadores de dilatación y concentración no están definidos en el Toolbox Fuzzy Logic de Matlab. El prototipo solo cambia a partir del momento en que aplicamos los modificadores por ello solo mostraremos la parte que cambia en la sección 5.3 y 5.4 se ilustró el modelo.

Escogimos una regla de alto grado de activación, guiándonos de los gráficos mostrados en el Tab de resultados ya que para estas reglas el cambio realizado con los modificadores si surtiría efecto en los resultados y además será un poco notorio en los gráficos.

### 5.5.1. Modificador not

Para realizar una prueba con este modificador, perturbamos la variable lingüística Velocidad del Viento, más específicamente, en la primera regla que se creó en el prototipo anterior modificamos uno de sus conjuntos borrosos: el conjunto borroso Moderado lo cambiamos a Not Moderado, esto se hace aplicando este modificador en el Tab Reglas a la primera regla (**figura 59**).



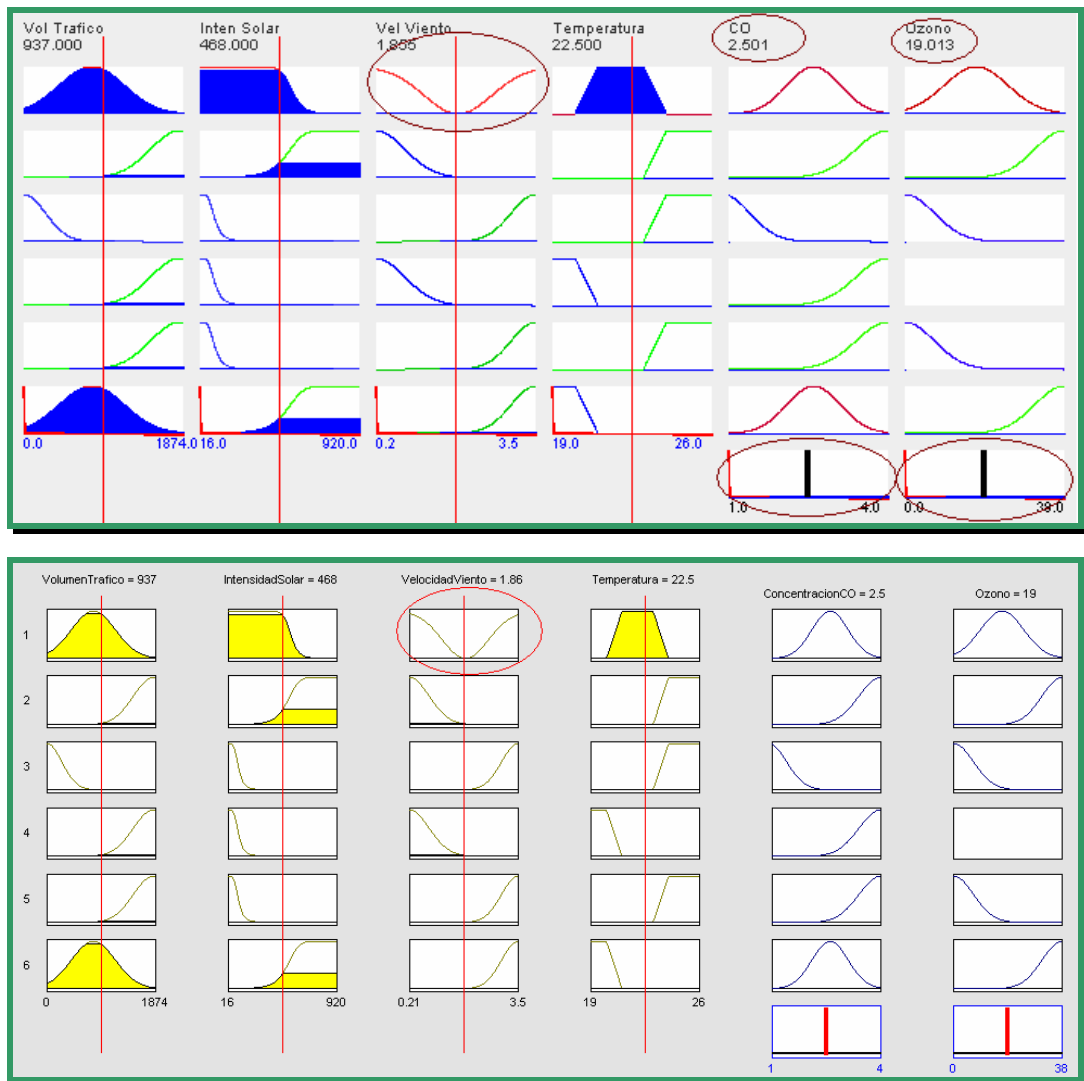
**Figura 59.** Modificador not en Tab Reglas

Este modificador crea un cambio en la gráfica y también en los valores de salida aunque en sus decimales así: el CO cambia de 2.596 a 2.501 y de el O<sub>3</sub> cambia de 17.201 a 19.013 (**figura 60**).

En el caso de la variable lingüística de salida  $O_3$  el cambio es un poco mas notoria que el CO, ya que cambia de en dos unidades. Presentamos la tabla de resultados:

Variable de salida	Prototipo	Matlab	Prototipo con modificador NOT	Matlab con modificador NOT
CO	2.596	2.6	2.501	2.5
$O_3$	17.201	17	19.013	19

**Tabla 7:** Sinopsis de resultados NOT



**Figura 60.** Modificador not en Tab Resultados

### 5.5.2. Modificador dilatación

Implementando una prueba con este modificador perturbamos la variable lingüística Intensidad solar, específicamente, en la primera regla que se creo en el prototipo anterior modificamos unos de sus conjuntos borrosos: el conjunto borroso Aurora lo cambiamos a AMPLIFI Aurora, que es la notación que utilizamos para dilatación, este modificador lo aplicamos en el Tab Reglas a la primera regla, (figura 61).

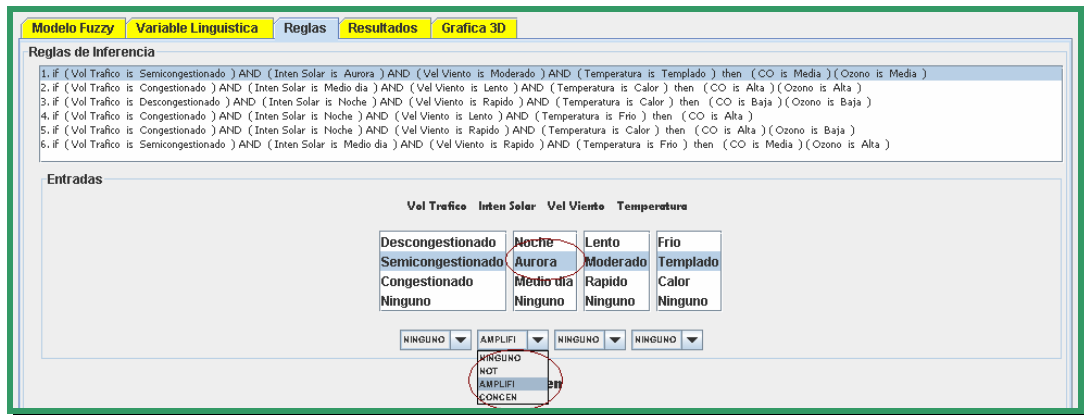


Figura 61. Modificador dilatación en Tab Reglas

Este modificador Dilatación, en este caso, implanta un pequeño cambio en la gráfica y también en los valores de salida aun mas pequeño que el modificador NOT en la sección anterior (sección 5.51) y también en sus decimales así: el CO cambia de 2.596 a 2.598 y de el O<sub>3</sub> cambia de 17.201 a 17.200 (figura 62).

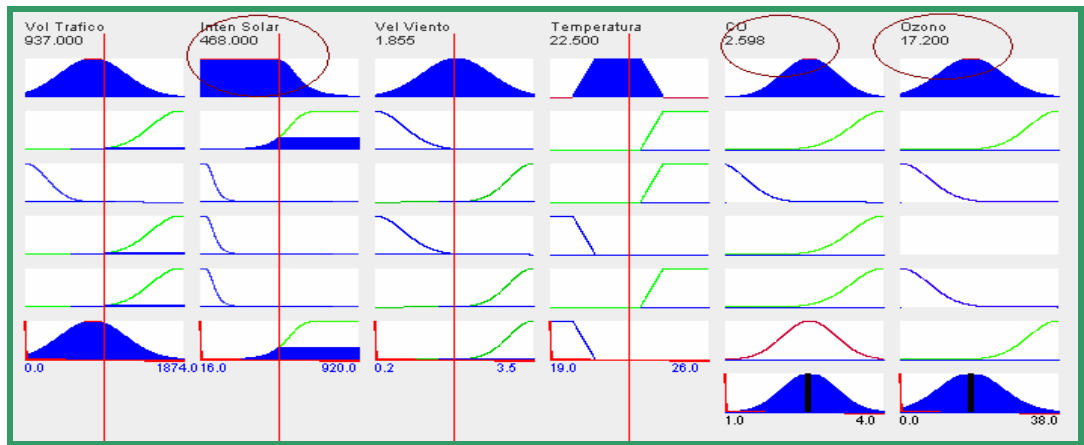


Figura 62. Modificador dilatación en Tab Resultados

### 5.5.3. Modificador concentración

Efectuando una prueba con este modificador alteramos la variable lingüística Intensidad solar, específicamente, en la primera regla que se creó en el prototipo anterior modificamos unos de sus conjuntos borrosos: el conjunto borroso Aurora lo cambiamos a CONCEN Aurora, que es la notación que utilizamos para el modificador concentración, este modificador lo aplicamos en el Tab Reglas a la primera regla, (figura 63).

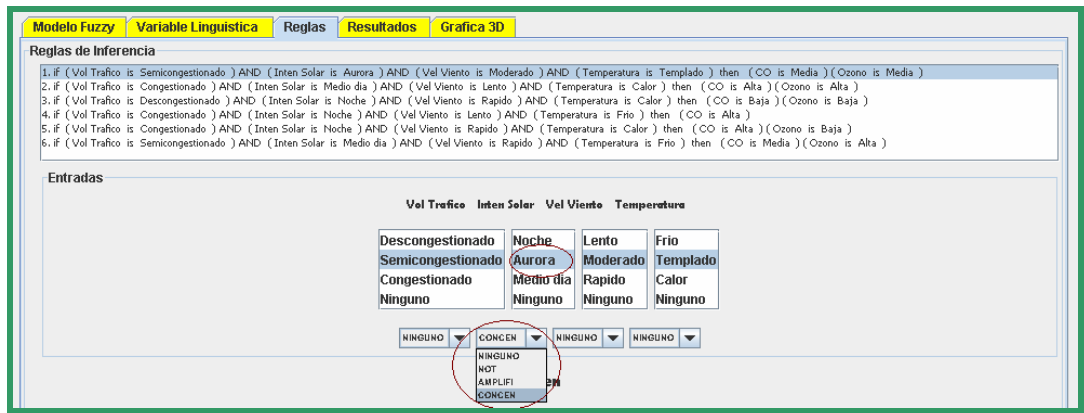
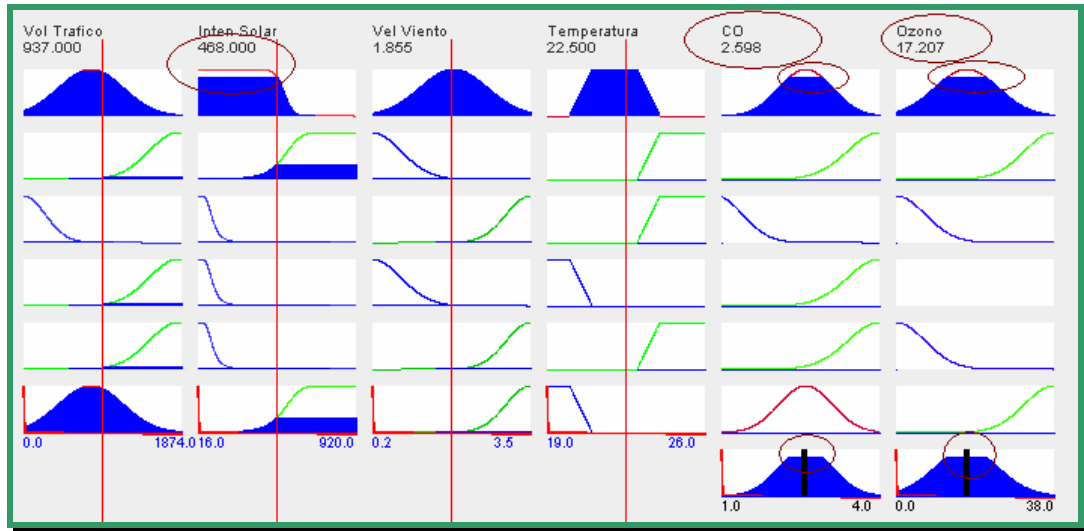


Figura 63. Modificador concentración en Tab Reglas

Este modificador Concentración, en este caso, implanta un pequeño cambio en la gráfica y también en los valores de salida aun mas pequeño que el modificador NOT en la sección anterior (sección 5.51) y también en sus decimales así: el CO cambia de 2.596 a 2.598 y de el O<sub>3</sub> cambia de 17.201 a 17.207 (figura 64).

La transformación con estos modificadores no siempre arrojan un cambio notable o robusto mas bien depende del rango con que se manejan las variables lingüísticas de entrada y salida tanto como de sus particiones borrosas; porque al dilatar o concentrar una función en estos conjuntos borrosos el cambio es poco en razón de elevación o declinación de gráfica en los puntos aplicados, además si en tal caso la regla del conjunto borroso modificado no es una que activa una regla, entonces el resultado no se vera afectado por que no estará en el calculo final de la salida.



**Figura 64.** Modificador concentración en Tab Resultados

Se deduce que en el Toolbox Fuzzy Logic de Matlab no se presentaría este cambio o sería un poco menos notorio, en cuanto a los resultados numéricos (pero si en las gráficas a igual escala que en el prototipo desarrollado y bajo las condiciones, entorno y casos nombrados en el párrafo anterior), ya que las salidas manejadas por este software solo utilizan una decimal.

## 6. CONCLUSIONES

- De la prueba realizada podemos concluir que el prototipo desarrollado muestra alta coherencia con los resultados esperados, con un alto grado de similitud en los gráficos mostrados para el resultado.
- Con este prototipo software nace una nueva herramienta para crear modelos de Lógica Difusa; 100% UIS.
- Puede ser utilizado para crear modelos donde se utilice la Lógica Difusa como la polución del aire y otros donde se pretenda aplicar esta metodología.
- Dada la metodología sirve como medio de consulta y apoyo a la toma de decisiones y control borroso.
- Con los tabs o pestañas, se evita tener varias ventanas abiertas en el escritorio, situación que puede provocar confusión a la hora de acceder a las múltiples tareas que se desarrollan dentro de la aplicación.
- Los diagramas de UML son de gran ayuda como diseño y guía para construir el software desarrollado.
- No en todos los casos al cambiar algunos valores de entrada, para las variables lingüísticas de entrada o de salida, o sus respectivos conjuntos borrosos, los resultados cambian notoriamente.

- Para poder construir un modelo muy parecido a la realidad se requiere de un experto del tema con la disposición de realizar un trabajo tan coherente.
- Tanto el prototipo desarrollado como el modelo implementado pueden ser modificados según criterios de los expertos
- Como el prototipo puede ser creado, configurado y después de ser analizado, guardado, esto satisface los diferentes puntos de vista que podrían tener los usuarios y sobre todo aquellos experto que deseen utilizar este prototipo.
- Utilizamos un espacio tridimensional, estos cálculos requieren de una gran carga de proceso
- Dado que está desarrollado en un lenguaje multiplataforma puede ser utilizado por una gran gama de usuario según sus preferencias esto maximiza su utilización.
- Este prototipo satisface diferentes gustos ya que se trata de un lenguaje multiplataforma y de Lógica Borrosa y también satisface muchos puntos de vista que podrían tener los usuarios tanto especialistas como inexpertos.
- Vemos que la polución del aire es tema que nos incumbe a todos y las autoridades pertinentes realizan un proceso de monitoreo pero está en nosotros evitar y contrarrestar uno de los factores que irradia y cataliza esta grave consecuencia.

## 7. RECOMENDACIONES

- Puede ser utilizado para las diferentes asignaturas y aplicaciones que se están desarrollando que utilizan la Lógica Borrosa para crear modelos.
- Este prototipo nos muestra el comportamiento de diferentes factores que intervienen en la contaminación ambiental y puede ser utilizado por los diferentes organismos pertinentes a la conservación ambiental.
- Para una posible y futura utilización de este prototipo se recomienda unas entradas lógicas y reales que obtenga unas salidas también lógicas muy parecidas a la realidad.
- Se trata de un prototipo donde se ilustra la forma o un modelo para la contaminación del aire con variables tomadas de diferentes trabajos de investigación y tesis de grado y de los diferentes organismos referentes a la contaminación de l aire donde cave la posibilidad de mejorarlo.
- Hoy en día es posible la simulación mediante cálculos basados en la proyección de entornos tridimensionales sobre pantallas bidimensionales, podemos ensayando una gama de colores, mejorar el espacio tridimensional.
- Si queremos realizar cambios o mejoras aprovechando el modelo recomendamos la vinculación de un experto en el tema para lograr resultados o salidas coherentes parecidas cercanas a la realidad.
- Con este proyecto damos a conocer una herramienta para trabajar con incertidumbre o datos incompletos y hacer conclusiones que nos pueden llevar a una conclusión muy parecida a una real y muy efectiva parecida a la concluida por un proceso lógico de razonamiento humano.
- Además demostramos que no solo es útil para procesos mecánicos o referentes solo a sistemas sino también a aspectos tan diversos como la

polución del aire por ello invitamos a robustecer su utilización en otros temas.

- La designación de java como lenguaje de programación de este prototipo nos crea la idea y un buen concepto de trabajar con software libre y de multiplataforma.
- La recuperación y conservación del medio ambiente es compromiso de todos

## 7. BIBLIOGRAFÍA

1. AJA, F. Santiago. Tesis de doctorado. Un marco matricial para la implementación de la inferencia borrosa aplicados al procesado de información no numérica. Universidad de Valladolid. España. 2003.
2. AMADO, J. P. CASTRO, J. C. y otros (2004). Trabajo de grado. Estudio Comparativo de Contaminación atmosférica por la operación de un S.I.T.M en Bucaramanga. UIS. Bucaramanga
3. BRAUDE, Eric J. Ingeniería del software. Una perspectiva orientada a objetos. Alfaomega. México, 2003.
4. CDMB: Corporación Autónoma para la Defensa de la Meseta de Bucaramanga. <http://www.cdm.gov.co/conozca/paginas/ibuca.htm>
5. CHAPA, S. "Arquitectura de Sistemas Expertos".  
URL: [http://delta.cs.cinvestav.mx/~schapa/red/intro\\_lm/node46.html](http://delta.cs.cinvestav.mx/~schapa/red/intro_lm/node46.html)
6. CORREDOR Montagut, Martha Vitalia. Principios de Inteligencia Artificial y Sistemas Expertos. Primera Edición. Ediciones UIS. Bucaramanga. 2000. Pags. 59-86.
7. EPA: Environmental Protection Agency.  
URL: <http://www.epa.gov/>
8. ETSIIT. "Inteligencia Artificial",  
URL: <http://etsiit.ugr.es/alumnos/mlii/inteligencia.html>
9. GRECH M., Pablo. Introducción a la Ingeniería, Un enfoque a través del Diseño, Primera Edición. Pearson Education, Colombia, 2001.
10. HILERA, José R., MARTÍNEZ Víctor J. Redes neuronales artificiales, fundamentos y aplicaciones. Alfaomega 2000. Capítulo 9.
11. <http://www.willydev.net/descargas/Articulos/General/umlTotal.pdf>  
Esta página de Internet, presenta documentación acerca de UML
12. JAIMES, Marcela. Propuesta de investigación. Modelado y simulación de la contaminación del aire atmosférico producido por fuentes móviles del área metropolitana de Bucaramanga. UIS. Bucaramanga.

13. JAIMES, M. CASTILLO, E. F. Cachón, J. L. Artículo. Modelo difuso de la dispersión del aire en Bucaramanga. UIS. Bucaramanga.
14. KLIR, George J., YUAN, Bo. Fuzzy Sets and Fuzzy Logic – Theory and Applications. Prentice Hall, 1995.
15. MENDEL, CIDS: Centro de investigación en desarrollo de sistemas. “Sistemas expertos basados en reglas”.  
URL: [http://ci2a.upeu.edu.pe/topicos/topico\\_sistbareglas.htm](http://ci2a.upeu.edu.pe/topicos/topico_sistbareglas.htm)
16. MENDEL, Jelly M. Uncertain. Rule – Based Fuzzy Logic System. Introduction and New Directions. Prentice Hall PTR, 2001.
17. McCONNELL, Steve. Desarrollo y gestión de proyectos informáticos. Cómo dominar planificaciones ajustadas de software. McGraw Hill, España, 1997.
18. NILSSON, Nils J. Inteligencia artificial. Una nueva síntesis. McGraw Hill, España, 2001.
19. PRESSMAN, Roger. Ingeniería del Software – Un enfoque práctico, Quinta Edición. McGraw Hill, España, 2002.
20. QUIROGA, L. Y. TÉLLEZ, Z. C. Trabajo de grado. Valoración estadística y comparativa del cálculo del Índice de la calidad del aire. UIS. Bucaramanga. (2005).
21. ROLSTON, David W. Principios de Inteligencia Artificial y Sistemas Expertos. McGraw Hill, Colombia, 1990.
22. ROMERO A, VARGAS P. Trabajo de grado. Organización, distribución en planta, planificación de montaje y estudio económico del centro de diagnóstico automotor para la dirección de tránsito de Bucaramanga.
23. SAMPER, J. “Sistemas Expertos: El conocimiento al poder”.  
Psicología.com URL: [http://www.psicologia.com/articulos/ar-jsamper01\\_2.htm](http://www.psicologia.com/articulos/ar-jsamper01_2.htm)
24. WIKIPEDIA. “Inteligencia Artificial”.  
URL: [http://es.wikipedia.org/wiki/Inteligencia\\_artificial](http://es.wikipedia.org/wiki/Inteligencia_artificial)
25. WU, Thomas C. Introducción a la programación orientada a objetos con Java, Segunda Edición. McGraw Hill, España, 2001.

## ANEXOS

### ANEXO A: MANUAL DE AYUDA

Variables de entrada.

Tab Modelo Fuzzy.

Figura Modelo Fuzzy.

Tab Variable Lingüística.

Conjuntos Borrosos.

Función de Pertenencia.

Figura Tab Variable Lingüística.

Tab de Reglas.

Figura Tab de Reglas.

Tab de Resultados.

Figura Tab de Resultados.

Tab Gráfica 3D.

Figura Tab Gráfica 3D.

Ventana Gráfica 3D.

En este manual se resume como realizar un modelo paso a paso mediante el prototipo de simulación.

Antes de empezar hay que tener en cuenta si el equipo en que se realizará el modelo es apto y cumple con las mínimas especificaciones.

Primero que todo se instala la maquina virtual de java que es indispensable para cualquier aplicación de java y también una librería que se utiliza para la gráfica en 3D que se llama Java3D. Estos programas pueden ser descargados desde la página de Sun Microsystems.

También hay que tener en cuenta que la persona que esté realizando el ejemplo del modelo debe ser una persona que conozca sobre Lógica Difusa y sobre contaminación ambiental, porque los datos que se ingresen deben ser reales para poder obtener resultados óptimos.

Ahora se procede a realizar un ejemplo paso a paso para que no haya ninguna duda a la hora de realizar el ejemplo del modelo.

1. Variables de entrada. Al comienzo hay que tener claro cuales son las variables de entrada y las de salida. Para tener buenos resultados se recomienda que máximo se deba utilizar o trabajar con cuatro variables de entrada y dos de salida. Esto se recomienda para obtener un mejor resultado del modelo.
2. El prototipo muestra una interface lo más amigable para el usuario, se hizo de modo que el usuario pueda a simple vista ver un orden de proceder, se refiere a los Tabs o pestañas que aparecen debajo de los menús. El orden a seguir será de izquierda a derecha.
3. Tab Modelo Fuzzy. El primer tab se llama Modelo Fuzzy, este es una tab de información, donde se muestra un grafico representado por rectángulos donde la primera columna (rectángulo de color amarillo) representa las variables de entrada, el rectángulo del centro (rectángulo de color gris) representa el tipo de modelo que se esta trabajando, en este caso es de tipo Mandani, y la última columna de rectángulos (rectángulos de color azul) representa las variables de salida. Y debajo de este grafico presenta más información sobre modelo que se va a trabajar (**figura A1**).



**Figura A1.** Tab Modelo Fuzzy

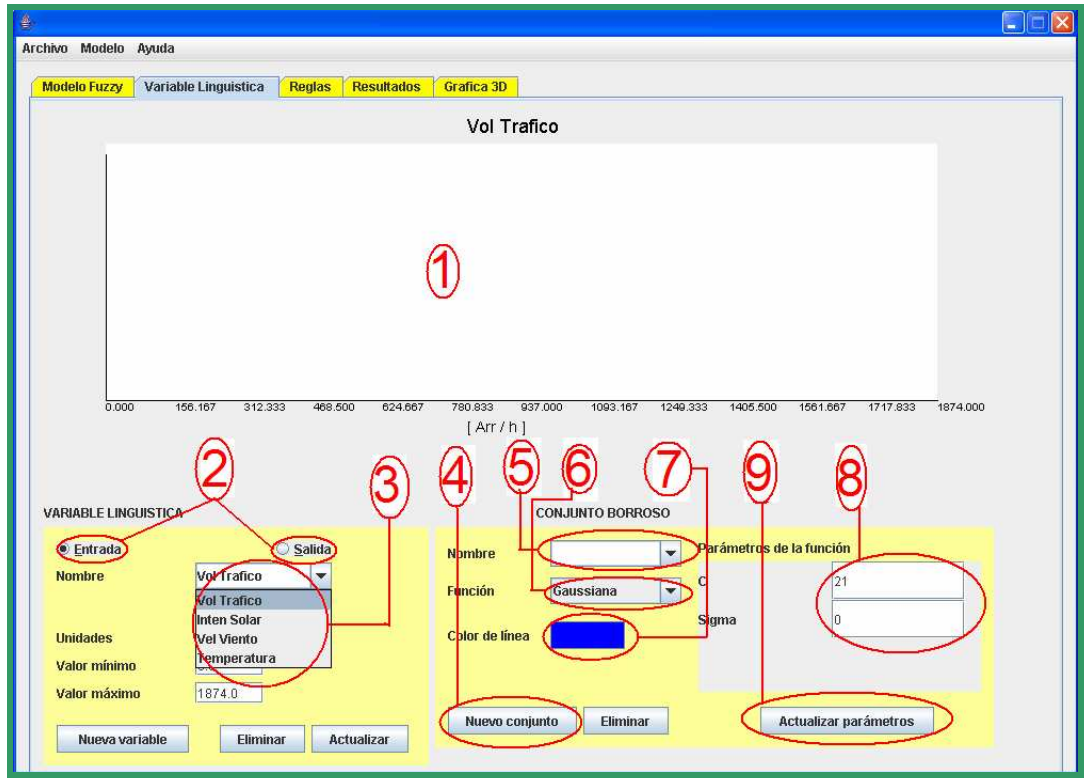
4. Tab Variable Lingüística. El segundo tab se llama Variable Lingüística, este se puede considerar el más importante porque es donde se incluirá toda la información de entrada. Este prototipo presenta por defecto ya unas variables de entrada y salida que tienen que ver con la contaminación ambiental, primero se selecciona el check box de entradas y cuando termina con estas, se selecciona el check box de salida, solo hay que seleccionar la variable que se quiere trabajar y se procede a ingresar los conjuntos borrosos. Si el usuario del prototipo quiere trabajar con una variable de entrada y no se encuentra en la lista, deberá pulsar el botón de nueva variable e inmediatamente digitar el nombre de la nueva variable, las unidades y el rango respectivo y después debe pulsar el botón de actualizar para guardar la información digitada. Si por el contrario el usuario no desea trabajar con alguna de las variables que aparecen por defecto, deberá seleccionar la variable y pulsar el botón de eliminar. En la parte superior aparece un plano

cartesiano, que es donde aparecerá todos los conjuntos borrosos que se crearan, se recomienda que máximo debe crear tres conjuntos borrosos.

5. Conjuntos Borrosos. Para ingresar los conjuntos borrosos se debe pulsar el botón de nuevo conjunto y luego se procede a ingresar los datos de los mismos. Dependiendo del tipo de variable de entrada se ingresaran los conjuntos borrosos, se refiere a que los nombres que se les de a los conjuntos borrosos deben ser coherentes con la variable lingüística. Por ejemplo, los conjuntos borrosos para la variable de entrada volumen del tráfico son: Descongestionado, Semidescongestionado y Congestionado.
6. Función de Pertenencia. El tipo de función de pertenencia a escoger ya depende de la variable de entrada o de la opinión del experto, esto ya queda a criterio del usuario. Este prototipo tiene por defecto cinco tipos de función de pertenencia: Gaussiana, Campana de Bell, Sigmoidal, trapezoidal y triangular. La posición en el plano cartesiano de cada conjunto borroso se sitúa de acuerdo al rango y al tipo de conjunto borroso.
7. la opción del color se refiere al color que llevará la gráfica de la función de pertenencia en el plano cartesiano, para poder diferenciarla de las otras gráficas. A continuación se presenta una imagen del tab de la variable lingüística y se hará una descripción rápida del orden que debe seguir el usuario para el correcto funcionamiento del prototipo.
  1. Es el espacio del plano cartesiano, será donde van todas las gráficas de función de pertenencia de los conjuntos borrosos para cada variable de entrada.
  2. Son los check box que nos hacen referencia al tipo de variable si es de entrada o salida. En la **figura A2** se muestra seleccionado

el check box de entrada y por defecto muestra los nombres de las variables de entrada.

3. Aparece la lista de las cuatro variables de entrada que están por defecto, si el usuario quiere una variable nueva puede presionar el botón de nueva variable y crearla y si lo que quiere es eliminar alguna de estas variables, solo selecciona la variable y presiona el botón de eliminar que se encuentra debajo y después actualizar para guardar la información.
4. Es el botón para empezar a crear los conjuntos borrosos, este se oprime después de haber seleccionado la variable lingüística de entrada.
5. Espacio para digitar el nombre del conjunto borroso, este nombre debe tener relación con la variable lingüística.
6. Aquí el usuario deberá escoger entre los cinco tipos de función de pertenencia (Gaussiana, Campana de Bell, Sigmoidal, Trapezoidal y Triangular) que aparecen por defecto, solo se trabajará con estas funciones, debido a que no hay otras funciones implementadas.
7. El usuario podrá seleccionar el color que quiera de una paleta de colores para la función de pertenencia de cada conjunto borroso.
8. El usuario digitará los valores correspondientes a cada función de pertenencia, aquí necesita la ayuda de un experto en el tema, para una correcta posición en el plano cartesiano.
9. El botón actualizar parámetros se presiona al final, cuando has terminado de digitar la información para cada conjunto borroso. Luego se procede a repetir los mismos pasos para ingresar otro conjunto borroso, solo hay que tener en cuenta que para una solución mas optima se recomienda que debe ingresar un número impar de conjuntos borrosos.

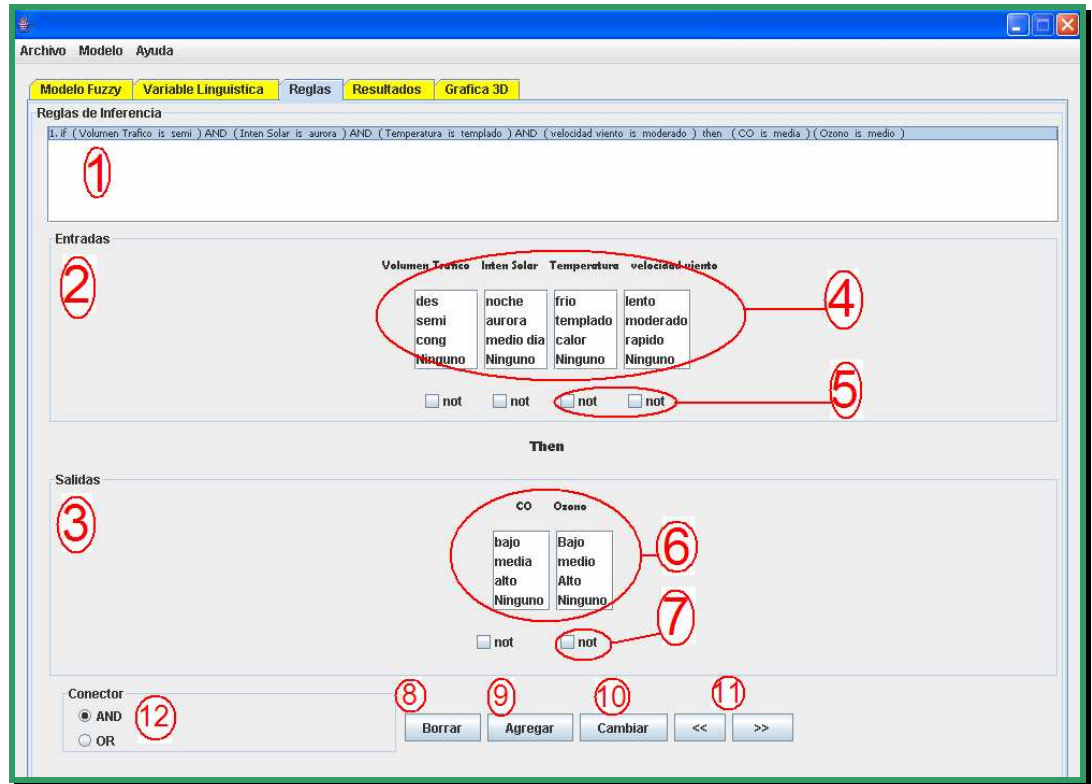


**Figura A2.** Tab de la variable lingüística.

8. Después de haber ingresado o digitado la información de las variables de entrada y salida pasamos al tab de las reglas, ahí se procede a registrar las reglas que con ayuda del experto se hace el análisis de las mismas. A se presenta una descripción rápida de lo que contiene el tab de las reglas.

1. Tab de Reglas. Es un campo de texto donde se estarán creando las reglas para ser vistas por el usuario, a medida que se crea una regla nueva se va pasando una tras de otra y se van numerando sucesivamente. No existe un límite para la cantidad de reglas a crear, pero para visualizarlas todas debemos movernos en este panel. **Figura A3.**
2. Es el panel donde se encuentran las variables de entrada.
3. Es el panel donde se encuentran las variables de salida.

4. Es el conjunto de todas las variables de entrada con sus respectivos conjuntos borrosos, además cada variable tiene la opción de “Ninguno” que es usada para cuando no se necesite esa variable para crear la regla. Para crear la primera regla solo se tiene que seleccionar el conjunto borroso de cada variable y el prototipo automáticamente le va creando la regla en la parte superior donde se encuentra el espacio de texto. Y para crear la segunda, tercera, y así sucesivamente las demás reglas solo se oprime el botón de agregar.
5. “not” es un check box que se selecciona para negar el conjunto borroso.
6. Es el conjunto de todas las variables de salida con sus respectivos conjuntos borrosos, también tiene la opción de “Ninguno”.
7. “not” es un check box que se selecciona para negar el conjunto borroso.
8. Botón borrar, sirve para eliminar una regla, la que se este seleccionando actualmente.
9. Botón agregar, para crear la primera regla no es necesario oprimir el botón agregar, solamente cuando el usuario quiere mas de una regla se oprime agregar y el prototipo le creará una regla nueva con las especificaciones de la ultima regla creada, solo es cuestión de seleccionarla y cambiarle los parámetros o conjuntos borrosos de las variables lingüísticas.
10. Botón cambiar, este botón tiene la funcionalidad de guardar los cambios que se le hacen a una regla, primero se modifica la regla y luego se oprime el botón cambiar.
11. Son botones de desplazamiento hacia la derecha o hacia la izquierda, cuando el nombre de los conjuntos borrosos salen muy largos.
12. Son los dos tipos de conectores (AND y OR) que se usan para enlazar los conjuntos borrosos en una regla.

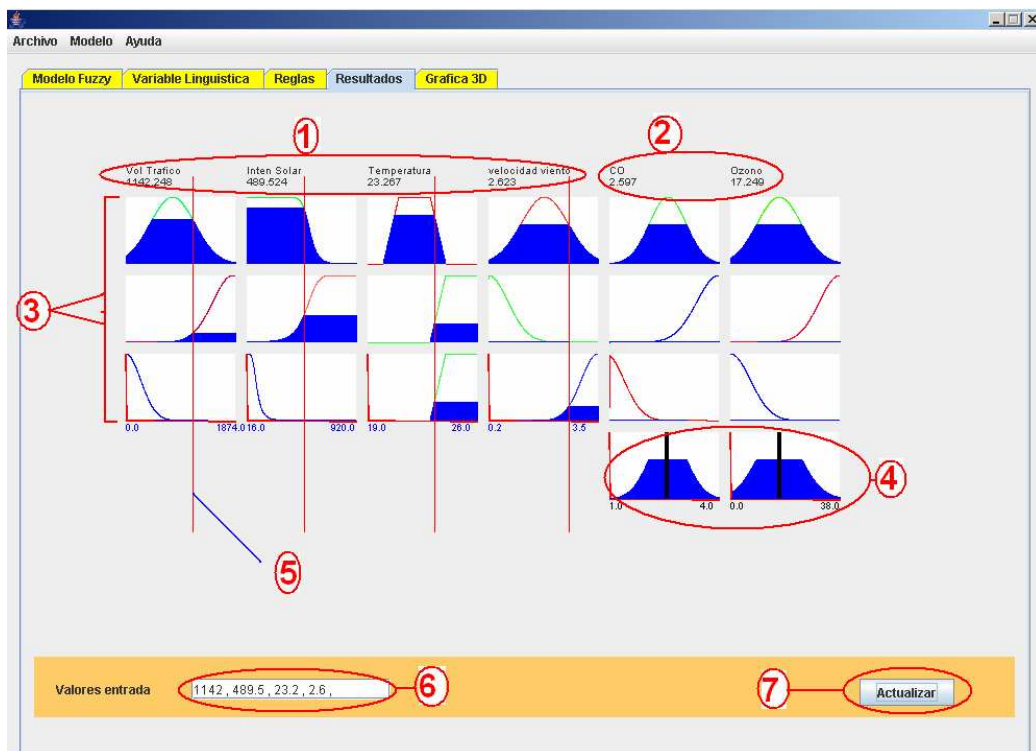


**Figura A3.** Tab de las reglas

9. Tab de Resultados. Luego de haber ingresado o digitado las reglas para el ejemplo, se pasa al tab de resultados, en este tab se ven representadas cada una de las reglas en forma gráfica. Cada conjunto borroso está representado con su respectiva función de pertenencia. A continuación en la **figura A4** se ve un ejemplo el cual se explica a continuación.

1. Son los nombres de las variables lingüísticas de entrada, además debajo aparece el valor en que esta actualmente esa variable, esos valores de cada variable los determina el usuario con la ayuda del experto.
2. Son las dos variables de salida que se usaron para el ejemplo, también tienen un valor cada una, pero ese valor si es calculado por el prototipo.
3. Cada fila de gráficas representa una regla, para este ejemplo se crearon tres reglas que se ven representadas en la **figura A4**.

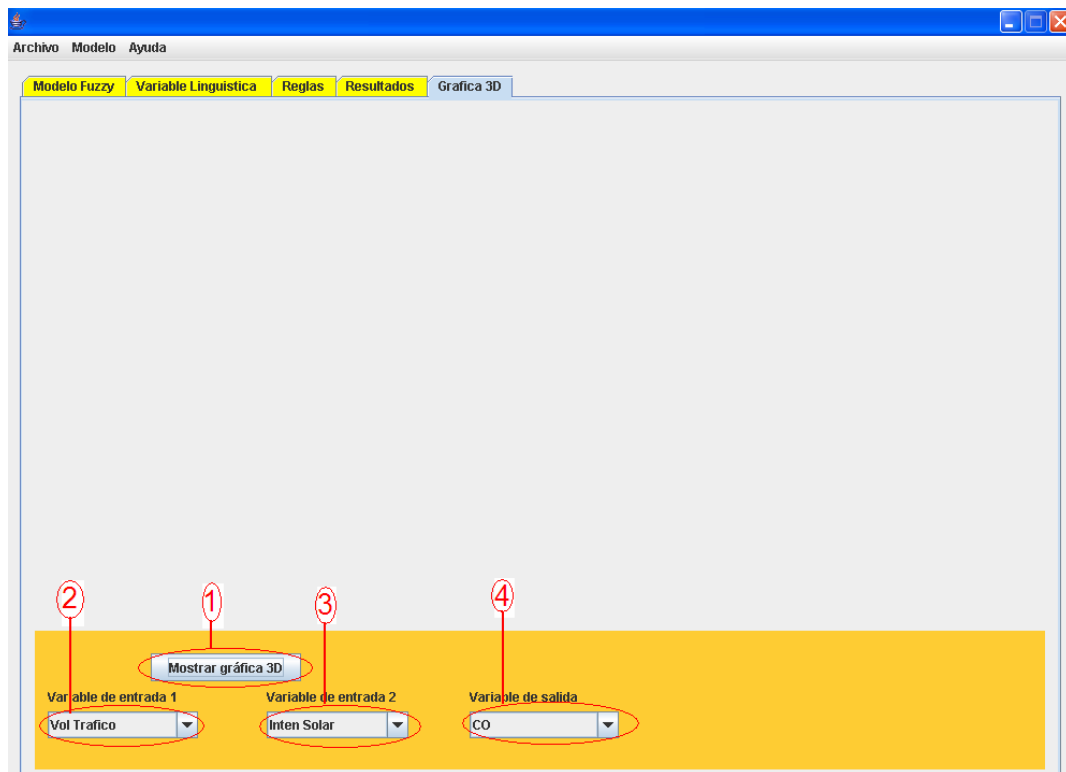
4. Estas gráficas representan la salida total, a las cuales se les aplicó el método de defusificación del centroide, el cual se ve representado mediante una barra negra que aparece en la mitad de cada gráfica.
5. Son barras de desplazamiento horizontal. Cada variable de entrada tiene una barra de desplazamiento horizontal, que solo se moverá en el rango que aparece en cada gráfica. El valor que representa esta barra en cada variable, es el que aparece en la parte superior, debajo del nombre de la variable.
6. Es un campo de entrada de datos. Ahí se introduce por teclado el valor de cada variable de entrada en caso de que no se quiera hacer por medio de las barras de desplazamiento.
7. Este botón se oprime una vez que haya introducido los datos por teclado en el campo de variables de entrada, el cual nos actualiza los datos en la parte superior y también la barra de desplazamiento de cada variable la sitúa en la posición correspondiente.



**Figura A4.** Tab de Resultados.

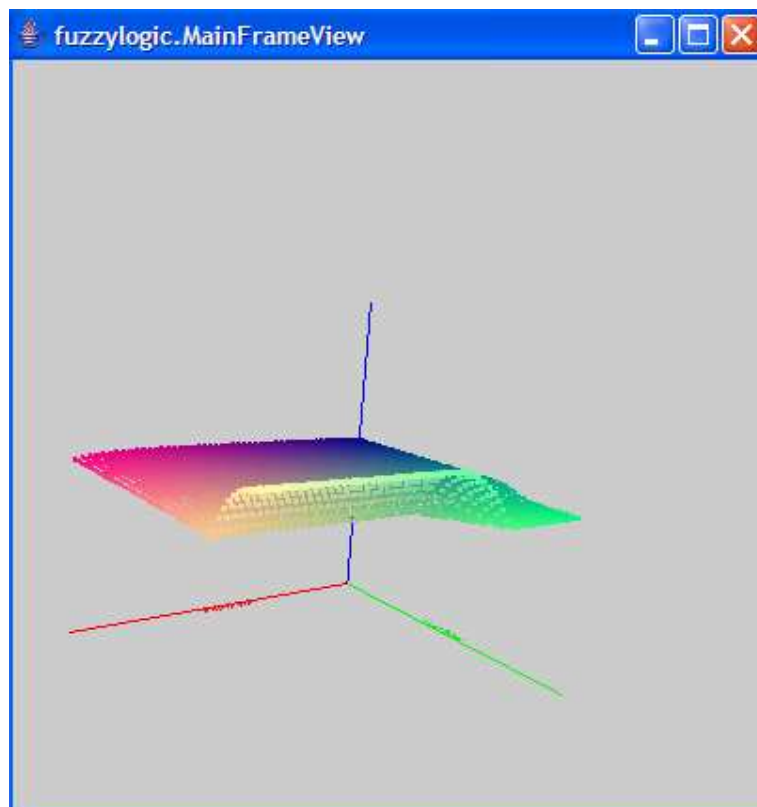
10. Tab Gráfica 3D. Por último se pasa al tab de gráfica 3D. en este tab solo se selecciona la variable que va en cada eje del plano tridimensional.

1. Botón mostrar Gráfica 3D, se oprime una vez ya haya escogido las variables a graficar.
2. Variable de entrada 1. Aparecen todas las variables de entrada a ser seleccionadas.
3. Variable de entrada 2. Aparecen todas las variables de entrada a ser seleccionadas.
4. Variable de salida. Aparecen todas las variables de salida a ser seleccionadas.



**Figura A5.** Tab de la gráfica 3D.

11. Una vez seleccionadas las variable se oprime el botón mostrar gráfica 3D. A continuación se muestra la gráfica en tres dimensiones.



**Figura A6.** Gráfica 3D.

1. Flecha arriba: zoom hacia delante
2. Flecha abajo: zoom hacia atrás
3. Flecha derecha: desplaza la gráfica hacia la izquierda
4. Flecha izquierda: desplaza la gráfica hacia la derecha
5. Re Pág: desplaza la gráfica hacia arriba
6. Av Pág: desplaza la gráfica hacia abajo
7. Alt Gr + flecha izquierda: rotación hacia la derecha
8. Alt Gr + flecha derecha: rotación hacia la izquierda
9. Alt Gr + Re Pág: rotación hacia el frente y hacia abajo
10. Alt Gr + Av Pág: rotación hacia arriba y hacia atrás