

DESARROLLO E IMPLEMENTACIÓN DE LA INTERFAZ GRÁFICA DEL
USUARIO Y SU INTEGRACIÓN CON EL PAQUETE “MOTOR DE INFERENCIA
DIFUSO” DEL PROTOTIPO SISTEMA EXPERTO PARA LA POLUCIÓN DEL
AIRE (SEXDP).

EDWIN ERNESTO ANAYA ALMEIDA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA 2007

DESARROLLO E IMPLEMENTACIÓN DE LA INTERFAZ GRÁFICA DEL
USUARIO Y SU INTEGRACIÓN CON EL PAQUETE “MOTOR DE INFERENCIA
DIFUSO” DEL PROTOTIPO SISTEMA EXPERTO PARA LA POLUCIÓN DEL
AIRE (SEXDP).

EDWIN ERNESTO ANAYA ALMEIDA

Trabajo de grado presentado como requisito para optar al título de
INGENIERO DE SISTEMAS

DIRECTOR
MSc. MANUEL GUILLERMO FLOREZ BECERRA

CODIRECTOR
DR. JORGE LUÍS CHACÓN VELASCO

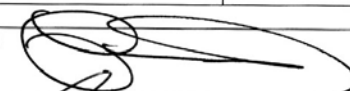
UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA 2007



NOTA DE PROYECTO DE GRADO

NOMBRE DEL ESTUDIANTE EDWIN ERNESTO ANAYA ALMEIDA		CODIGO 1981917
TITULO DEL PROYECTO 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 (SEXDP).		
REGISTRO No. 1483	FACULTAD: INGENIERÍAS FISICOMECAÑICAS	CARRERA: INGENIERÍA DE SISTEMAS
CALIFICACIÓN: APROBADO		CRÉDITOS ///
DIRECTOR DE PROYECTO:		
NOMBRE: MANUEL GUILLERMO FLÓREZ BECERRA		FIRMA: 
CALIFICADORES:		
 HENRY ARGÜELLO LEANTES	 JUAN CARLOS REYES FIGUEROA	FECHA AÑO MES DIA 2007 08 27

NOTA DE PROYECTO DE GRADO

NOMBRE DEL ESTUDIANTE EDWIN ERNESTO ANAYA ALMEIDA		CODIGO 1981917
TITULO DEL PROYECTO 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 (SEXDP).		
REGISTRO No. 1483	FACULTAD: INGENIERÍAS FISICOMECAÑICAS	CARRERA: INGENIERÍA DE SISTEMAS
CALIFICACIÓN: CUATRO.DOS (4.2)		CRÉDITOS 08
DIRECTOR DE PROYECTO:		
NOMBRE: MANUEL GUILLERMO FLÓREZ BECERRA		FIRMA: 
CALIFICADORES:		
 HENRY ARGÜELLO LEANTES	 JUAN CARLOS REYES FIGUEROA	FECHA AÑO MES DIA 2007 08 27

CONTENIDO

INTRODUCCIÓN	1
1 DESCRIPCIÓN DEL PROYECTO	4
1.1 DEFINICIÓN DEL PROBLEMA	4
1.2 OBJETIVOS DEL PROYECTO	10
1.2.1 Objetivo general	10
1.2.2 Objetivos específicos	10
1.3 JUSTIFICACIÓN, IMPACTO Y VIABILIDAD	11
1.3.1 Justificación	11
1.3.2 Impacto y viabilidad	12
2 MARCO TEÓRICO	13
2.1 SISTEMAS EXPERTOS	13
2.1.1 Introducción	13
2.1.2 Inteligencia artificial (IA)	14
2.1.3 Sistemas expertos (SE)	15
2.1.4 Estructura básica de un sistema experto	16
2.1.5 Campos de Aplicación	17
2.1.6 Ventajas y desventajas de los sistemas expertos	19
2.1.7 Ejemplos de sistemas expertos	20
2.2 CONCEPTOS FUNDAMENTALES DE LA LÓGICA DIFUSA	21
2.2.1 Introducción a la Lógica Difusa: Breve Historia	22
2.2.2 Conjuntos Difusos y Funciones Características	23
2.2.2.1 Operaciones con conjuntos difusos	26
2.2.2.2 Relaciones difusas	27
2.2.3 Inferencia Difusa	29
2.2.3.1 Fundamentos de lógica proposicional	29

2.2.3.2	Implicación difusa	32
2.2.3.3	Reglas difusas	34
2.2.4	Diagrama de Bloques de un Sistema Basado en Técnicas de Lógica Difusa	35
2.2.4.1	Mecanismos de inferencia	36
2.2.4.1.1	Conectivos lógicos entre antecedentes	38
2.2.4.1.2	Implicación difusa. Superficies de implicación	40
2.2.4.1.3	Agregación lógica	41
2.2.4.2	Métodos de defusificación	42
2.3	ALGUNAS TERMINOLOGÍAS	43
3	DESARROLLO DEL PROTOTIPO	49
3.1	FASE DE FORMACIÓN	49
3.1.1	Análisis de la Información Recopilada	49
3.1.1.1	Planeación de Proyectos	50
3.1.1.2	Metodologías de Desarrollo software	50
3.2	FASE DE DESARROLLO	51
3.2.1	Metodología	51
3.2.2	Lenguaje Unificado de Modelado UML	53
3.2.3	Primer Prototipo	53
3.2.4	Segundo Prototipo	58
4	INTEGRACIÓN DE LA INTERFAZ GRÁFICA DEL USUARIO CON EL MOTOR DE INFERENCIA	63
4.1	Descripción de la Interfaz Gráfica Principal	64
4.2	Descripción de la ventana de la Reglas	69
4.3	Descripción de la ventana de la resultados	72
4.4	Descripción de la ventana la GRÁFICA 3D	76
5	PRUEBAS DEL PROTOTIPO	80

5.1	DESCRIPCIÓN DE LAS VARIABLES DE ENTRADA Y SALIDA	80
5.1.1	Volumen del Tráfico	80
5.1.2	Intensidad Solar	81
5.1.3	Velocidad del Viento	82
5.1.4	Temperatura	83
5.2	DESCRIPCIÓN DE LOS GASES CONTAMINANTES	83
5.2.1	Monóxido de Carbono (CO)	84
5.2.2	Óxidos de Nitrógeno (NOX)	85
5.2.3	Ozono (O3)	86
5.2.4	Material Particulado (PST, PM10 y PM2,5).	86
5.2.5	Dióxido de Azufre (SO2)	88
5.3	INFORME AMBIENTAL DE BUCARAMANGA	88
	CONCLUSIONES	94
	RECOMENDACIONES	96
	BIBLIOGRAFÍA	97
	ANEXOS	100

LISTA DE FIGURAS

Figura 1. Diagrama de Paquetes de SEXDP	8
Figura 2. Paquete Interfaz Gráfica del Usuario	8
Figura 3. Subpaquete “para” del paquete “Motor de Inferencia”	8
Figura 4. Lógica clásica versus lógica difusa	33
Figura 5. Esquema general de un sistema basado en lógica difusa	35
Figura 6. Actuación de los operadores and, or y not, según la lógica clásica y la lógica difusa	39
Figura 7. Conjunto difuso o borroso	43
Figura 8. Función de Pertenencia Triangular	44
Figura 9. Función de Pertenencia Trapezoidal	44
Figura 10. Función de Pertenencia Tipo Gaussiana	44
Figura 11. Función de Pertenencia Curva de Bell	45
Figura 12. Función de pertenencia Singleton	45
Figura 13. Función de Pertenencia Tipo Sigmoidal	45
Figura 14. Variable lingüística	46
Figura 15. Base de reglas	46
Figura 16. Mecanismos de inferencia	47
Figura 17. Prototipado Evolutivo	52
Figura 18. Diagrama de casos de uso de la interfaz gráfica	54
Figura 19. Ventana principal de la interfaz gráfica (primer prototipo)	55
Figura 20. Casos de uso del módulo parámetros	56
Figura 21. Ventana principal para ingresar al módulo Parámetros	57
Figura 22. Ventana principal módulo parámetros. (Primer prototipo)	57
Figura 23. Módulo parámetros opciones validadas	58
Figura 24. Ventana principal	59
Figura 25. Módulo de parámetros y variables lingüísticas	60
Figura 26. Módulo de parámetros y variables lingüísticas integrada a la	

interfaz principal	60
Figura 27. Ejemplo1 del módulo de parámetros y variables lingüísticas	61
Figura 28. Ejemplo2 del módulo de parámetros y variables lingüísticas	62
Figura 29. Ventana principal de la variable lingüística y conjuntos borrosos	65
Figura 30. Ventana principal de las reglas	70
Figura 31. Ventana principal de los resultados	73
Figura 32. Ventana principal del grafico en 3D	76

LISTA DE TABLAS

Tabla 1. Ubicación de las estaciones	5
Tabla 2. Modelos Funcionales de los Sistemas Expertos	18
Tabla 3. Tabla de verdad de las principales operaciones lógicas	30
Tabla 4. Correspondencia entre operadores lógicos y de teoría de conjuntos	30
Tabla 5. Correspondencia entre operadores lógicos y algebraicos	31
Tabla 6. Tablas clásicas de verdad de los operadores and, or y not.	38
Tabla 7. Tablas de verdad para los operadores and, or y not difusos.	38
Tabla 8. Datos tomados por la estación centro de la CDMB en el año 2003.	82
Tabla 9. Unidades de los contaminantes.	84
Tabla 10. Ejemplo de valores medidos en una estación.	90
Tabla 11. Valores de IBUCA correspondientes a cada contaminante según el ejemplo.	91
Tabla 12. Categorización de los valores de IBUCA.	92
Tabla 13. Valores máximos permitidos para cada contaminante según la nueva Resolución 0601.	93

AGRADECIMIENTOS

*Doy gracias a mi padre Luis por darme
fortalezas para ser quien soy
A mi madre Maria por ser el gran soporte que tengo
en mi vida y por soportarme en los momentos difíciles de mi vida
A mis hermanos Carlos, Angel y Cecilia por apoyo constante
A todos los amigos Yovan, Jaiver, Paula, Johanna,
Astrid por el ánimo que me dieron
Al director y codirector por sus grandes enseñanzas y apoyo
continuo y al grupo Giema por su colaboración en especial a Raúl
A todas las personas que me apoyaron para la realización del
proyecto*

Edwin Ernesto Anaya Almeida

RESUMEN

TITULO:

DESARROLLO E IMPLEMENTACION DE LA INTERFAZ GRÁFICA DEL USUARIO Y SU INTEGRACION CON EL PAQUETE “MOTOR DE INFERENCIA DIFUSO” DEL PROTOTIPO SISTEMA EXPERTO PARA LA POLUCION DEL AIRE (SEXDP)*.

AUTOR:

Edwin Ernesto Anaya Almeida**

PALABRAS CLAVES:

Lógica difusa, motor de inferencia, sistema experto, polución aire, software libre.

DESCRIPCION:

El presente proyecto muestra la aplicación que tiene la lógica difusa o borrosa en el campo de diagnóstico de la polución del aire producida por fuentes móviles; esta interfaz de usuario se ha integrado a los módulos del motor de inferencia para conformar un proyecto principal llamado Sistema Experto Difuso para la Polución del Aire (SEXDP) que fue financiado por el DIF de físico mecánicas. Este prototipo es la primera versión que se realizó en el área de estudio de la polución del aire utilizando la lógica difusa.

El principal aporte que traerá este primer prototipo de sistema experto a la comunidad universitaria será diagnosticar la cantidad de polución en el aire, lo cual es de gran importancia para las personas que trabajan en el campo del medio ambiente, como es el caso del Grupo de Investigación en Energía y Medio Ambiente (GIEMA) y el Centro de Investigaciones Ambientales (CEIAM), quienes interactuarán directamente con el prototipo.

Dado que la salud pública es una prioridad social, este prototipo además sirve como apoyo en las tomas de decisiones en aspectos relacionados con los efectos de los contaminantes sobre la salud humana. Para el desarrollo de este prototipo se utilizó software libre como Java y los diseños se realizaron usando la metodología UML (Lenguaje Unificado de Modelado), esto es de gran ventaja debido a que no hay gastos por licencias de software.

* Trabajo de investigación

** Facultad de Ciencias Físico Mecánicas, Escuela de Ingeniería de Sistemas e Informática, Director: Ms. Manuel Guillermo Florez Becerra

ABSTRACT

TITLE:

DEVELOPMENT AND IMPLEMENTATION OF THE GRAPHICAL USER INTERFACE AND ITS INTEGRATION WITH THE PACKAGE "FUZZY INFERENCE MOTOR" OF THE PROTOTYPE EXPERT SYSTEM FOR THE POLLUTION OF THE AIR (SEXDP) *.

AUTHOR

Edwin Ernesto Anaya Almeida **

KEY WORDS

Fuzzy Logic, Inference Motor, Expert System, Pollution Air, Free Software.

DESCRIPTION

The present project shows to the importance that the fuzzy logic has in the field of air pollution diagnosis produced by movables sources, this user interface has been integrated to the modules of the inference motor to conform a main project called Diffuse Expert System for the Air Pollution (SEXDP), such was financed by the physical-mechanical DIFE. This prototype is the first version made in the air pollution area using the fuzzy logic.

The impact that this first prototype of expert system will be to bring to the university community a diagnose of the air pollution, it will be of great importance for people who work in the environment field as the Group of Investigation in Energy and Environment (GIEMA) and the Environmental investigations Center (CEIAM), because they are the people who will work with the prototype.

In the other hand the public health is a social priority and this prototype will serve as a support to make decision in the aspects related to the effects of the polluting agents on the human health. For the development of this prototype was used free software like Java and the designs were made using methodology UML (Unified Modeling Language), this is a great advantage because doesn't need to pay by licenses.

* Work of Investigation

** Physic-Mechanical Sciences Faculty, School of Engineering of Systems and Computer science, Direct by: Ms. Manuel Guillermo Florez Becerra

INTRODUCCIÓN

El propósito de este proyecto de investigación es modelar la polución del aire aprovechando las grandes ventajas que tiene la Lógica Difusa. Este proyecto de investigación muestra, brevemente, una descripción de lo que es la contaminación ambiental producida por los agentes móviles, haciendo de este un proyecto novedoso, debido a que nadie ha trabajado este tema con lógica difusa. Esta investigación surgió como proyecto de maestría en el Grupo de Investigación en Energía y Medio Ambiente (GIEMA) de la Escuela de Ingeniería Mecánica de la Universidad Industrial de Santander.

El objetivo principal es desarrollar una interfaz Hombre-Máquina que pueda dar diagnóstico de la polución del aire, en esto tiene que ver mucho los expertos en el tema, debido a que un sistema experto depende de la información que le suministra el experto humano (investigador, ingeniero, médico, analista, empresario, etc.) para resolver un problema determinado.

La interacción entre un sistema experto y un usuario se realiza en lenguaje natural. También es altamente interactiva y sigue el patrón de la conversación entre seres humanos. Para conducir este proceso de manera aceptable para el usuario es muy importante el diseño de la interfaz de usuario. Un requerimiento básico del interfaz es la sencillez para interactuar. Para obtener información fiable del usuario hay que poner especial cuidado en el diseño de las cuestiones. Esto puede requerir diseñar el interfaz usando menús o gráficos¹.

¹ Viejo Hernando Diego (2003). Sistemas Expertos. Consultado en <http://www.divulgacion.com/cursos/cursos.xml?numero=2&nombre=2003-9-26a&numLecc=1>.

El sistema experto modela el proceso de razonamiento humano con un paquete conocido como el motor de inferencia. Dicho motor de inferencia trabaja con la información contenida en la base de conocimientos y la base de hechos para deducir nuevos hechos. Contrasta los hechos particulares de la base de hechos con el conocimiento contenido en la base de conocimientos para obtener conclusiones acerca del problema.

El presente trabajo se ve representado en 6 capítulos los cuales sintetizan todo el esfuerzo que se realizó en el grupo Giema con los autores del proyecto:

DESCRIPCIÓN DEL PROYECTO Se presenta una definición del problema que estamos tratando, que es el de la contaminación del aire producida por agentes móviles y además se da el objetivo general y objetivos específicos que se plantearon junto con la justificación, impacto y viabilidad del proyecto.

MARCO TEÓRICO Aquí se realiza un resumen general de lo que tiene que ver con sistemas expertos y lógica difusa, además se hace referencia de la fuente bibliográfica.

DESARROLLO DEL PROTOTIPO Este capítulo plantea quizás lo más importante que es la fase de formación y desarrollo de los prototipos todo basado en el análisis de la información y metodología.

INTEGRACIÓN DE LA INTERFAZ GRÁFICA Presenta como fue la integración de la interfaz gráfica de usuario con los módulos y por supuesto se realiza unas pruebas para ver que prototipo final funciona adecuadamente.

PRUEBAS DEL PROTOTIPO Las pruebas se realizan para verificar la calidad y el desempeño del prototipo de sistema experto.

CONCLUSIONES Y RECOMENDACIONES Las conclusiones son reflexiones después de haber terminado este proyecto y las recomendaciones son sugerencias que se hacen para el uso del proyecto y para dar luces de hacia donde deben apuntar los futuros desarrollos del prototipo.

BIBLIOGRAFÍA Contiene las referencias bibliográficas que se utilizaron durante el proceso del proyecto.

1. DESCRIPCIÓN DEL PROYECTO

1.1 DEFINICIÓN DEL PROBLEMA

En la actualidad existen problemas que contribuyen al deterioro de la calidad del aire, sin embargo, aquí solo analizaré la polución producida por los vehículos. La contaminación del aire altera los elementos que componen la atmósfera, afectando el desarrollo de la vida, de nuestros ecosistemas y del medio ambiente en general.

Los vehículos que ruedan por la ciudad de Bucaramanga, han contribuido al incremento de la concentración de gases tóxicos en el aire y material particulado, como por ejemplo: óxidos de nitrógeno, monóxido de carbono, metano, dióxido de carbono, ozono, entre otros, lo que ha deteriorado las condiciones ambientales en algunos sectores y puntos neurálgicos de la ciudad.

La Red de Monitoreo de Calidad del Aire de la CDMB² posee 5 estaciones químicas y cuatro estaciones meteorológicas ubicadas, estratégicamente, en el área metropolitana de Bucaramanga con el objetivo de monitorear en tiempo real cinco contaminantes criterio generados por fuentes contaminantes fijas³ y móviles⁴.

En la tabla 1 se muestra el nombre, la localización y los parámetros medidos de las diferentes estaciones de monitoreo.

² CDMB: Corporación Autónoma Regional para la Defensa de la meseta de Bucaramanga

³ Fuentes Fijas: Empresas e Industrias.

⁴ Fuentes Móviles: Vehículos Automotores.

Tabla 1. Ubicación de las estaciones

No	NOMBRE	LOCALIZACIÓN	PARÁMETROS MEDIDOS
1	CENTRO	Esquina calle 34 con carrera 15	Material Particulado PM10 Monóxido de Carbono Óxidos de Nitrógeno Dióxido de Azufre Ozono Meteorología
2	CIUDADELA	Terraza Colegio Aurelio Martínez Mutis, Real de minas	Monóxido de Carbono Ozono
3	FLORIDA	Terraza edificio Telebucaramanga zona sur	Material Particulado PM10 Monóxido de Carbono
4	CHIMITA	Terpel	Material Particulado PM10 Monóxido de Carbono Óxidos de Nitrógeno Dióxido de Azufre Ozono Meteorología
5	PTAR	PTAR CDMB	Meteorología
6	UIS	Terraza Ing. Química UIS	Meteorología
7	NORTE	Terraza Hospital Local del Norte	Óxidos de Nitrógeno

Tabla tomada de la página www.cdmb.gov.co

Como se puede observar en la tabla 1, los parámetros medidos son en mayor proporción en la estación Centro y Chimitá, debido a la gran confluencia de automóviles de esas zonas de Bucaramanga. De lo anterior se puede concluir que los vehículos que conforman el transporte público masivo, constituyen uno de los agentes de propagación de la contaminación atmosférica más representativos

debido a la obsolescencia y mal estado de los vehículos, la falta de mantenimiento preventivo y correctivo; y el uso de combustibles de mala calidad.

El incremento preocupante de la polución en los centros urbanos, se debe a las actividades antrópicas⁵, pero con marcada diferencia, por la emisión de contaminantes provenientes del parque automotor, que utiliza combustibles fósiles para su movilización. Estos agentes contaminantes degradan la calidad del aire, provocan lluvia ácida, contribuyen al efecto invernadero, destruyen el medio ambiente y son causantes de diversas enfermedades en la población humana, desmejorando su calidad de vida.

Conforme a los lineamientos de las normas ambientales, en el área de jurisdicción, la CDMB tiene implementado el programa de certificación vehicular, a través de Centros de Diagnóstico, debidamente aprobados para tal fin. Estos establecimientos tienen la responsabilidad de emitir un diagnóstico veraz, sobre la composición de las emisiones provenientes de la combustión de los automotores que requieran este examen en sus instalaciones, y respaldar el resultado de la prueba realizada, con un documento debidamente diligenciado, firmado y sellado, por el funcionario encargado del diagnosticentro respectivo.

La comunidad del Área Metropolitana de Bucaramanga (AMB)⁶ exige que el aire que se respire, no represente ningún peligro para la salud. Para hacer un diagnóstico de la calidad del aire en la ciudad de Bucaramanga, es necesario disponer de herramientas y sistemas de información potentes y flexibles que ayuden a las autoridades, secretaria de salud y medio ambiente y de control (CDMB) o hacer un diagnóstico lo mas eficaz posible y de manera sencilla para el usuario y tomar las acciones respectivas. Para la realización del proyecto de investigación “Modelado y Simulación de la Contaminación del aire atmosférico

⁵ Antrópico: acciones hechas o relacionadas por el hombre

⁶ Área Metropolitana de Bucaramanga

producido por fuentes móviles del AMB, considerando los aspectos cualitativos y de incertidumbre” financiado por el DIF de Fisicomecánica, es necesario desarrollar una Interfaz Hombre-Maquina que hace parte del Sistema Experto Difuso para la Polución del Aire (SEXDP)⁷, que sea de fácil utilización y en un ambiente amigable por parte de los expertos en la determinación de la calidad del aire de Bucaramanga.

Gracias al uso que se hace de la metodología “Orientada a Objetos” en el proyecto General (SEXDP) y con el propósito de facilitar la dirección y coordinación del desarrollo de este proyecto se dividió el sistema en paquetes.

Por un lado el paquete Dominio de Arquitectura agrupa los paquetes para la plataforma del software sobre los cuales la aplicación funciona:

La Interfaz Gráfica del Usuario. Permite al usuario elegir comandos, iniciar programas y ver listas de archivos y otras opciones utilizando las representaciones visuales (iconos) y las listas de elementos del menú. Las selecciones pueden activarse bien a través del teclado o con el ratón.

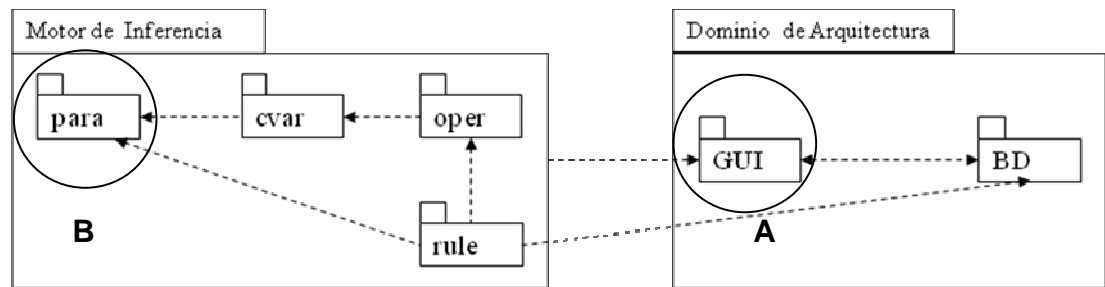
Base de Datos Contiene los datos sobre un problema que se han descubierto durante una consulta. Durante una consulta con el sistema experto, el usuario introduce la información del problema actual en la base de datos.

Motor de Inferencia. El sistema experto modela el proceso de razonamiento humano con un módulo conocido como el motor de inferencia. Dicho motor de inferencia trabaja con la información contenida en la base de datos para deducir nuevos datos. Contrasta los hechos particulares de la base de datos con el

⁷ SEXDP: Sistema Experto Difuso para la Polución del Aire

conocimiento contenido en la base de conocimientos para obtener conclusiones acerca del problema.

Figura 1. Diagrama de Paquetes de **SEXDP**



Fuente: tesis, Modelado y Simulación de la Contaminación del aire atmosférico producido por fuentes móviles del AMB, considerando los aspectos cualitativos y de incertidumbre⁸

A. Paquete librería “Interfaz Gráfica del Usuario”

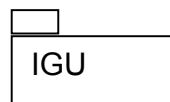


Figura 2. Paquete Interfaz Gráfica del Usuario

Este paquete librería provee al programador de la herramienta con las clases necesarias para desarrollar la Interfaz gráfica del usuario; tales como ventana, y del tipo caja: caja de texto, caja de lista, caja combo y botón, etc. También es importante la clase menú y la barra que contiene los menús. El objeto principal de este módulo es construir estos objetos que acabamos de mencionar, aplicando el lenguaje de programación ya definido para este proyecto que es “java”.

⁸ Jaimes Becerra, Marcela, Ing. De Sistemas

B. Para:

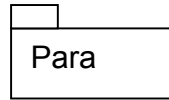


Figura 3. Subpaquete “para” del paquete “**Motor de Inferencia**”

Este paquete módulo provee al desarrollador con las funciones más 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.

1.2. OBJETIVOS DEL PROYECTO

1.2.1 Objetivo general

Construcción de la Interfaz Gráfica del Usuario y de su integración con el paquete “Motor de Inferencia Difuso” del prototipo de software de simulación de la polución del aire en Bucaramanga (SEXDP).

1.2.2 Objetivos específicos

- ✍ Hacer una revisión bibliográfica de los diferentes Sistemas Expertos basados en Lógica Difusa para la determinación de la contaminación del aire y áreas afines.
- ✍ Realizar el diseño e implementación de la interfaz Hombre-Maquina documentado con UML.
- ✍ Implementar los ambientes de Inicio principal, configuración del FIS (Sistema de Inferencia Difuso), partición de los conjuntos borrosos, variables de salida, reglas de inferencia y evaluación e implementación de los resultados.
- ✍ Crear, Inicializar, añadir y remover los conjuntos de variables lingüísticas.
- ✍ Realizar un manual de documentación que sirva de soporte para usuario del prototipo.

1.3 JUSTIFICACIÓN, IMPACTO Y VIABILIDAD

1.3.1 Justificación

El aire es un componente natural para el desarrollo de la vida en la tierra, sin él no podrían existir en nuestro planeta las plantas, los animales ni los seres humanos. Existen varias causas que contribuyen a la contaminación del aire, entre las múltiples causas este proyecto solo se enfocará en el tráfico vehicular y su contribución al deterioro de la calidad del aire.

Será de gran ayuda un prototipo de un Sistema Experto basado en la Lógica Difusa para analizar problemas de la polución del aire del AMB. La Lógica Difusa crea aproximaciones matemáticas en la resolución de problemas complejos y de características no lineales y donde se dispone de poca información, produciendo resultados aproximados a partir de datos imprecisos, siendo por ello, especialmente útil en aplicaciones de tipo informático (Inteligencia Artificial). La lógica difusa a pesar de su corta historia ha tenido un crecimiento muy rápido, debido a que es capaz de resolver problemas relacionados con la incertidumbre de la información o del conocimiento, proporcionando un método formal para la representación del conocimiento de forma compresible y cerca al lenguaje natural por parte de los ciudadanos comunes. Esto hace que se le pueda asegurar y casi garantizar un amplio campo de aplicaciones con un alto grado de interés.

Para este caso, se desarrollará una interfaz grafica de usuario para el prototipo de software de simulación basado en la lógica difusa para predecir la polución del aire en el área metropolitana de Bucaramanga producidas por las fuentes móviles. Se realizará la primera versión en el área de la calidad del aire. Dentro de este proyecto de investigación el desarrollo de una Interfaz Hombre-Maquina es importante para el empleo exitoso a nivel de usuario del Sistema Experto.

Impacto y viabilidad

Al analizar el impacto y la viabilidad del desarrollo de este proyecto, se tuvieron en cuenta los siguientes aspectos:

	Técnico	Económico	Social
Impacto	El prototipo como un Sistema Experto ayudará al diagnóstico de la polución del aire, el cual será beneficioso para la población de la ciudad.	Los estudiantes de la UIS tendrán acceso al prototipo sin ningún costo.	Mejora en forma notable la confianza entre las personas que tienen problemas de salud por culpa de la polución del aire.
Viabilidad	Los recursos tecnológicos para desarrollar este proyecto están dispuestos de manera inmediata. Se tiene fácil acceso a las herramientas de desarrollo software.	Las tecnologías de desarrollo software a utilizar, no generan costos, al ser de libre distribución.	No afecta el libre desarrollo de las actividades dentro de la ciudad y la universidad.

2. MARCO TEÓRICO

2.1. SISTEMAS EXPERTOS

Desde que naciera la Inteligencia Artificial como tal ciencia, a principios de los 60, hasta la actualidad, se han venido realizando numerosos trabajos, para el desarrollo de los diversos campos que componen la Inteligencia Artificial. De todos esos campos, quizá el que puede tener mayor número de aplicaciones prácticas sea el de los Sistemas Expertos, siendo dichas aplicaciones de utilidad en temas tan variados que pueden ir desde la medicina hasta la enseñanza.

Palabras claves: Sistemas Expertos, inteligencia, conocimiento, razonamiento

2.1.1 Introducción

En palabras de un gran matemático, Blaise Pascal: "Como no se puede ser universal y saber todo acerca de todo, es mejor saber un poco de cada tema, que saber todo sobre un solo tema ", así fue como inició la inquietud de los investigadores en los años 40, con la idea de un sistema general de resolución de problemas (General Problem Solver: GPS), que por medio de una serie de algoritmos basados en análisis, más o menos exhaustivos, fuera capaz de resolver toda clase de problemas relativos a juegos de estrategias y demostraciones automáticas. La ambición era grande, así como lo fue la decepción que tuvieron al ver que a pesar de los progresos teóricos y de algunos programas espectaculares, no obtuvieron los resultados que se esperaban⁹.

⁹ Castro, G. "Tendencias de los Sistemas Expertos". Netmedia.info URL: http://www.netmedia.info/netmedia/articulos.php?id_sec=32&id_art=2255

Entonces se da un cambio en la dirección del problema, se enfocarían en un dominio específico, y recurriendo al conocimiento sobre el tema obtenido de las mismas personas, pretenden simular el razonamiento de un experto humano ante alguna situación.

Así es como surge una de las ramas de mayor avance dentro de la Inteligencia Artificial, los Sistemas Expertos; con la idea de automatizar la labor del experto, partiendo en ocasiones de información insuficiente o incompleta.

A continuación se presentará una introducción al amplio tema de la Inteligencia Artificial, para luego llegar al campo de los Sistemas Expertos, dándonos una descripción general de lo que es, revisando su concepto, estructura, campos de aplicación y ejemplos que fortalezcan la comprensión de los mismos.

2.1.2. Inteligencia artificial (IA)

La Inteligencia Artificial comenzó como el resultado de la investigación en psicología cognitiva y lógica matemática. Se ha enfocado en la explicación del trabajo mental y construcción de algoritmos de solución a problemas de propósito general.

La Inteligencia Artificial es una combinación de la ciencia del computador, fisiología y filosofía, tan general y amplio como eso; reúne varios campos (robótica, sistemas expertos, por ejemplo), todos los cuales tienen en común la creación de máquinas que pueden "pensar".

La idea de construir una máquina, que pueda ejecutar tareas que parecen necesitar de la inteligencia humana para llevarse a cabo es un atractivo. Las tareas que han sido estudiadas desde este punto de vista incluyen juegos,

traducción de idiomas, comprensión de idiomas, diagnóstico de fallas, robótica, suministro de asesoría experta en diversos temas.

Es así como los sistemas de administración de base de datos cada vez más sofisticados, la estructura de datos y el desarrollo de algoritmos de inserción, borrado y locación de datos, así como el intento de crear máquinas capaces de realizar tareas que son pensadas como típicas del ámbito de la inteligencia humana, acuñaron el término Inteligencia Artificial en 1956.

Trabajos teóricos fundamentales fueron el desarrollo de algoritmos matemáticos por Warren McCullock y Walter Pitts, en 1943, necesarios para posibilitar el trabajo de clasificación, o funcionamiento en sentido general, de una red neuronal. En 1949 Donald Hebb desarrolló un algoritmo de aprendizaje para dichas redes neuronales creando, en conjunto con los trabajos de McCullock y Pitts, la escuela creacionista. Esta escuela se considera hoy como el origen de la Inteligencia Artificial, sin embargo se trató poco por muchos años, dando paso al razonamiento simbólico basado en reglas de producción, lo que se conoce como sistemas expertos.

2.1.3. Sistemas Expertos (SE)

Un Sistema Experto es una aplicación que simula el comportamiento de un experto humano, en el sentido de que es capaz de decidir cuestiones, aunque sea en un campo restringido.

Para esto, se debe tener en cuenta que la principal característica del experto humano viene a ser el conocimiento o habilidades profundas en ese campo concreto, por consiguiente, un Sistema Experto debe ser capaz de representar ese conocimiento profundo con el objetivo de utilizarlo para resolver problemas, justificar su comportamiento e incorporar nuevos conocimientos. Se podría incluir

también el hecho de poder comunicarse en lenguaje natural con las personas, aunque esta capacidad no es tan determinante como las anteriores de lo que se puede definir como Sistema Experto¹⁰.

Un experto humano es alguien con un conocimiento profundo, que no solamente soluciona problemas específicos sino que también los pone dentro de un contexto más amplio. Ellos "transfieren" todos sus conocimientos y experiencias a una computadora que posiblemente los sustituya en su labor. Pero los sistemas aún carecen de algo que tal vez jamás vayan a tener: Inteligencia natural.

Los sistemas expertos se aplican por norma general en problemas que implican un procedimiento basado en el conocimiento. O sea, un procedimiento que comprende la utilización de normas o estructuras que contengan conocimientos y experiencias de expertos, deducción lógica de conclusiones, capacidad de interpretar datos ambiguos y la manipulación de conocimientos afectados por la probabilidad.

La mejor forma de entender el funcionamiento de un Sistema Experto es conociendo los componentes que conforman la estructura del mismo, y las funciones que desempeñan individualmente. Dichos componentes se tratarán seguidamente.

2.1.4. Estructura básica de un SE

Un Sistema Experto está conformado por:

- Base de conocimientos (BC): Contiene conocimiento modelado extraído del diálogo con el experto

¹⁰ Ibarreta, J. "Sistemas Expertos: Áreas de aplicación". Geocities. URL: <http://www.geocities.com/javierml.geo/doc/SistemasExpertos.html>

- Base de hechos (Memoria de trabajo): contiene los hechos sobre un problema que se ha descubierto durante el análisis
- Motor de inferencia: Modela el proceso de razonamiento humano
- Módulos de justificación: Explica el razonamiento utilizado por el sistema para llegar a una determinada conclusión
- Interfaz de usuario: es la interacción entre el SE y el usuario, y se realiza mediante el lenguaje natural

El conocimiento del experto se obtiene de alguna forma (Subsistema de adquisición de conocimientos) y se organiza en una base de conocimientos, y en función de los datos disponibles de la aplicación (base de hechos o base de datos) 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 interfase hombre-máquina. Además algunos sistemas expertos manejan un componente de explicación.

2.1.5. Campos de aplicación

Un sistema experto es muy eficaz cuando tiene que analizar una gran cantidad de información, interpretándola y proporcionando una recomendación a partir de la misma, a continuación una tabla mostrando el uso de los Sistemas Expertos ante distintos problemas.

Tabla 2 Modelos Funcionales de los Sistemas Expertos

CATEGORIA	TIPO DE PROBLEMA	USO
Interpretación	Deducir situaciones a partir de datos observados.	Análisis de imágenes, reconocimiento del habla, inversiones financieras.
Predicción	Inferir posibles consecuencias a partir de una situación.	Predicción meteorológica, previsión del tráfico, evolución de la bolsa.
Diagnóstico	Deducir fallos a partir de sus efectos.	Diagnostico medico, detección de fallos en electrónica.
Diseño	Configurar objetos bajo ciertas especificaciones.	Diseño de circuitos, automóviles, edificios, etc.
Planificación	Desarrollar planes para llegar a unas metas.	Programaron de proyectos e inversiones, planificación militar.
Monitorización o supervisión	Controlar situaciones donde hay planes vulnerables.	Control de centrales nucleares y factorías químicas.
Depuración	Prescribir remedios para funcionamientos erróneos.	Desarrollo de software y circuitos electrónicos.
Reparación	Efectuar lo necesario para hacer una corrección.	Reparar sistemas informáticos, automóviles, etc.
Instrucción	Diagnostico, depuración y corrección de una conducta.	Corrección de errores, enseñanza.
Control	Mantener un sistema por un camino previamente trazado. Interpreta, predice y supervisa su conducta.	Estrategia militar, control de tráfico aéreo.
Enseñanza	Recoger el conocimiento y mostrarlo.	Aprendizaje de experiencia.

A pesar de que los campos de aplicación son grandes, es importante conocer las ventajas y desventajas que se presentan para la utilización de y Sistema Experto.

2.1.6. Ventajas y Desventajas de los Sistemas Expertos

A la hora de tomar la decisión de desarrollar un Sistema Experto, se deben analizar las distintas ventajas y desventajas que estos tienen.

Ventajas

Entre las principales ventajas de un Sistema Experto, se pueden mencionar:

- Los Sistemas Expertos que se codifican y tratan el conocimiento de un experto, alcanzan el nivel de prestaciones de un experto humano
- La separación entre base de conocimiento y motor de inferencia proporciona una gran flexibilidad al Sistema
- Los Sistemas Expertos permiten llegar a una reflexión sobre el conocimiento y los procedimientos, para toma de decisiones realizados por los expertos
- Posibilidad de trazar el razonamiento seguido por el Sistema Experto.
- Algunos Sistemas Expertos utilizan coeficientes de verosimilitud para ponderar sus deducciones
- En manipulación de símbolos, los Sistemas Expertos abren el campo a nuevas aplicaciones, no automatizadas hasta la fecha

Desventajas

Aunque son grandes las ventajas, no hay que dejar de lado las desventajas que se pueden presentar:

- La publicidad realizada alrededor de esta técnica seduce a muchas personas pero crea aspiraciones imposibles de satisfacer
- La extracción del conocimiento es el problema más complejo que se les plantea a los ingenieros de conocimientos
- Ciertos problemas están ligados a la representación del conocimiento
- Los SE son incapaces de reconocer un problema para el que su propio conocimiento es inaplicable o insuficiente

- Los motores de inferencia poseen algunos límites
- Las interfaces con usuarios no son lo suficientemente amigables
- Las aplicaciones suministradas por los SE son, frecuentemente, esotéricas
- Falta personal competente para investigar y desarrollar aplicaciones
- Campo de aplicaciones restringido y específico
- El costo es bastante alto

Aunque se conozcan las ventajas y las desventajas de los Sistemas Expertos, no se logra una visión clara de los Sistemas Expertos y su capacidad, hasta que se conocen algunos ejemplos en aplicaciones reales.

2.1.7. Ejemplos de Sistemas Expertos

A continuación se presentan algunos Sistemas Expertos que se han desarrollado para la solución de diversos problemas.

- MYCIN¹¹ es un Sistema Experto para la realización de diagnósticos en el área de la medicina, iniciado por Ed Feigenbaum y posteriormente desarrollado por E. Shortliffe y sus colaboradores. Su función es la de aconsejar a los médicos en la investigación y determinación de diagnósticos en el campo de las enfermedades infecciosas de la sangre
- XCON¹² es un Sistema Experto para configuraciones, desarrollado por la Digital Equipment Corporation. Según los deseos individuales del cliente se configuran redes de ordenadores VAX. Ya que el abanico de productos que se ofrecen en el mercado es muy amplio, la configuración completa y correcta de un sistema de estas características es un problema de gran

¹¹ Criado, J. "Sistemas Expertos" Worldonline.es URL: <http://home.worldonline.es/jmariocr/>

¹² Criado, J. "Sistemas Expertos" Worldonline.es URL: <http://home.worldonline.es/jmariocr/>

complejidad. Responde esencialmente a dos preguntas: ¿Pueden conjugarse los componentes solicitados por el cliente de forma conveniente y razonable? Y ¿Los componentes de sistema especificados son compatibles y completos?. Las respuestas a estas preguntas son muy detalladas. XCON es capaz de comprobar y completar los pedidos entrantes mucho más rápido y mejor que las personas encargadas de hacerlo antes que él

- DELTA¹³ sistema experto que ayuda a los mecánicos en el diagnóstico y reparación de locomotoras diesel-eléctricas, DELTA no sólo da consejos expertos, sino que también presenta informaciones por medio de un reproductor de vídeo
- En 1978 se desarrollo otro sistema experto de éxito: PROSPECTOR. Este quizás impulso más la carrera por desarrollar mejores sistemas expertos, dado que su misión era predecir la posibilidad de encontrar depósitos de minerales en una región en concreto. Minerales como petróleo, gas natural, helio

2.2. CONCEPTOS FUNDAMENTALES DE LÓGICA DIFUSA¹⁴

La técnica de lógica difusa o lógica borrosa, nos permite manejar y procesar diferentes tipos de información que son inexactos o imprecisos. Es básicamente una lógica multievaluada que permite definir valores intermedios a los valores clásicos como sí/no, true/false, que han modelado satisfactoriamente una gran

¹³ Ibarreta, J. "Sistemas Expertos: Áreas de aplicación". Geocities. URL: <http://www.geocities.com/javierml.geo/doc/SistemasExpertos.html>

¹⁴ Artículo tomado de: http://www.tdx.cesca.es/TESIS_UPC/AVAILABLE/TDX-0207105-105056//04Rpp04de11.pdf

parte del razonamiento natural. Pero el razonamiento humano utiliza valores de verdad que no tienen porque ser necesariamente tan deterministas. Por medio de la lógica difusa pueden definirse nociones del tipo "más bien caliente" o "poco frío" que pueden formularse matemáticamente y ser procesados por computadoras.

Conviene recalcar que lo que es difuso, borroso, impreciso o vago no es la lógica en sí, sino el objeto que estudia: expresa la falta de definición del concepto al que se aplica. La lógica difusa permite tratar información imprecisa, como estatura media o temperatura baja, en términos de conjuntos borrosos que se combinan en reglas para definir acciones: si la temperatura es alta entonces enfriar mucho.

De esta manera, los sistemas de control basados en lógica difusa combinan variables de entrada, definidas en términos de conjuntos difusos, por medio de grupos de reglas que producen uno o varios valores de salida.

2.2.1. Introducción a la Lógica Difusa: Breve Historia

La lógica borrosa se inició a mediados de los años 60 por Lotfi A. Zadeh, profesor de ciencia de computadoras en la Universidad de California en Berkeley. Anteriormente Max Black en un artículo titulado 'Vagueness: An exercise in Logical Analysis' y Kart Menger con los artículos de 1942 'Statistical Metrics' y los de los años 50 sobre relaciones borrosas, sentaron las bases de lo que hoy es una teoría tan utilizada y con tan buenos resultados.

Debido a la variedad de sus aplicaciones la lógica difusa o Fuzzy Logic parece estar introducida en todos los sectores; control de complejos procesos industriales, diseño de dispositivos artificiales de deducción automática, construcción de artefactos electrónicos de uso doméstico y de entretenimiento, sistemas de

diagnóstico, entre otros. Más adelante se expondrá ejemplos concretos en varios campos donde se utiliza actualmente la lógica difusa.

Paralelamente al desarrollo de las aplicaciones de la lógica difusa, investigadores teóricos siguen, en la década de los ochenta, el camino iniciado por Mamdani. Así, Takagi y Sugeno desarrollan la primera aproximación para construir reglas fuzzy a partir de datos de entrenamiento, y aunque en un principio no tiene mucha repercusión, más tarde será el punto de partida para investigar la identificación de modelos fuzzy. Otro de los factores que contribuye a seguir con la investigación en este campo es el creciente interés en las redes neuronales y su similitud con los sistemas fuzzy; la tendencia es buscar vías de relación entre las dos técnicas y los resultados son los llamados neuro-fuzzy systems, sistemas fuzzy que usan métodos de aprendizaje basados en redes neuronales para identificar y optimizar sus parámetros. B. Kosko es conocido por su contribución a los sistemas neurofuzzy y con sus publicaciones introdujo en la lógica difusa a muchos lectores interesados en las redes neuronales.

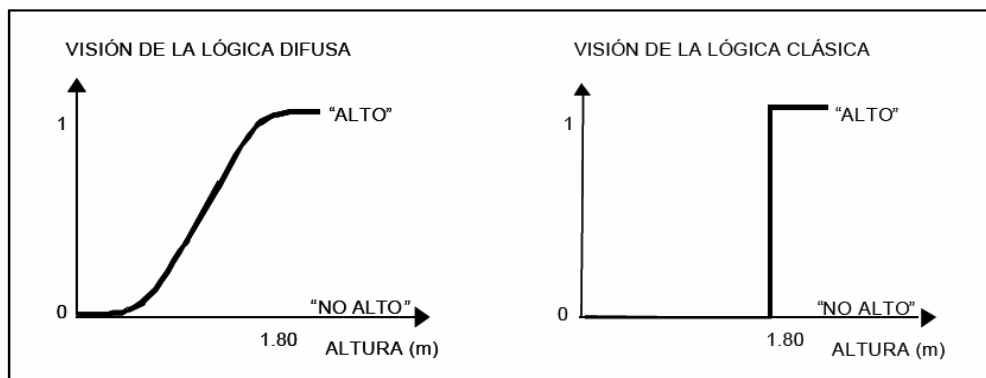
En la década de los noventa, además de las redes neuronales y los sistemas fuzzy, hacen su aparición los algoritmos genéticos. Estas tres técnicas computacionales, que pueden combinarse de múltiple maneras y se pueden considerar complementarias, son herramientas de trabajo muy potentes en el campo de los sistemas de control en la última década.

2.2.2. Conjuntos Difusos y Funciones Características

El primer ejemplo utilizado por Lofti A. Zadeh, para ilustrar el concepto de conjunto difuso, fue el conjunto “hombres altos”. Según la teoría de la lógica clásica el conjunto “hombres altos” es un conjunto al que pertenecerían los hombres con una estatura mayor a un cierto valor, que podemos establecer en 1.80 metros, por ejemplo, y todos los hombres con una altura inferior a este valor quedarían fuera

del conjunto. Así tendríamos que un hombre que mide 1.81 metros de estatura pertenecería al conjunto hombre altos, y en cambio un hombre que mida 1.79 metros de altura ya no pertenecería a ese conjunto. Sin embargo, no parece muy lógico decir que un hombre es alto y otro no lo es cuando su altura difiere en dos centímetros. El enfoque de la lógica difusa considera que el conjunto “hombres altos” es un conjunto que no tiene una frontera clara para pertenecer o no pertenecer a él: mediante una función que define la transición de “alto” a “no alto” se asigna a cada valor de altura un grado de pertenencia al conjunto, entre 0 y 1. Así por ejemplo, un hombre que mida 1.79 podría pertenecer al conjunto difuso “hombres altos” con un grado 0.8 de pertenencia, uno que mida 1.81 con un grado 0.85, y uno que mida 1.50 m con un grado 0.1. Visto desde esta perspectiva se puede considerar que la lógica clásica es un caso límite de la lógica difusa en el que se asigna un grado de pertenencia 1 a los hombres con una altura mayor o igual a 1.80 y un grado de pertenencia 0 a los que tienen una altura menor (ver figura 5).

Figura 5 Lógica clásica versus lógica difusa.



Formalmente, un conjunto clásico A , en un universo de discurso U , se puede definir de varias formas: enumerando los elementos que pertenecen al conjunto, especificando las propiedades que deben cumplir los elementos que pertenecen a ese conjunto o, en términos de la función de pertenencia $\mu_A(x)$, como se muestra en la ecuación 1:

$$\mu_A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{si } x \notin A \end{cases} \quad (1)$$

Podemos además decir que el conjunto A es matemáticamente equivalente a su función de pertenencia o característica $\mu_A(x)$, ya que 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, ver ecuación 2:

$$A = \{(x, \mu_A(x)) \mid x \in U\} \quad (2)$$

Muchos conceptos de teoría clásica de conjuntos se pueden hacer extensivos a los conjuntos difusos, otros son exclusivos e inherentes a la teoría de conjuntos difusos.

Algunos de los que más utilizados son los siguientes:

- El soporte de un conjunto difuso A en el universo de discurso U es un conjunto “crisp” (numérico) que contiene todos los elementos de U que tienen un valor de pertenencia distinto de cero en A, esto es ecuación 3,

$$\text{sup}(x) = \{x \in U \mid \mu_A(x) > 0\} \quad (3)$$

Si el soporte de un conjunto difuso no contiene ningún elemento tendremos un conjunto difuso vacío. Si el soporte de un conjunto difuso es un solo punto tendremos lo que se conoce como “singleton” difuso.

- El punto de cruce de un conjunto difuso es el punto de U cuyo valor de pertenencia al conjunto es igual a 0.5.
- Dos conjuntos difusos A y B son iguales si y sólo si sus funciones características $\mu_A(x)$ y $\mu_B(x)$ son iguales
- El conjunto difuso B contiene al conjunto difuso A, esto es $A \subseteq B$, si y sólo si $\mu_A(x) \leq \mu_B(x)$ para todo $x \in U$.

2.2.2.1 Operaciones con conjuntos difusos

Las operaciones básicas entre conjuntos difusos son las siguientes:

- El conjunto complementario \bar{A} de un conjunto difuso A es aquel cuya función característica viene definida por, ver ecuación 4:

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

- La unión de dos conjuntos difusos A y B es un conjunto difuso $A \cup B$ en U cuya función de pertenencia es, ver ecuación 5:

$$\mu_{A \cup B}(x) = \max [\mu_A(x), \mu_B(x)] \quad (5)$$

- La intersección de dos conjuntos difusos A y B es un conjunto difuso $A \cap B$ en U con función característica, ver ecuación 6:

$$\mu_{A \cap B}(x) = \min [\mu_A(x), \mu_B(x)] \quad (6)$$

Estas tres operaciones definidas para conjuntos difusos cumplen, al igual que en la teoría clásica de conjuntos, asociatividad, conmutatividad y distributividad así como las leyes de Morgan.

Sin embargo, también hay que destacar que existen dos leyes fundamentales de la teoría clásica de conjuntos como son el Principio de contradicción:

$A \cup \bar{A} = U$, y el Principio de exclusión: $A \cap \bar{A} = \emptyset$ que no se cumplen en la teoría de conjuntos difusos; de hecho una de las formas para describir en qué se diferencia la teoría clásica de conjuntos de la teoría difusa es explicar que estas dos leyes en términos de lógica difusa no se cumplen. En consecuencia, algunas de las teorías derivadas de la teoría de conjuntos como por ejemplo la de la probabilidad será muy diferente planteada en términos difusos.

Las funciones que definen la unión y la intersección de conjuntos difusos pueden generalizarse, a condición de cumplir ciertas restricciones. Las funciones que cumplen estas condiciones se conocen como Conorma Triangular (T-Conorma) y

Norma Triangular (T-Norma). Los principales operadores que cumplen las condiciones para ser t-conormas son el operador máximo y la suma algebraica $[\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x)]$ y los principales operadores que cumplen las condiciones para ser t-normas son el operador mínimo y el producto algebraico $[\mu_{A \cap B}(x) = \mu_A(x)\mu_B(x)]$. En la mayoría de las aplicaciones a la ingeniería de la lógica difusa se usan como t-conorma el operador máximo y como t-norma los operadores mínimo o producto.

2.2.2.2 Relaciones difusas

Una relación difusa representa el grado de presencia o ausencia de asociación, interacción o interconexión entre elementos de dos o más conjuntos difusos, por ejemplo: “x es mayor que y”. Supongamos U y V dos universos de discurso, la relación difusa R(U,V) es un conjunto difuso en el espacio producto UxV que se caracteriza por la función de pertenencia $\mu_R(x, y)$ donde x pertenece a U e y pertenece a V, es decir, ver ecuación (7)

$$R(U, V) = \{((x, y), \mu_R(x, y)) | (x, y) \in U \times V\} \quad (7)$$

en el caso de las relaciones difusas $\mu_R(x, y) \in [0, 1]$ y en caso de las relaciones clásicas $\mu_R(x, y) = 0$ ó 1.

Como las relaciones difusas son en si mismas un conjunto difuso en el espacio producto, las operaciones entre conjuntos y los operadores definidos anteriormente también pueden ser aplicadas a ellas. Supongamos R(x,y) y S(x,y) dos relaciones en el mismo espacio producto UxV. La intersección o unión entre R y S, que son composiciones entre las dos relaciones, se definen como, ver ecuación (8):

$$\begin{aligned} \mu_{R \cap S}(x, y) &= \mu_R(x, y) * \mu_S(x, y) \\ \mu_{R \cup S}(x, y) &= \mu_R(x, y) \oplus \mu_S(x, y) \end{aligned} \quad (8)$$

Dónde $*$ es cualquier t-norma, y \oplus es cualquier t-conorma.

Si consideramos las relaciones difusas R y S que pertenecen a diferentes espacios producto $R(U, V)$ y $S(V, W)$, por ejemplo “ x es mayor que y ” y “ y es cercano a z ”, su composición difusa se define de forma análoga a la composición clásica teniendo en cuenta que en el caso difuso la relación difusa R tiene asociada una función característica $\mu_R(x, y)$ que toma valores en el intervalo $[0,1]$ y la relación difusa S también tiene asociada una función característica $\mu_S(y,z)$ que de igual forma toma valores en el intervalo $[0,1]$. Entonces la composición difusa entre R y S , es decir $S \circ R$, cuando R y S pertenecen a universos discretos de discurso, se define como una relación difusa en $U \times W$ cuya función de pertenencia viene dada por, ver ecuación 9:

$$\mu_{R \circ S}(x, z) = \sup_{y \in V} [\mu_R(x, y) * \mu_S(y, z)] \quad (9)$$

Dónde el operador \sup es el máximo y el operador $*$ puede ser cualquier t-norma. En función de la t-norma elegida podemos obtener distintas composiciones; las dos composiciones más usadas son la composición máx-min y la composición máx-product:

- La composición máx-min de las relaciones difusas $R(U, V)$ y $S(V, W)$, es una relación difusa $S \circ R$ en $U \times W$ definida por la función de pertenencia

$$\mu_{R \circ S}(x, z) = \max_{y \in V} [\min(\mu_R(x, y), \mu_S(y, z))] \quad (10)$$

Dónde $(x,z) \in U \times W$

- La composición máx-product de las relaciones difusas $R(U,V)$ y $S(V,W)$, es una relación difusa $S \circ R$ en $U \times W$ definida por la función característica

$$\mu_{R \circ S}(x, z) = \max_{y \in V} [\mu_R(x, y) \bullet \mu_S(y, z)] \quad (11)$$

Dónde $(x,z) \in U \times W$.

2.2.3 Inferencia Difusa

Se llama reglas difusas al conjunto de proposiciones IF-THEN que modelan el problema que se quiere resolver. Una regla difusa simple tiene la forma:

“si u es A entonces v es B ” dónde A y B son conjuntos difusos definidos en los rangos de “ u ” y “ v ” respectivamente.

Una regla expresa un tipo de relación entre los conjuntos A y B cuya función característica sería, $\mu_{A \rightarrow B}(x, y)$ y representa lo que conocemos como implicación lógica. La elección apropiada de esta función característica está sujeta a las reglas de la lógica proposicional. Como es bien sabido se puede establecer un isomorfismo entre la teoría de conjuntos, la lógica proposicional y el álgebra booleana que garantiza que cada teorema enunciado en una de ellas tiene un homólogo en las otras dos. La existencia de estos isomorfismos nos permitirá traducir las reglas difusas a relaciones entre conjuntos difusos y éstas a términos de operadores algebraicos con los que podremos trabajar.

2.2.3.1 Fundamentos de lógica proposicional

En la teoría de la lógica clásica una proposición sólo puede ser cierta o falsa, no admite términos medios; además las proposiciones pueden combinarse de muchas maneras, utilizando tres operaciones fundamentales:

- Conjunción ($p \wedge q$): las dos proposiciones son ciertas simultáneamente
- Disyunción ($p \vee q$): cualquiera de las dos proposiciones es cierta
- Implicación ($p \rightarrow q$): el cumplimiento o la verdad de una de las proposiciones tiene como consecuencia el cumplimiento de la otra; generalmente toma la forma de una regla si-entonces. La parte de la regla encabezada por el condicional si, “si u es A ”, es el antecedente o premisa de la regla, mientras que la parte encabezada por entonces, “entonces v es B ”, es el consecuente o conclusión de la regla.

También existe el operador

- Negación ($\sim p$) que invierte el sentido de la proposición.

La tabla de verdad (tabla 3) de estas operaciones que se pueden realizar entre las proposiciones es la que se muestra a continuación:

Tabla 3. Tabla de verdad de las principales operaciones lógicas

P	Q	$p \wedge q$	$P \vee q$	$p \rightarrow q$	$\sim p$
V	V	V	V	V	F
V	F	F	V	F	F
F	V	F	V	V	V
F	F	F	F	V	V

Algunas equivalencias de estos operadores con los operadores utilizados en teoría de conjuntos son las siguientes (tabla 4):

Tabla 4. Correspondencia entre operadores lógicos y de teoría de conjuntos

Lógica proposicional	Teoría de conjuntos
\wedge	\cap
\vee	\cup
\sim	$-$

con los operadores algebraicos (tabla 5):

Tabla 5. Correspondencia entre operadores lógicos y algebraicos

Lógica proposicional	Álgebra de Boole
V	1
F	0
\wedge	x
\vee	+
\sim	'
\leftrightarrow	=
P,q,r	a,b,c

Dónde la comilla ' representa el complementario y a,b,c son elementos del conjunto [0,1].

Una tautología se define como una proposición formada por la combinación de otras proposiciones y cuya verdad es independiente de la certeza o falsedad de las proposiciones que la forman. La tautología más importante para el ámbito en el que trabajamos es:

$$\cdot (p \rightarrow q) \leftrightarrow \sim.[p \wedge (\sim q).]$$

Que también puede ser expresada como:

$$\cdot (p \rightarrow q) \leftrightarrow \cdot (p) \cdot \vee q$$

La importancia de las tautologías reside en que nos permitirán expresar la función característica de la relación de implicación $p \rightarrow q$ en términos de las funciones características de p, q , $\sim p$ y $\sim q$.

En la teoría clásica proposicional existen dos importantes reglas de inferencia, el Modus Tollens y el Modus Ponens:

- El Modus Ponens o razonamiento directo puede resumirse de la siguiente forma:

Premisa 1: "x es A"

Premisa 2: "SI x es A, ENTONCES y es B"

Consecuencia: "y es B"

El Modus Ponens está asociado a la implicación "A implica B" ($A \rightarrow B$) y en términos de lógica proposicional se expresa: $(p \wedge (p \rightarrow q)) \rightarrow q$.

- El Modus Tollens o razonamiento inverso que puede resumirse de siguiente forma:

Premisa 1: "y es No B"

Premisa 2: "SI x es A ENTONCES y es B"

Consecuencia: "x es NO A"

En términos de lógica proposicional esto se expresa:

$$(\bar{q} \wedge (p \rightarrow q)) \rightarrow \bar{p}$$

El Modus Ponens es el utilizado en las aplicaciones de la lógica a la ingeniería ya que conserva la relación causa-efecto mientras que el Modus Tollens apenas se utiliza.

2.2.3.2 Implicación difusa

Al igual que para describir las nociones básicas de la teoría de conjuntos difusos podemos establecer un paralelismo con las de la teoría clásica de conjuntos, también los fundamentos de la teoría de la lógica difusa parten y toman los conceptos fundamentales de la lógica clásica.

Como ya hemos visto, en términos de teoría de lógica difusa la proposición "SI u es A, ENTONCES v es B" donde $u \in U$ y $v \in V$, tiene asociada una función característica $\mu_{A \rightarrow B}(x, y)$ que toma valores en el intervalo $[0,1]$. Es decir, cada una de las reglas o proposiciones if-then es a su vez un conjunto difuso con su función característica que mide el grado de verdad de la relación de implicación

entre x e y. Ejemplos de posibles funciones características asociadas, extraídas de aplicar las analogías entre operadores y la tautología antes mencionada, son, ver ecuaciones 12:

$$\begin{aligned}\mu_{A \rightarrow B}(x, y) &= 1 - \mu_{A \cap B}(x, y) = 1 - \min [\mu_A(x), 1 - \mu_B(y)] \\ \mu_{A \rightarrow B}(x, y) &= \max [1 - \mu_A(x), \mu_B(y)] \\ \mu_{A \rightarrow B}(x, y) &= 1 - \mu_A(x)(1 - \mu_B(y))\end{aligned}\tag{12}$$

En lógica difusa el Modus Ponens se extiende a lo que se llama Modus Ponens.

Generalizado y que puede resumirse de la siguiente forma:

Premisa 1: “u es A* ”

Premisa 2: “SI u es A ENTONCES v es B ”

Consecuencia: “v es B* ”

En dónde el conjunto difuso A* no tiene por qué ser necesariamente el mismo que el conjunto difuso A del antecedente de la regla y el conjunto difuso B* tampoco tiene por qué ser necesariamente el mismo que el conjunto difuso B que aparece en el consecuente de la regla.

Como vemos en lógica clásica una regla se ejecuta sólo si la primera premisa es exactamente la misma que el antecedente de la regla y el resultado de cada regla ejecutada es exacto al consecuente, en cambio en lógica difusa, una regla es ejecutada si existe un grado de similitud distinto de cero entre la primera premisa y el antecedente de la regla y el resultado de la ejecución de la regla es un consecuente que tiene un grado de similitud distinto de cero con el consecuente de la regla.

Así pues el Modus Ponens generalizado es una composición difusa en la que la primera relación difusa es el conjunto difuso A* y que puede expresarse, ver ecuación 13:

$$\mu_{B^*}(y) = \sup_{x \in A^*} [\mu_{A^*}(x) * \mu_{A \rightarrow B}(x, y)]\tag{13}$$

Teniendo en cuenta que, en las aplicaciones de la lógica difusa a la ingeniería la función característica de la implicación se construye con los operadores mínimos y

producto, que además de ser los más simples conservan la relación causa-efecto, tendremos dos opciones a elegir, ver ecuación 14:

$$\begin{aligned}\mu_{A \rightarrow B}(x, y) &= \min [\mu_A(x), \mu_B(y)] \\ \mu_{A \rightarrow B}(x, y) &= \mu_A(x) \bullet \mu_B(y)\end{aligned}\quad (14)$$

2.2.3.3 Reglas difusas

Una regla difusa base es un conjunto de reglas SI-ENTONCES que pueden ser expresadas de la siguiente forma:

R^m : SI u_1 es A_1^m y u_2 es A_2^m y u_p es A_p^m ,
ENTONCES v es B^m

Con $m=1,2,\dots, M$

Y donde A_i^m y B^m son conjuntos difusos en $U_i \subset \mathfrak{R}$ (números reales) y $V \in \mathfrak{R}$ respectivamente, $u = (u_1, u_2, \dots, u_n) \in U_1 \times U_2 \times \dots \times U_n$ y $v \in V$, y $x = x_1, x_2, \dots, x_n \in U$ e $y \in V$ son los valores numéricos concretos de u y v , también respectivamente.

Vemos que esta regla tiene además la particularidad de que es una regla multi antecedente; este tipo de reglas, que combina varias variables en el antecedente, es el más utilizado en el diseño de sistemas difusos. Un sistema difuso estará formado por varias reglas difusas base con diferentes consecuentes, ya que una regla con multi antecedente y multi consecuente siempre podrá ser descompuesta en un conjunto de reglas base con multi antecedente pero un solo consecuente.

Existen dos caminos para obtener el conjunto de reglas correspondiente a un conjunto de datos numéricos:

- Dejar que los datos establezcan los conjuntos difusos que aparecen en los antecedentes y los consecuentes
- Predefinir los conjuntos difusos para antecedentes y consecuentes y luego asociar los datos a esos conjuntos.

Para llegar a obtener el conjunto completo de reglas que modelan un problema se puede partir de considerar todas las combinaciones de reglas P_t que es posible establecer teóricamente, entre el número de antecedentes p y el número de conjuntos difusos de entrada A_p considerados para cada antecedente. Así, para cada consecuente, el número teórico de reglas posibles será:

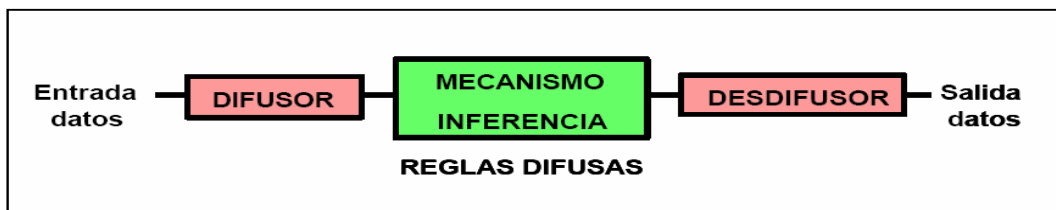
$$P_t = \prod_n A_n \text{ para } n = 1 \dots p;$$

Sin embargo entre estas P_t reglas teóricamente posibles para cada consecuente, habrá algunas que no tengan sentido físico y otras que no se ajusten a las características del problema a resolver. Se deberá pues seleccionar, de entre todas las reglas posibles, el conjunto de reglas más adecuadas al problema que se considera.

2.2.4 Diagrama de Bloques de un Sistema Basado en Técnicas de Lógica Difusa

El esquema de un sistema basado en técnicas de lógica difusa se presenta en la figura 5.

Figura 5. Esquema general de un sistema basado en lógica difusa



Está compuesto por los siguientes bloques

- **BLOQUE DIFUSOR:** bloque en el que a cada variable de entrada se le asigna un grado de pertenencia a cada uno de los conjuntos difusos que se ha considerado, mediante las funciones características asociadas a estos conjuntos difusos. Las entradas a este bloque son valores concretos de las

variables de entrada y las salidas son grados de pertenencia a los conjuntos difusos considerados.

- **BLOQUE DE INFERENCIA:** bloque que, mediante los mecanismos de inferencia que veremos más adelante, relaciona conjuntos difusos de entrada y de salida y que representa a las reglas que definen el sistema. Las entradas a este bloque son conjuntos difusos (grados de pertenencia) y las salidas son también conjuntos difusos, asociados a la variable de salida.
- **DESDIFUSOR:** bloque en el cual a partir del conjunto difuso obtenido en el mecanismo de inferencia y mediante los métodos matemáticos de desdifusión, se obtiene un valor concreto de la variable de salida, es decir, el resultado.

2.2.4.1 Mecanismos de Inferencia

Los mecanismos de inferencia son aquellos en los que se usan los principios de la lógica difusa explicados en el apartado 2.3 (inferencia difusa) para realizar un mapeo de los conjuntos difusos de entrada a los conjuntos difusos de salida. Cada regla es interpretada como una implicación difusa. Es decir, el bloque de inferencia es aquel en el cual se realiza la “traducción matemática” de las reglas difusas: estas reglas modelan el sistema pero para poder trabajar con ellas y extraer un resultado se debe de evaluar matemáticamente la información que reflejan. Como ya se ha mencionado anteriormente, las reglas más utilizadas para diseñar un sistema basado en lógica difusa toman la forma:

“Si u_1 es A_1 y u_2 es A_2 y u_3 es A_3 ENTONCES v es B “

Podemos decir que la implicación de cada regla (el conectivo lógico ENTONCES) es un conjunto difuso cuya función característica sería:

$\mu_{A \rightarrow B}(x, y)$ donde $x = (x_1, x_2, \dots, x_p)$

El resultado de evaluar el multi-antecedente también resultaría un conjunto difuso con función característica, ver ecuación 15:

$$\mu_{A_x}(x) = \mu_{x_1}(x_1) * \mu_{x_2}(x_2) * \dots * \mu_{x_p}(x_p) \quad (15)$$

Donde $*$ representa una t-norma.

Además, como ya hemos visto en el apartado anterior de inferencia difusa podemos asociar las reglas difusas al modus ponens generalizado: cada regla R_m determina un conjunto difuso B_m que es el resultado de la composición entre el conjunto difuso resultante de evaluar el antecedente y el conjunto difuso resultante de la implicación, es decir, $B_m = A_x \circ R_m$. La función característica asociada a estos conjuntos difusos de salida, como ya se ha visto, ver ecuación 16:

$$\mu_{B^m}(y) = \mu_{A_x} \circ R^m = \sup_{x \in A_x} [\mu_{A_x}(x) * \mu_{A \rightarrow B}(x, y)] \quad (16)$$

Y finalmente, el conjunto difuso de salida $B = A_x \circ [R_1, R_2, \dots, R_m]$, es el resultado de la agregación de todas las reglas que componen la regla base, es decir, de la combinación de los conjuntos difusos B_m resultantes de todas las reglas. Esta combinación se realiza generalmente mediante una t-conorma ya que, aunque no hay una razón teórica convincente que argumente que sea ésta la única manera de hacerlo (de hecho existen y funcionan los sistemas difusos aditivos y la adición no es una t-conorma), en aplicaciones a la ingeniería se obtienen resultados correctos y razonables usando este tipo de operadores.

Entonces concluimos que:

$$B = B_1 \oplus B_2 \oplus \dots \oplus B_M.$$

A la vista de lo anteriormente expuesto, queda claro que para procesar la información contenida en las reglas y obtener un resultado se deben elegir los operadores matemáticos que corresponden a los siguientes operadores lógicos:

- Los conectivos lógicos entre antecedentes, es decir, los operadores Y (AND, $*$), O (OR, \oplus) y NO (NOT, \neg):
"Si u_1 es A_1 y u_2 es A_2 y u_3 es A_3 ENTONCES v es B "
- El operador lógico ENTONCES (\rightarrow): implicación.

Además, esto debe realizarse para un conjunto de reglas mediante:

- La unión del resultado de todas las reglas (\oplus): agregación.

Vamos a ver cómo se realizan estas tres operaciones.

2.2.4.1.1 Conectivos lógicos entre antecedentes

Como ya se ha mencionado anteriormente, se puede considerar que la lógica clásica es un caso límite de la lógica difusa. En este caso límite, las tablas de verdad de los operadores lógicos clásicos y las de los operadores lógicos difusos deben coincidir. Considerando las tablas clásicas de verdad de los operadores AND, OR y NOT (Tabla 6):

Tabla 6. Tablas clásicas de verdad de los operadores and, or y not.

A	B	AND	A	B	OR
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	1	1	1

A	NOT A
0	1
1	0

Se establece una equivalencia entre la lógica clásica y la lógica difusa, teniendo en cuenta que, la equivalencia de estas tablas de verdad en lógica difusa debe preservar estos resultados y extender los demás valores a números reales entre 0 y 1. Como ya hemos visto anteriormente, una posibilidad para esta equivalencia consiste en la siguiente generalización de los conectivos lógicos:

- \cdot AND \Rightarrow t-norma
- \vee OR \Rightarrow t-conorma
- \neg NOT \Rightarrow operadores negación

En este caso las mismas tablas de verdad para la lógica difusa serían (Tabla 7):

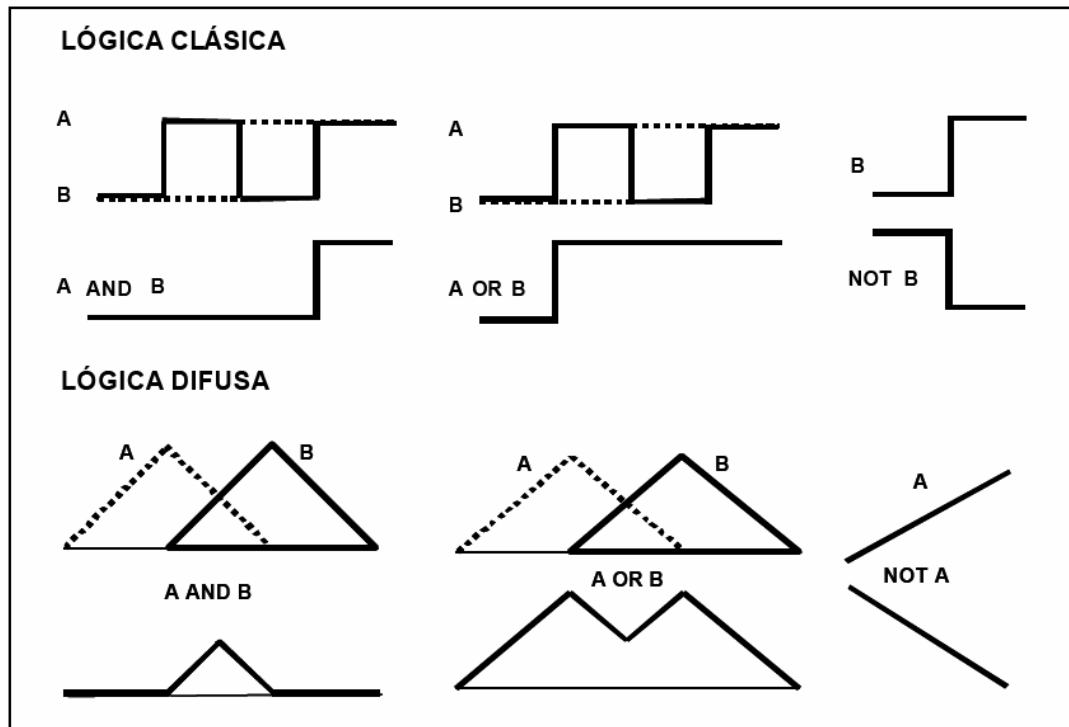
Tabla 7. Tablas de verdad para los operadores and, or y not difusos.

A	B	T-norma(A,B)	A	B	T-conor(A,B)
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	1	1	1

A	compl
0	1
1	0

La misma información expresada de manera gráfica sería la siguiente (figura 6):

Figura 6. Actuación de los operadores and, or y not, según la lógica clásica y la lógica difusa



Como ya hemos visto en apartados anteriores, la intersección (AND) de dos conjuntos difusos A y B se modela mediante una familia de operadores llamados t-normas, siendo el mínimo y el producto algebraico, los dos casos de t-normas más sencillos y utilizados.

Igualmente, la unión (OR) de dos conjuntos difusos se modela mediante otra familia de operadores llamados t-conormas o s-normas, cuyo representante más habitual es el máximo. Por último la negación (NOT) también tiene una familia de operadores para ser modelada, siendo la complementariedad aditiva uno de los más habituales.

Así, sustituyendo en cada regla los conectivos lógicos entre antecedentes por uno de sus operadores lógicos equivalentes, se podrá operar con los escalares que representa cada antecedente y obtener como resultado del multi-antecedente un escalar.

Considerando que el antecedente de una regla tiene la forma:

“Si x_1 es A_1 y x_2 es A_2 y x_3 es A_3si x_p es A_p”

- Si traducimos el operador lógico “y” mediante la operación mínimo (min), el antecedente será el resultado de, ver ecuación 17:

$$\min \{ \mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots, \mu_{A_p}(x_p) \} \quad (17)$$

- Si traducimos el operador lógico “y” mediante el producto (prod), el antecedente será el resultado de, ver ecuación 18:

$$\text{prod} \{ \mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots, \mu_{A_p}(x_p) \} \quad (18)$$

2.2.4.1.2 Implicación difusa. Superficies de implicación

El conectivo lógico ENTONCES representa la implicación entre antecedente y consecuente. Como ya hemos visto anteriormente, para que se cumpla la equivalencia entre lógica proposicional o clásica y lógica difusa, se debe traducir por una t-norma, cuyos representantes más utilizados en el ámbito de la ingeniería son el mínimo y el producto algebraico ya que preservan la relación causa-efecto y el sentido físico.

En cada regla, la implicación se realiza de la siguiente manera:

- Si se elige el operador mínimo para realizar la implicación:

$$\mu_{B_m^*}(y) = \min \{ \text{"escalar resultado del antecedente e"}, \mu_{B_m}(y) \}$$

Es decir, el escalar resultante del antecedente modifica el conjunto difuso de salida (su función característica) y pasa de ser B a ser B*. Gráficamente, la función característica del conjunto difuso de salida quedará truncada por el valor escalar que resulte del antecedente.

- Si se elige el operador producto

$$\mu_{B_m^*}(y) = \text{prod} \{ \text{"escalar resultado del antecedente"}, \mu_{B_m}(y) \}$$

Gráficamente, la función característica del conjunto difuso de salida quedará escalada según el valor del antecedente.

Cada regla define una superficie de implicación: para cada posible valor del antecedente, se puede obtener el conjunto difuso que resulta de realizar la implicación; la superposición de todos estos conjuntos difusos forma la superficie de implicación de esa regla. Todas las reglas que tienen el mismo consecuente, (elegido un operador para realizar la implicación), definirán la misma superficie de implicación ya que la superficie se construye para todos los valores que puede tomar el antecedente. Al particularizar la implicación para el antecedente resultante de cada regla, se obtiene el resultado concreto de esa regla, representado en la superficie de implicación por la intersección de esta superficie con el plano vertical que pasa por el valor del antecedente.

2.2.4.1.3 Agregación lógica

Una vez evaluadas todas las reglas y obtenidos los conjuntos difusos de salida modificados, hay que realizar la agregación de todas las reglas para obtener un resultado único de la actuación de todas ellas. Esta agregación es una unión lógica y una vez más para conservar la equivalencia entre lógicas clásica y difusa se traduce por una t-conorma: máximo o suma algebraica, obteniéndose así el conjunto difuso de salida, asociado a la variable de salida:

- Si se elige el operador máximo para realizar la agregación, el conjunto difuso de salida será, ver ecuación 19 :

$$\mu_B(y) = \max\{\mu_{B_1}(y), \mu_{B_2}(y), \dots, \mu_{B_m}(y)\} \quad (19)$$

- Si se elige el operador suma, el resultado de la agregación será, ver ecuación 20 :

$$\mu_B(y) = \text{sum}\{\mu_{B_1}(y), \mu_{B_2}(y), \dots, \mu_{B_m}(y)\} \quad (20)$$

2.2.4.2 Métodos de defusificación

El bloque desdifusor realiza la función contraria al difusor. El difusor tiene como entradas valores concretos de las variables de entrada y como salidas grados de pertenencia a conjuntos difusos (entre 0 y 1). La entrada al bloque desdifusor es el conjunto difuso de salida, resultado del bloque de inferencia y la salida es un valor concreto de la variable de salida. Para obtener, a partir del conjunto difuso de salida que resulta de la agregación de todas las reglas, un resultado escalar, se aplican métodos matemáticos. Ejemplos sencillos de algunos de estos métodos de cálculo son:

- **Método del máximo:** se elige como valor para la variable de salida aquel para el cual la función característica del conjunto difuso de salida es máxima. En general no es un método óptimo, ya que este valor máximo puede ser alcanzado por varias salidas.
- **Método del centroide:** utiliza como salida del sistema el centro de gravedad de la función característica de salida. Matemáticamente, ver ecuación 21 :

$$\bar{y} = (\int y\mu_B(y)dy) / (\int \mu_B(y)dy) \quad (21)$$

Es el método más utilizado en aplicaciones de la lógica difusa a la ingeniería ya que se obtiene una solución única, aunque a veces es difícil de calcular.

- **Método de la altura:** se calcula para cada regla el centro de gravedad del conjunto difuso de salida B_m y después se calcula la salida del sistema como la media ponderada, ver ecuación 22:

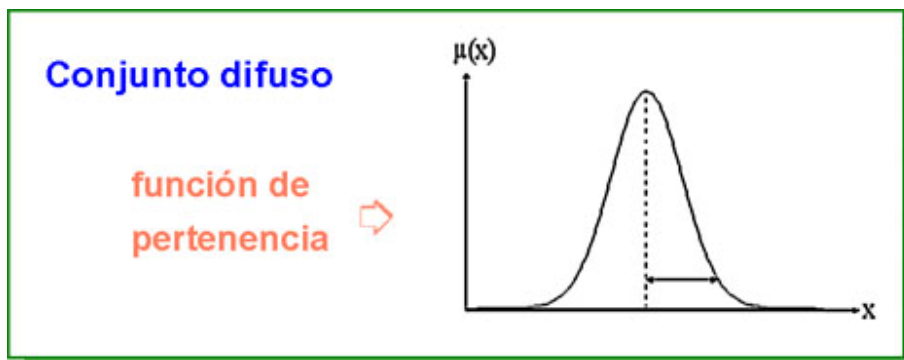
$$y_h = (\int \bar{y}_m \mu_{B_m}(\bar{y}_m) dy) / (\int \mu_{B_m}(\bar{y}_m) dy) \quad (22)$$

2.3. ALGUNAS TERMINOLOGIAS

Conjunto difuso o borroso

El concepto fundamental en que se basa la lógica difusa es el de conjunto difuso, un tipo de conjunto caracterizado porque los elementos del universo de discurso en el que está definido pueden pertenecer a él en un cierto grado, representado por una función de pertenencia.

Figura 7. Conjunto difuso o borroso

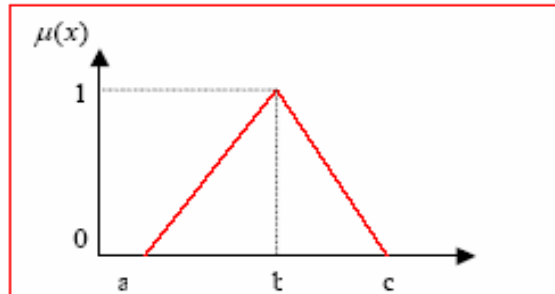


Función de pertenencia

Las funciones de pertenencia son asignadas a las etiquetas lingüísticas que forman un tipo de variable lingüística. Como ejemplos de funciones de pertenencia tenemos:

❖ Función de Pertenencia Triangular

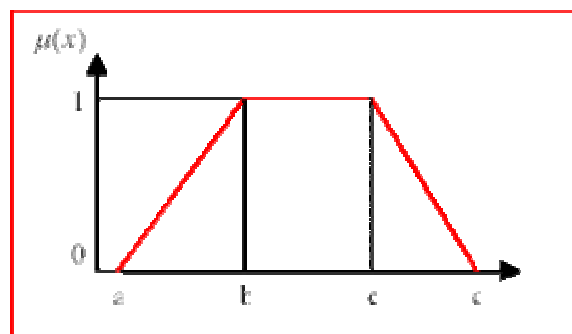
Figura 8. Función de Pertenencia Triangular.



Fuente: tesis de grado COBOR 1¹⁵

❖ Función de Pertenencia Trapezoidal

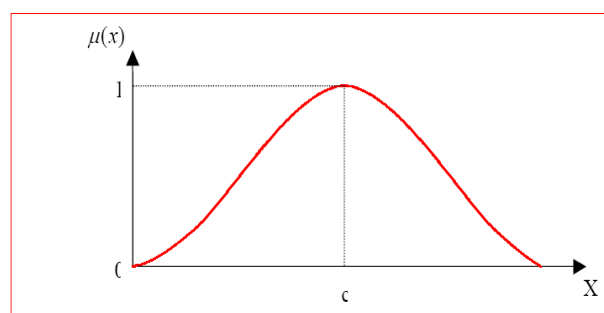
Figura 9. Función de Pertenencia Trapezoidal.



Fuente: tesis de grado COBOR 1

❖ Función de Pertenencia Tipo Gaussiana o Campana de Gauss

Figura 10. Función de Pertenencia Tipo Gaussiana.

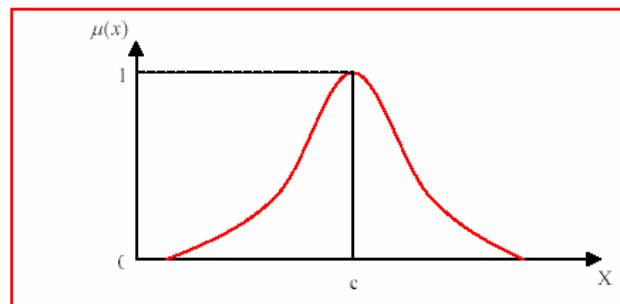


Fuente: tesis de grado COBOR 1

¹⁵ Reyes Figueroa, Juan Carlos, Landinez Cruz, Juan José. Software de Apoyo Para el Control Inteligente en Lógica Borrosa, 2004

❖ Función de Pertenencia Curva de Bell

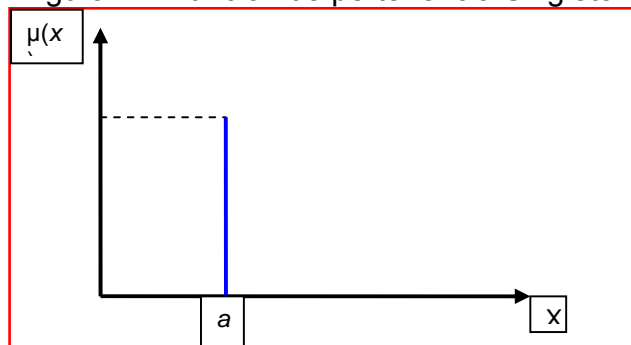
Figura 11. Función de Pertenencia Curva de Bell



Fuente: tesis de grado COBOR 1

❖ Función de pertenencia Singleton o Conjunto Borroso Unitario “Singleton”

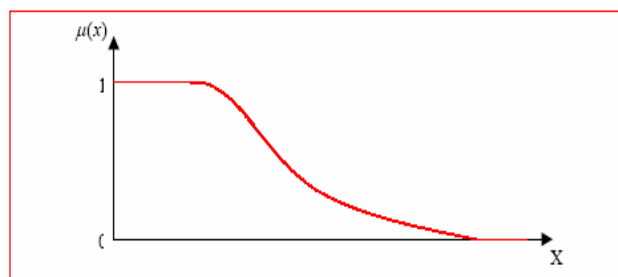
Figura 12. Función de pertenencia Singleton



Fuente: tesis de grado COBOR 1

❖ Función de Pertenencia Tipo Sigmoidal

Figura 13. Función de Pertenencia Tipo Sigmoidal



Fuente: tesis de grado COBOR 1

Variable lingüística

Una variable lingüística es una variable cuyos valores pueden expresarse mediante términos del lenguaje natural. Los diferentes términos o valores lingüísticos se representan mediante conjuntos difusos caracterizados por funciones de pertenencia dentro del universo de discurso.

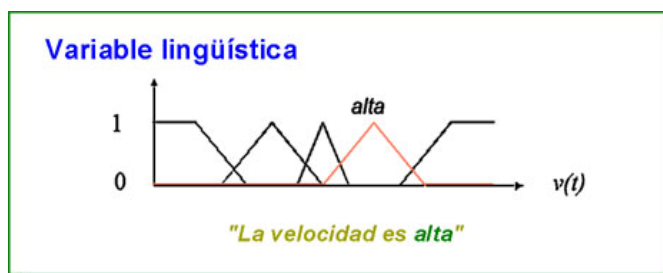
– A menudo queremos describir el estado de un objeto o fenómeno: Para ello usamos una Variable cuyo valor hace la descripción.

- Ejemplos: Temperatura, Volumen, Velocidad, etc.

– Una Variable lingüística admite que sus valores sean Etiquetas Lingüísticas, que son términos lingüísticos definidos como conjuntos difusos (sobre cierto dominio subyacente).

- Ejemplos: Temperatura “Cálida”, o “aproximadamente 25°C”.

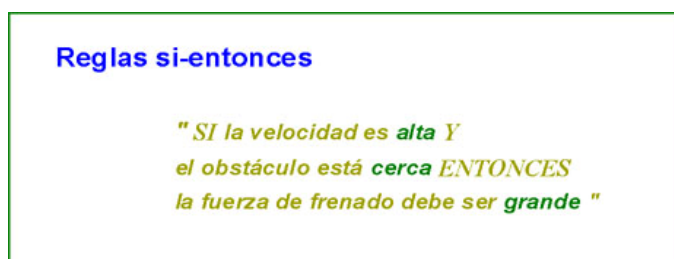
Figura 14. Variable lingüística



Base de reglas

Un sistema de inferencia basado en lógica difusa, o **sistema difuso**, está formado por un conjunto de reglas del tipo “si x es A entonces y es B ”, donde x e y son las variables del sistema, y A y B son términos lingüísticos como ‘alto’, ‘bajo’, ‘medio’, ‘positivo’, ‘negativo’, etc.

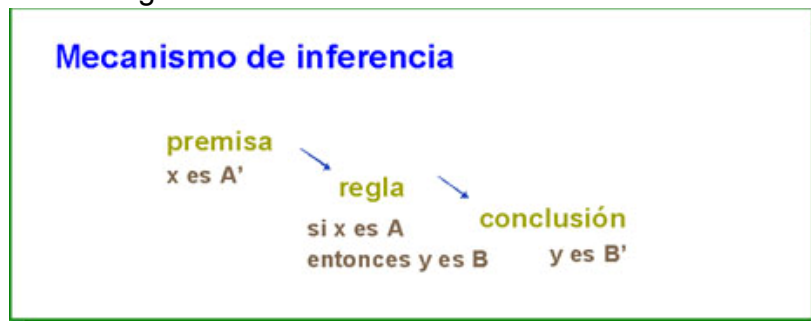
Figura 15. Base de reglas



Mecanismo de inferencia

Las técnicas de razonamiento utilizadas en lógica difusa son una generalización de los mecanismos de inferencia empleados en la lógica bivaluada tradicional, pero a diferencia de un sistema experto convencional, en un sistema difuso varias reglas pueden estar activas simultáneamente con diferentes grados de activación. Cuando varía una de sus entradas el sistema evalúa todas las reglas para inferir una conclusión o salida.

Figura 16. Mecanismos de inferencia



Control difuso

Un sistema difuso es un sistema basado en el conocimiento. Dicho conocimiento se estructura en un conjunto de reglas simbólicas del tipo "si-entonces", que utilizan términos del lenguaje natural para representar información vaga o imprecisa. Por otra parte, desde el punto de vista matemático, un sistema difuso es un sistema determinista que proporciona un mapeado no-lineal entre entradas y salidas. Las dos características anteriores son la base del control difuso, sin duda el campo de la Ingeniería donde la lógica difusa ha sido aplicada en más ocasiones y con mejores resultados

Motor de inferencia

El motor de inferencia del sistema difuso evalúa las diferentes reglas de la base de conocimiento. El grado de activación de cada regla es calculado a partir de los grados de activación de los antecedentes y de acuerdo con la interpretación de los diferentes conectivos empleados.

3. DESARROLLO DEL PROTOTIPO

3.1 FASE DE FORMACIÓN

Para la realización del proyecto se tuvo en cuenta las diferentes herramientas que serian de mayor utilidad para la realización del mismo. Para comenzar se hizo un estudio sobre UML, dado que este nos facilita hacer diagramas para la realización del prototipo debido a que es orientado a objetos. Luego se realizaron varios cursos sobre el lenguaje de aplicación en este caso sobre java, ya que es un lenguaje muy poderoso en la actualidad y además es software libre. Al final en el proceso de formación se realizaron cursos sobre Lógica Difusa dictados por los codirectores Dr. Jorge Luís Chacón y la ingeniera Marcela Jaimes y el profesor Msc. Manuel Guillermo Flórez que durante todo el proceso de desarrollo del prototipo estuvieron presentes para cualquier inquietud sobre el tema.

3.1.1 Análisis de la Información Recopilada

La recopilación y análisis de la bibliografía se hace con el objetivo de adquirir y reforzar conocimientos teóricos sobre Lógica Difusa, además de bibliografía relacionada con la tecnología a utilizar. Estas fuentes bibliográficas no tan solo se pudieron presentar en medio impreso, sino también en medio digital.

Bajo una previa investigación y clasificación de la bibliografía existente, se seleccionaron aquellos textos que son más relevantes y de mayor contenido de información para este proyecto, clasificándolos de la siguiente manera:

3.1.1.1 Planeación de Proyectos

- ✓ GÓMEZ, Flórez Luís Carlos. *Planeación de proyectos: Un enfoque para Ingeniería de Sistemas e Informática*. Primera edición, Bucaramanga, Ediciones UIS, 2001.

Este libro es una guía para la elaboración del plan de trabajo de grado; dio pautas para la elaboración de este plan.

- ✓ ECO, Umberto. *Cómo se hace una tesis: Técnicas y procedimientos de investigación, estudio y escritura*. Vigésimo cuarta edición, Barcelona, Editorial Gedisa S.A., 2001.

Este libro expone una metodología para elaborar tesis. Dio referencias sobre cómo disponer de los tiempos de trabajo, llevar a cabo la búsqueda bibliográfica, organizar el material encontrado y finalmente, estructurar el trabajo elaborado.

3.1.1.2 Metodologías de Desarrollo software

- ✓ PRESMAN, Roger. *Ingeniería del Software: Un enfoque práctico*. Cuarta edición, México, Editorial McGraw Hill, 1998.

Este libro contiene la teoría relacionada con las metodologías aplicadas en el diseño y desarrollo de software, que colaboraron a la hora de hacer la elección sobre la metodología que más se adapta al desarrollo del proyecto.

- ✓ FOWLER, Martin y SCOTT, Kendall. *UML gota a gota*. Primera edición, México, Editorial Pearson, 1999.

Proporciona las diversas técnicas de modelado de UML, describiendo la notación y semántica de cada uno de los diagramas. Sus fundamentos serán utilizados en el diseño de la herramienta software.

- ✓ MARTIN, Robert. UML para Programadores java. Primera edición, España, Editorial Pearson, Prentice Hall, 2004.

Es el primer libro que trata a UML como una herramienta para ayudar a los programadores a hacer más fácil su trabajo diario. Explica como tildar UML en proyectos reales.

- ✓ <http://www.willydev.net/descargas/Articulos/General/umlTotal.pdf>

Esta página de Internet, incluye una extensa documentación acerca de UML.

3.2 FASE DE DESARROLLO

3.2.1 Metodología

El modelo de desarrollo seleccionado para la elaboración del proyecto, adopta el enfoque metodológico empleado en el desarrollo software basado en modelos de ciclos de vida. Dicha metodología combina el diseño expresado a través del Lenguaje Unificado de Modelado, UML, con un modelo de ciclo de vida de desarrollo software de prototipado evolutivo.

El modelo de ciclo de vida de desarrollo software, se escogió después de haber realizado un análisis de los modelos de ciclos de vida más utilizados en el desarrollo de proyectos software, para lo que se tuvo en cuenta inicialmente las siguientes características del proyecto *SEXDP*:

- En la metodología de prototipado evolutivo se construyen prototipos o acercamientos al sistema que sirven para identificar los requisitos del cliente.

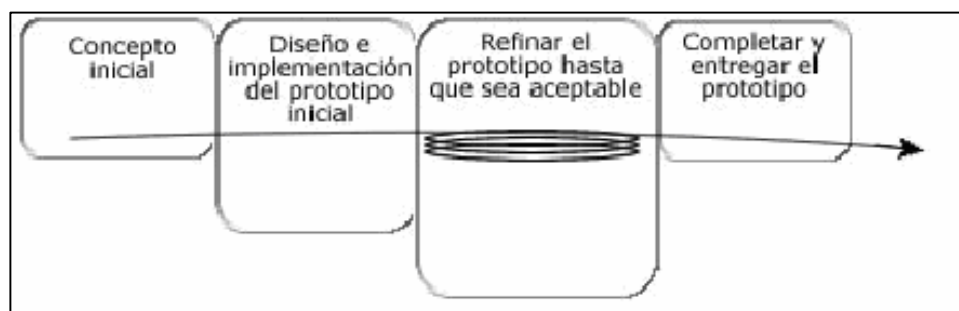
- Las especificaciones deben poder ajustarse a posibles modificaciones y adaptaciones que se hagan en el transcurso del desarrollo del proyecto.

Teniendo en cuenta estas características, los modelos no evolutivos deben ser descartados, pues estos modelos exigen un conocimiento exacto de los requerimientos de lo que se desea construir desde el inicio y requieren de una fuerte planificación, lo cual los hace muy flexibles para posteriores modificaciones en el diseño, así mismo se necesita un profundo conocimiento de las herramientas a utilizar. Es así, que el desarrollo de *SEXDP* se ve condicionado a la utilización de modelos evolutivos entre los que se ha escogido el prototipado evolutivo, pues este:

- No exige una fuerte planificación.
- Funciona con incrementos, los cuales arrojan como resultado un prototipo, para luego agregar nuevas funcionalidades hasta alcanzar lo que el cliente desea.
- Se adapta a la mayoría de las solicitudes de modificación de los clientes, a diferencia del modelo de entrega por etapas.
- Genera signos visibles de progreso.
- No es tan complicado ni requiere de tanta sofisticación como otros modelos, por ejemplo el modelo de desarrollo espiral.

La Figura 17, describe un esquema de flujo de trabajo a través de las fases que se ejecutan para obtener un producto final. El proceso parte de un concepto inicial donde se definen el análisis y especificaciones de los requerimientos iniciales.

Figura 17. Prototipado Evolutivo



3.2.2 Lenguaje Unificado de Modelado UML

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar los componentes de un sistema software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. Es de propósito general para el modelado orientado a objetos. También es un lenguaje de modelamiento visual que permite una abstracción del sistema y sus componentes.

UML no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguajes de programación, así como construir modelos por ingeniería inversa a partir de programas existentes.

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.

El rol de los modelos se puede sintetizar en:

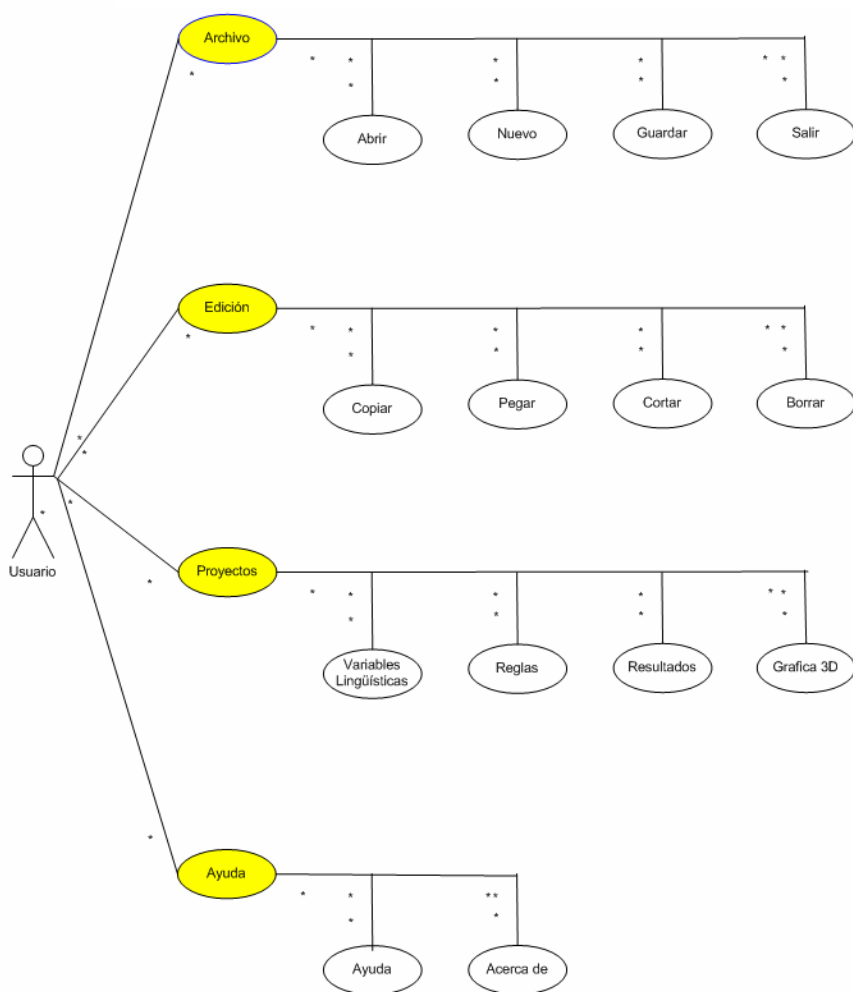
- Ayudar a entender la información, la función y el comportamiento del sistema, haciendo por tanto más fácil y sistemática la tarea de analizar requerimientos.
- Convertirse en el punto de comparación entre lo logrado y lo planificado.
- Fundamentar el diseño, proporcionando una representación lógica, de la implementación.

3.2.3. Primer Prototipo

Para el desarrollo del primer prototipo de la interfaz gráfica se implementa todas opciones necesarias que debe tener una interfaz para el usuario. Con esta interfaz

se desarrolla la integración con los diferentes módulos que se realizaron, logrando así una adecuada presentación para el manejo de las variables y las reglas. A continuación se presenta un diagrama de los casos de uso que se desarrollo para esta interfaz.

Figura 18. Diagrama casos de uso de la interfaz Gráfica



Estos son los casos de uso básicos que debe tener una interfaz gráfica (ver figura 18) y como resultado de este diagrama se obtuvo la siguiente ventana con sus correspondientes funciones realizadas en Java. Se presentara una breve explicación de cada menú.

Figura 19. Ventana principal de la interfaz gráfica (primer prototipo)



Menú Archivo:

Abrir: le permite al usuario abrir un proyecto ya realizado el cual podrá ser modificado.

Nuevo: el usuario tendrá la opción de crear un nuevo proyecto.

Guardar: además podrá guardarlo en cualquier parte del disco local.

Salir: el usuario saldrá de la aplicación y antes de salir se le preguntará si desea guardar cambios o no.

Menú Edición:

En este menú se encuentra lo más básico que puede tener una interfaz gráfica las cuales son: copiar, cortar, pegar y borrar conjuntos borrosos.

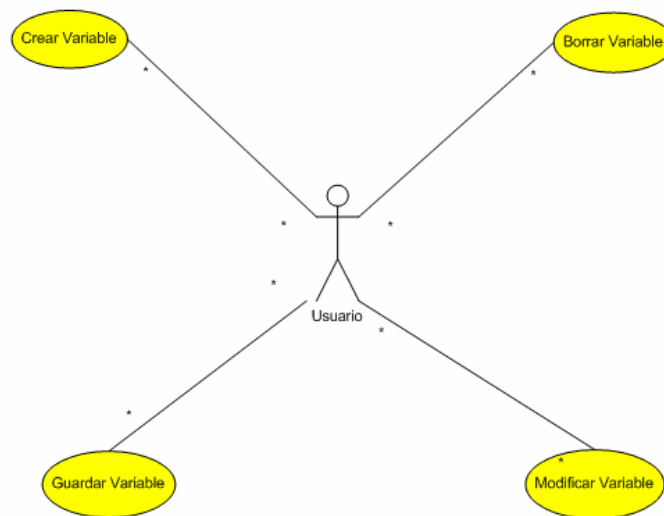
Menú proyectos:

La opción del menú más importante es la de proyectos, ya que esta es la que nos permitirá interactuar con todos los módulos que se realizaron. Para nuestro caso solo presentare el módulo de los Parámetros, que es el módulo correspondiente a los objetivos planteados, en la siguiente versión del prototipo se presentara el software con todos los módulos correspondientes.

Menú ayuda:

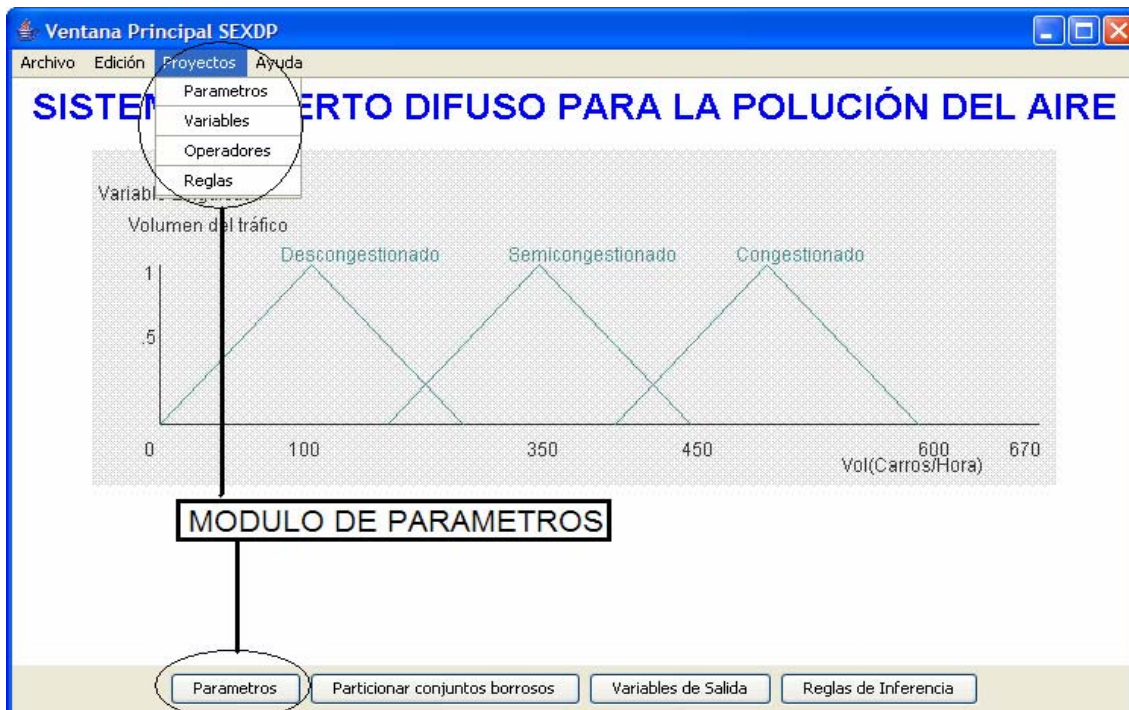
Para terminar con la barra de menús aparece la opción de ayuda, la cual nos permitirá acceder a un breve manual de ayuda que nos guiara paso a paso para el correcto funcionamiento del prototipo, también esta la opción de “Acerca de” que corresponde a los créditos de las personas que realizamos este proyecto.

Figura 20. Diagrama casos de uso del módulo Parámetros



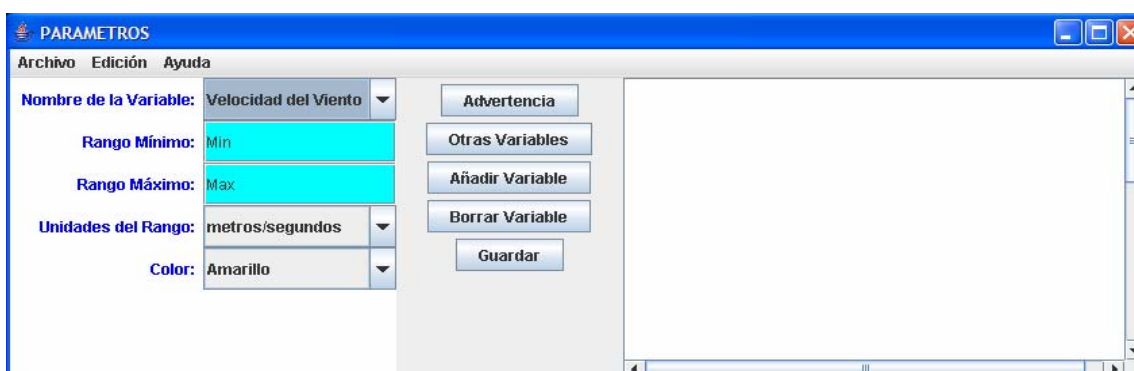
Este es uno de los módulos principales del prototipo ya que aquí es donde se empezara a ingresar los datos para los conjuntos borrosos.

Figura 21. Ventana principal para ingresar al módulo Parámetros.



Existen dos formas para ingresar al módulo de los parámetros, la primera mediante la barra de menús en la parte superior en la opción proyectos o en la barra de botones que aparece en la parte inferior de la pantalla, que al seleccionarla nos aparecerá la siguiente ventana.

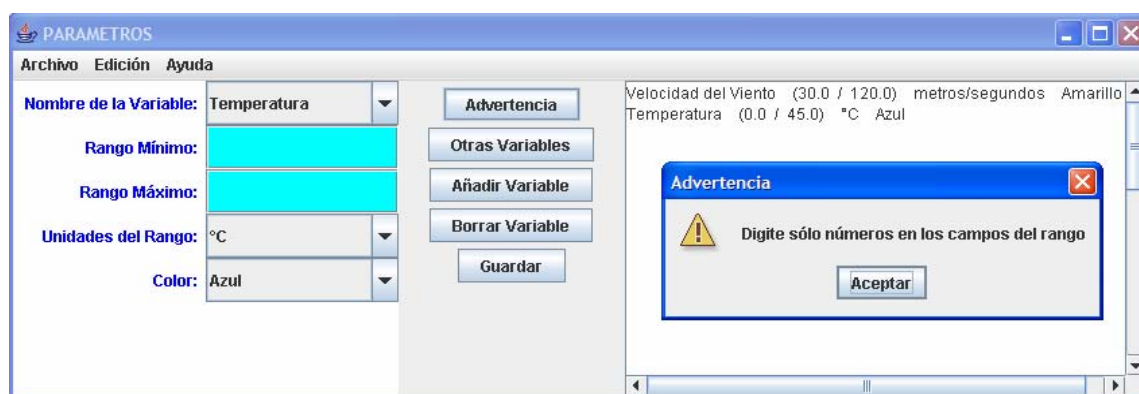
Figura 22. Ventana principal módulo parámetros. (Primer prototipo)



En este módulo se ingresara el nombre de las variables que vamos a estudiar, también se ingresara el rango y las unidades de dicha variable y un color para distinguirla de las demás. Además tiene la opción de agregar nuevas variables con sus respectivas unidades y guardarlas en un archivo de texto.

Todas las opciones del menú superior como Archivo, edición y ayuda están disponibles para un correcto funcionamiento del módulo, además los campos del rango están validados para la entrada correspondiente de los valores.

Figura 23. Módulo parámetros opciones validadas



3.2.4. Segundo Prototipo

Para el segundo prototipo se crea una ventana principal o tab, la cual muestra las variables de entrada que están por defecto (vede), en el centro el tipo de modelo que se esta trabajando que es Mandani y al lado derecho se muestra las variables de salida por defecto (azul), en la parte inferior aparece información sobre el prototipo como el nombre del ejemplo, el tipo y métodos (ver figura 26).

Figura 24. Ventana principal



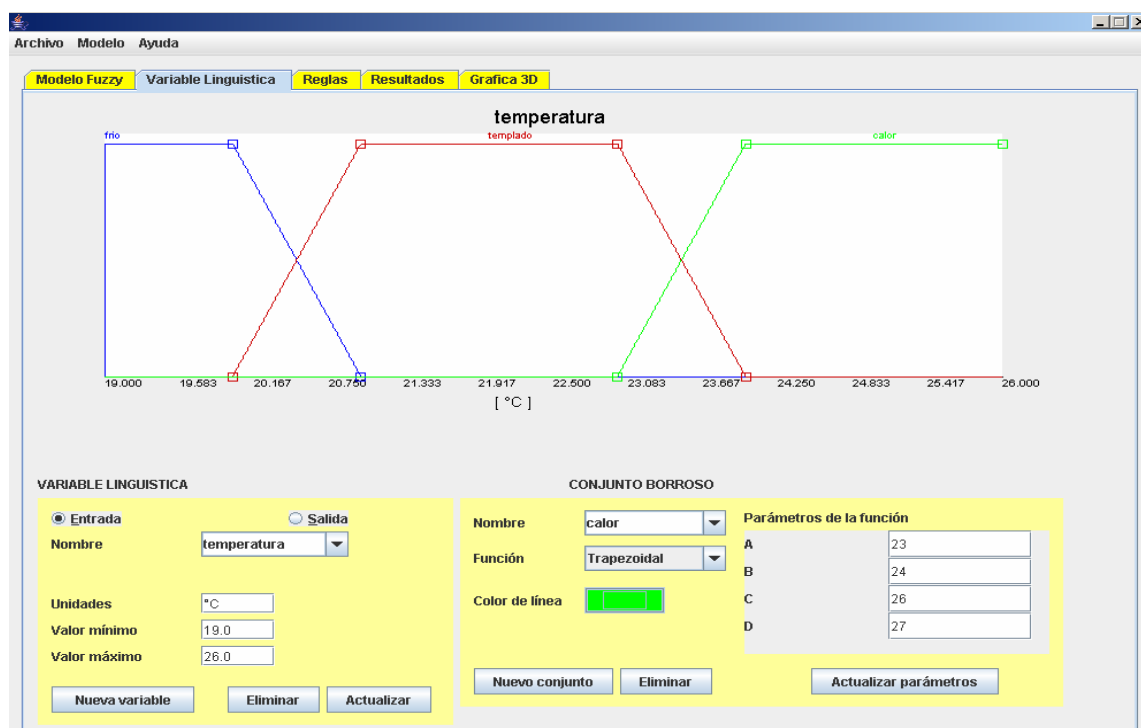
Para el prototipo se tuvo en cuenta la integración y la relación que había entre la Interfaz Gráfica y los respectivos módulos. Se vio la necesidad de unificar los módulos de parámetros y variables, debido a las dificultades que había para su relación, por consiguiente se realiza una sola interfaz para estos dos módulos (ver figura 25). Con respecto a los otros módulos no hubo problema para la integración.

Figura 25. Módulo de parámetros y variables lingüísticas.

Figura 26. Módulo de parámetros y variables lingüísticas integradas a la interfaz principal.

Para la interfaz se intento hacer lo más amigable para usuario, solamente el usuario tiene que tener un poco de conocimientos en el área de la contaminación del aire y sobre lógica difusa, y si no los tiene hay un manual que le ayudara a utilizar el prototipo sin ninguna dificultad, este manual se encuentra en el menú ayuda.

Figura 27. Ejemplo1 del módulo de parámetros y variables lingüísticas



En la figura 27 se muestra un ejemplo de una variable lingüística con sus respectivos conjuntos borrosos. Para este ejemplo se utilizó la variable lingüística de la “temperatura” con sus respectivas unidades y su valor mínimo y máximo, al lado derecho se observa el conjunto borroso “frio”, a continuación se selecciona el tipo de función que en este caso es trapezoidal y luego un color para poderla diferenciar de los otros conjuntos borrosos y por último se le dan los parámetros que definen a este conjunto borroso. A continuación se muestra otro ejemplo de la velocidad del viento como variable lingüística

Figura 28. Ejemplo2 del módulo de parámetros y variables lingüísticas.



4. INTEGRACIÓN DE LA INTERFAZ GRÁFICA DEL USUARIO CON EL MOTOR DE INFERENCIA

Para realizar la integración, de los módulos con la interfaz gráfica de usuario, hubo varios inconvenientes, debido a los diferentes métodos de programación que teníamos los autores del prototipo, por lo cual se vio la necesidad de realizar una sola ventana para facilitar la amigabilidad del prototipo y así dejar una interfaz de usuario más fácil de usar por parte de los usuarios finales. Los módulos que se integraron fue el de los parámetros y el de las variables, dejando así una sola ventana para analizar las gráficas y las variables de entrada y salida.

Esto se logro con el desarrollo del ultimo prototipo ya que en los primeros nunca se intento de hacer una integración de lo que se había hecho. Con respecto al módulo de los operadores y al de las reglas no hubo ninguna dificultad, debido a que se realizo un excelente trabajo de equipo para así lograr el resultado que esperábamos.

Con respecto a la integración hubo la necesidad de trabajar en un solo diseño de presentación, y para esto la mejor opción de trabajo era con la herramienta llamada NetBeans 5.5, ya que esta presenta una excelente presentación y facilidad para trabajar con los objetos deseados. NetBeans, distribuido gratuitamente por la propia Sun (Creadora de Java), es un entorno completísimo al que no le falta absolutamente nada, pero tiene un problema enorme: Está hecho en Java. Esto está bien para demostrar la potencia de java a la hora de realizar aplicaciones de una calidad tan excepcional como NetBeans, pero precisamente, al estar hecho en Java sufre del mayor de los inconvenientes: Ocupa mucho espacio y es extremada e intolerablemente lento. Esta herramienta se encuentra disponible en la pagina Web <http://www.netbeans.org>, y también es necesario una librería llamada Java 3D, la cual es indispensable para realizar la gráfica en tres dimensiones. Otra herramienta que se utilizó fue JCreator pro, esta tiene ciertas

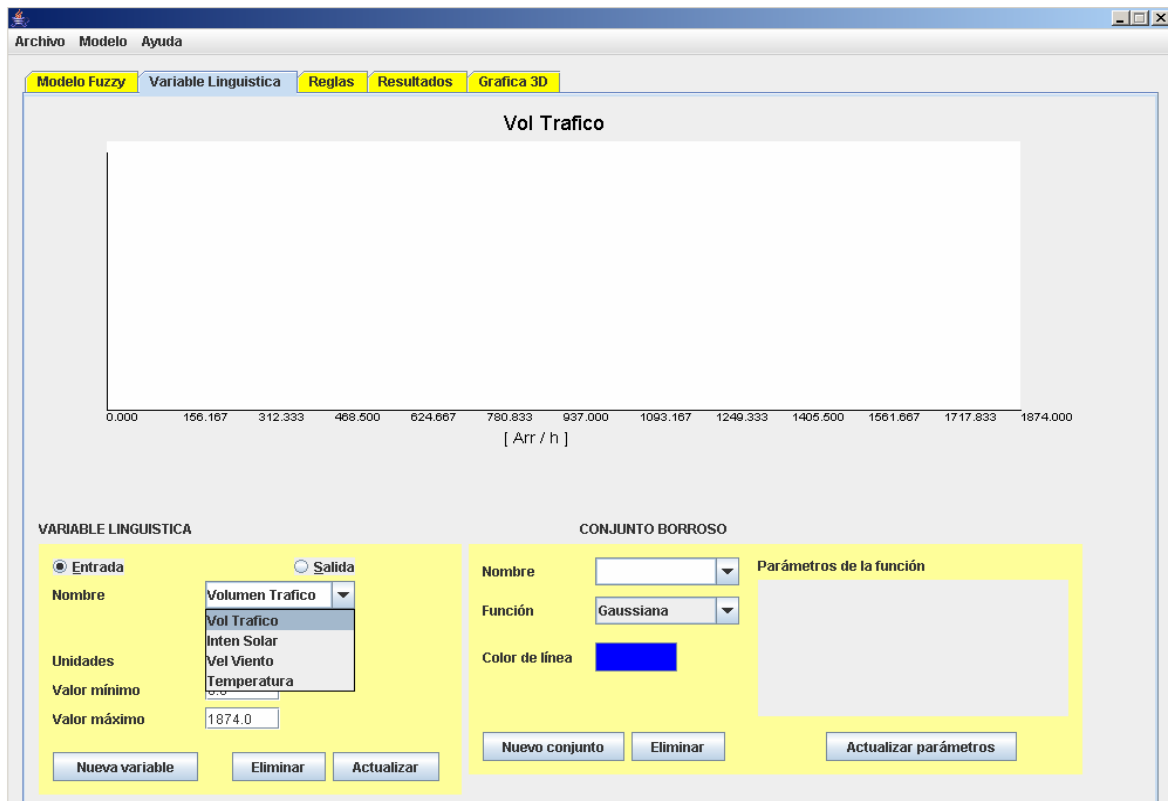
ventajas como Implementación de colores en código, numeración de líneas, sencillez, robusto, rápido, come pocos recursos, fácil manejo. Jcreator seguramente sea una utilidad que poca gente conozca ya que es una herramienta para la programación en Java. Simplemente por la sencillez de la misma y que esta echa explícitamente para eso, para la programación en Java. JCreator es un entorno de desarrollo integrado (IDE) muy peculiar, ya que se basa en una filosofía radicalmente distinta al resto de IDEs para Java, y es la sencillez.

Otro aspecto y se cree que es el mas importante es la realización de los diagramas de clases en el Lenguaje Unificado de Modelado o UML, ya que estos diagramas facilitan las representaciones a partir de las cuales los desarrolladores podrán trabajar y se tendrá un mejor desempeño a la hora en que se integre los programas finales. Esta es una de las herramientas más emocionantes en el mundo actual del desarrollo de sistemas. Esto se debe a que permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas.

4.1 Descripción de la Interfaz Gráfica Principal

Para realizar la descripción de la Interfaz Gráfica de Usuario se debe explicar cada uno de los métodos que la conforman. A continuación se muestra la Figura 30 de la IGU y los métodos principales que se usaron para realizar la integración con los demás módulos.

Figura 29. Ventana principal de la variable lingüística y conjuntos borrosos



Para comenzar con el programa iniciamos con el botón de nueva variable, este nos permitirá ingresar los datos del nombre, unidades y el rango. El método utilizado es el siguiente:

➤ Botón Nueva Variable

```
private void btnAgregarVariableActionPerformed(java.awt.event.ActionEvent evt) {
    var=null;
    conjuntoEdicion=null;
    VariableLinguistica v;
    v=new VariableLinguistica("Variable"+variables.size(),0,1);
    variables.add(v);
    variablesEntrada.add(v);
}
```

```

        cmbVariableLinguistica.addItem(v);
        cmbVariableLinguistica.setSelectedItem(v);
        rbtnEntrada.setSelected(true);
        repaint();
    }

```

Cuando queremos eliminar una Variable Lingüística solo tenemos que presionar el botón Eliminar, el cual borrara la variable que este actualmente seleccionada. El método usado para realizar esta función es:

➤ Botón Eliminar Variable Lingüística

```

private void btnEliminarVariableActionPerformed(java.awt.event.ActionEvent evt) {
    if(rbtnEntrada.isSelected()==true){
        cmbVariableLinguistica.removeItem(var);
        variablesEntrada.remove(var);
    }else{
        cmbVariableLinguistica.removeItem(var);
        variablesSalida.remove(var);
    }
    conjuntoEdicion=null;
    var=null;
    inicializarParametros();
    repaint();
}

```

Cuando se termina de ingresar los datos de la variable lingüística se procede a presionar el botón de Actualizar, el cual me graficara el plano cartesiano y pondrá los respectiva datos de la variable creada.

➤ Botón Actualizar

```
public void actualizarVariable(){
    var.xFinalRango=Float.parseFloat(txtValorMaximo.getText());
    var.xInicialRango=Float.parseFloat(txtValorMinimo.getText());
    var.nombreVariable=cmbVariableLinguistica.getSelectedItem().toString();
    var.unidades=txtUnidades.getText();
    for (ConjuntoBorroso c: var.conjuntoBorroso){
        c.fnpFuncionPertenencia.setXInicialRango(var.xInicialRango);
        c.fnpFuncionPertenencia.setXFinalRango(var.xFinalRango);
    }
    actualizarConjunto();
}

public void actualizarConjunto(){
    if(conjuntoEdicion!=null && cmbConjuntoBorroso.getSelectedIndex()>=0){
        conjuntoEdicion.nombreConjuntoBorroso=cmbConjuntoBorroso.getSelectedItem().toString();
        conjuntoEdicion.actualizarParametros(cmbConjuntoBorroso.getSelectedItem().toString(),
        btnColorLinea.getBackground());
        repaint();
    }
}
```

Ahora se pasa al panel del conjunto borroso. Aquí al igual que en el panel de variable lingüística hay que seleccionar el botón de nuevo conjunto para poder seleccionar el nombre del conjunto borroso, el tipo de función (Gaussiana, Campana de Bell, Sigmoidal, Trapezoidal y Triangular) y por último el color que llevará la gráfica de dicha función.

➤ Botón Nuevo conjunto

```
private void btnAgregarConjuntoActionPerformed(java.awt.event.ActionEvent evt) {
    if(var!=null)
    {
```

```

conjuntoEdicion=null;
ConjuntoBorroso c = new ConjuntoBorroso("Bajo "
                                         +cmbConjuntoBorroso.getItemCount(),
                                         getTipoFuncionPertenencia());
c.colorLinea=btnColorLinea.getBackground();
pnParametros.removeAll();
pnParametros.add(c.fnpFuncionPertenencia.getPanelParametrosFuncion());
var.conjuntoBorroso.add(c);
cmbConjuntoBorroso.addItem(c);
cmbConjuntoBorroso.setSelectedItem(c);
System.out.println("conjunto seleccionado "+cmbConjuntoBorroso.getSelectedIndex());
conjuntoEdicion=c;
repaint();
}
}

```

El método usado para eliminar un conjunto borroso es similar al de eliminar variables lingüísticas. El método es el siguiente:

➤ Botón Eliminar Conjunto Borroso

```

private void btnEliminarConjuntoActionPerformed(java.awt.event.ActionEvent evt) {
    //
    var.conjuntoBorroso.remove(conjuntoEdicion);
    cmbConjuntoBorroso.removeAllItems();
    for(ConjuntoBorroso c:var.conjuntoBorroso){
        cmbConjuntoBorroso.addItem(c);
    }
    repaint();
}

```

finalmente los métodos usados para actualizar los conjuntos borrosos son los siguientes:

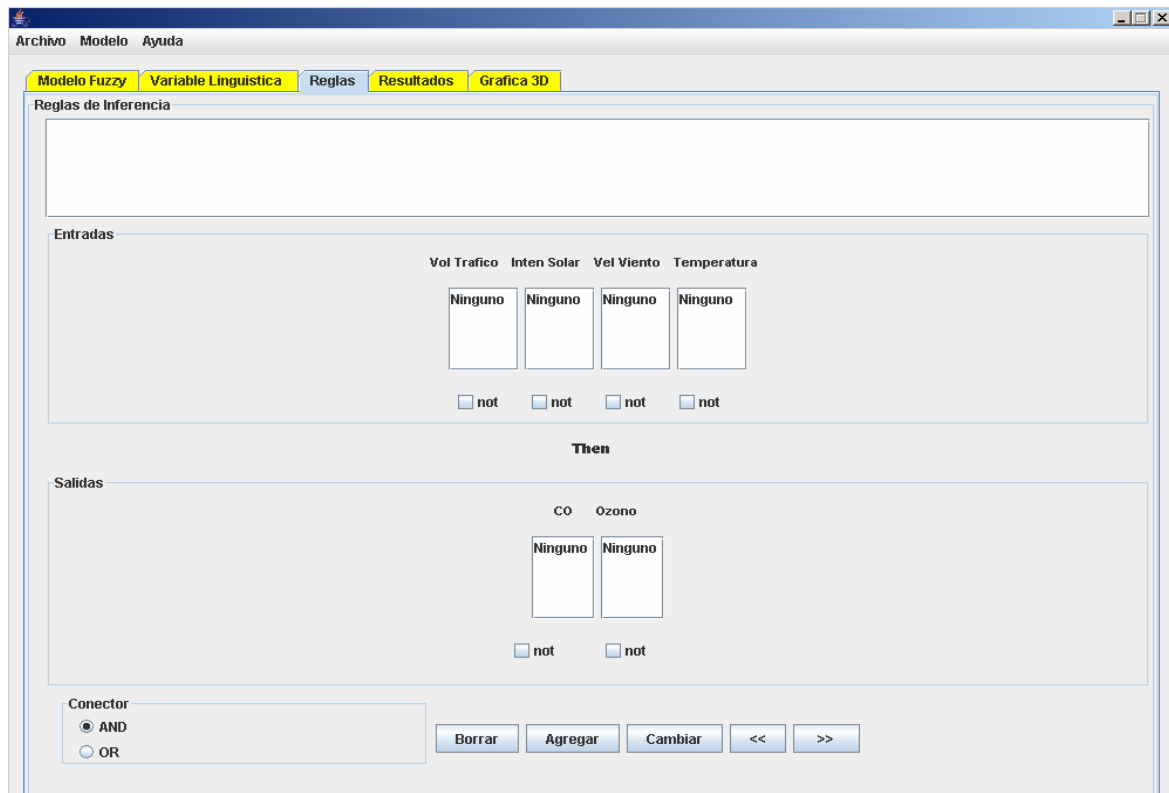
➤ Botón actualizar parámetros

```
private void btnActualizarParametrosActionPerformed(java.awt.event.ActionEvent evt) {  
    conjuntoEdicion.nombreConjuntoBorroso=cmbConjuntoBorroso.getSelectedItem().toString();  
        actualizarConjunto();  
    }  
  
    public void actualizarConjunto(){  
        if(conjuntoEdicion!=null && cmbConjuntoBorroso.getSelectedIndex()>=0){  
  
            conjuntoEdicion.nombreConjuntoBorroso=cmbConjuntoBorroso.getSelectedItem().toString();  
                conjuntoEdicion.actualizarParametros(cmbConjuntoBorroso.getSelectedItem().toString(),  
btnColorLinea.getBackground());  
        }  
    }    repaint();  
    }  
}
```

4.2 Descripción de la ventana de la Reglas

En este panel estarán todas las reglas necesarias que el usuario ingresará manualmente aplicando cada conector. A continuación se presenta una descripción de los respectivos métodos y constructores que implementaron para realizar el módulo.

Figura 30. Ventana principal de las reglas



El constructor para esta ventana es el siguiente:

```
public PanelEntradas_1(ArrayList<VariableLinguistica>
variableEntrada,ArrayList<VariableLinguistica>    variableSalida,    ArrayList<ReglaInferencia>
rglInferencia ) {
    this.variableEntrada=variableEntrada;
    this.variableSalida=variableSalida;
    this.rglInferencia=rglInferencia;
    initComponents();
    iniciarPanel();
    union="AND";
}
```

Los métodos más representados son:

```

public void actualizarModelo(){
    lstReglas.setModel(new javax.swing.AbstractListModel() {
        public int getSize() {
            return rgllInferencia.size();
        }
        public Object getElementAt(int i) {

            return rgllInferencia.get(i);
        }
    });
    lstReglas.setSelectedIndex(rgllInferencia.size()-1);
}

private void borrarRegla(java.awt.event.MouseEvent evt) {
    int idRegla= lstReglas.getSelectedIndex();
    if(idRegla<0){
        if(rgllInferencia.size(>0){
        }
        else{
            rgllInferencia.add(new ReglaInferencia(variableEntrada, variableSalida));
        }
        lstReglas.setSelectedIndex(rgllInferencia.size()-1);
    }
    idRegla=lstReglas.getSelectedIndex();
    rgllInferencia.remove(lstReglas.getSelectedIndex());
    try{
        lstReglas.setModel(new javax.swing.AbstractListModel() {
            public int getSize() {
                return rgllInferencia.size();
            }
            public Object getElementAt(int i) {

                return rgllInferencia.get(i);
            }
        });
    }
}

```

```

catch(Error e){
    System.out.println("Error "+e);
}
lstReglas.setSelectedIndex(reglas.size()-1);
}

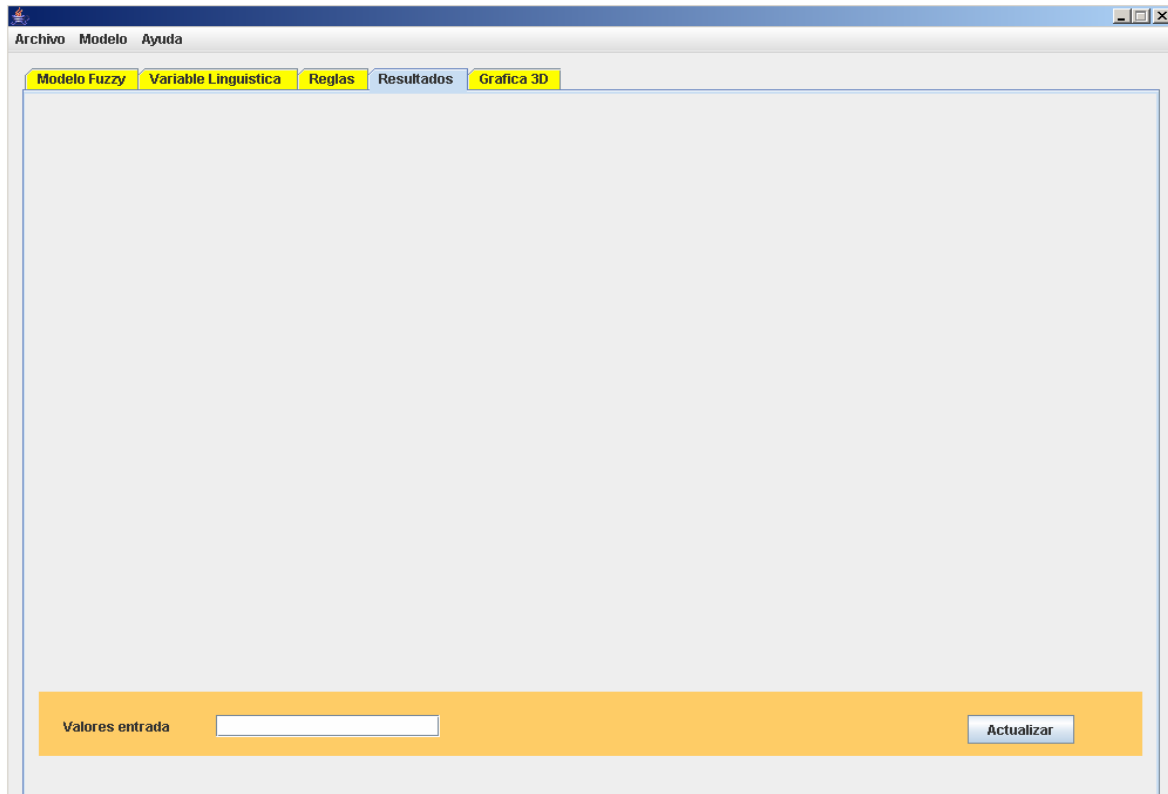
private void guardarRegla(java.awt.event.MouseEvent evt) {
    if(((JButton)evt.getComponent()).getActionCommand().equals("Guardar") &&
        lstReglas.getSelectedValue()!=null){
        ReglaInferencia nuevaRegla=new ReglaInferencia
        ((ReglaInferencia)lstReglas.getSelectedValue(),variableEntrada, variableSalida);
        rgllInferencia.add(nuevaRegla);
        lstReglas.setModel(new javax.swing.AbstractListModel() {
            public int getSize() {
                return rgllInferencia.size();
            }
            public Object getElementAt(int i) {
                return rgllInferencia.get(i);
            }
        });
        lstReglas.setSelectedIndex(reglas.size()-1);
    }
    guardarArchivo(evt);
}

```

4.3 Descripción de la ventana de resultados

En esta ventana se presentan todas las variables de entrada y salida del sistema y además las muestra en forma gráfica 2D. Después de la figura 32 que esta a continuación se mostrará los constructores y métodos usados.

Figura 31. Ventana principal de los resultados



Constructores usados para el desarrollo del módulo:

```
public PanelResultados(ArrayList<VariableLinguistica> varIn,  
                        ArrayList<VariableLinguistica> varOut,  
                        ArrayList<ReglaInferencia> reglas) {  
    this.variablesEntrada=varIn;  
    this.variablesSalida=varOut;  
    this.reglasInferencia=reglas;  
    initComponents();  
    inicializarControles();  
}  
public PanelResultados(){  
    initComponents();  
    inicializarControles();  
}
```

Métodos usados para el desarrollo:

```
public void inicializarControles(){
    xLinea=new ArrayList<Integer>();
    int fila=-1;
    int columna=-1;
    int x=0;
    if(reglasInferencia.size())>0){
        columna=-1;
        pasoX=(int)((xFinalGrafico-xInicialGrafico)
            /reglasInferencia.get(0).getRegla().size());
        espacio=(int)(pasoX*0.1);
        for(ConjuntoBorroso c: reglasInferencia.get(0).getRegla()){
            columna++;
            if(columna<variablesEntrada.size()){
                x=(int)(xInicialGrafico+columna*pasoX);
                xLinea.add(x);
            }
        }
    }
}

private void initComponents() {
    pnVariables = new javax.swing.JPanel();
    btnActualizar = new javax.swing.JButton();
    addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {
        public void mouseDragged(java.awt.event.MouseEvent evt) {
            formMouseDragged(evt);
        }
        public void mouseMoved(java.awt.event.MouseEvent evt) {
            formMouseMoved(evt);
        }
    });
    addMouseListener(new java.awt.event.MouseAdapter() {
        public void mousePressed(java.awt.event.MouseEvent evt) {
            formMousePressed(evt);
        }
    });
}
```

```

        public void mouseReleased(java.awt.event.MouseEvent evt) {
            formMouseReleased(evt);
        }
    });
    pnVariables.setBackground(new java.awt.Color(255, 204, 102));
    btnActualizar.setText("Actualizar");
    org.jdesktop.layout.GroupLayout pnVariablesLayout = new
org.jdesktop.layout.GroupLayout(pnVariables);
    pnVariables.setLayout(pnVariablesLayout);
    pnVariablesLayout.setHorizontalGroup(
        pnVariablesLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
        .add(org.jdesktop.layout.GroupLayout.TRAILING,
pnVariablesLayout.createSequentialGroup()
        .addContainerGap(384, Short.MAX_VALUE)
        .add(btnActualizar)
        .add(77, 77, 77))
    );
    pnVariablesLayout.setVerticalGroup(
        pnVariablesLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
        .add(org.jdesktop.layout.GroupLayout.TRAILING,
pnVariablesLayout.createSequentialGroup()
        .addContainerGap(130, Short.MAX_VALUE)
        .add(btnActualizar)
        .addContainerGap())
    );
    org.jdesktop.layout.GroupLayout layout = new org.jdesktop.layout.GroupLayout(this);
    this.setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
        .add(layout.createSequentialGroup()
        .addContainerGap()
        .add(pnVariables, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addContainerGap())
    );
    layout.setVerticalGroup(

```

```

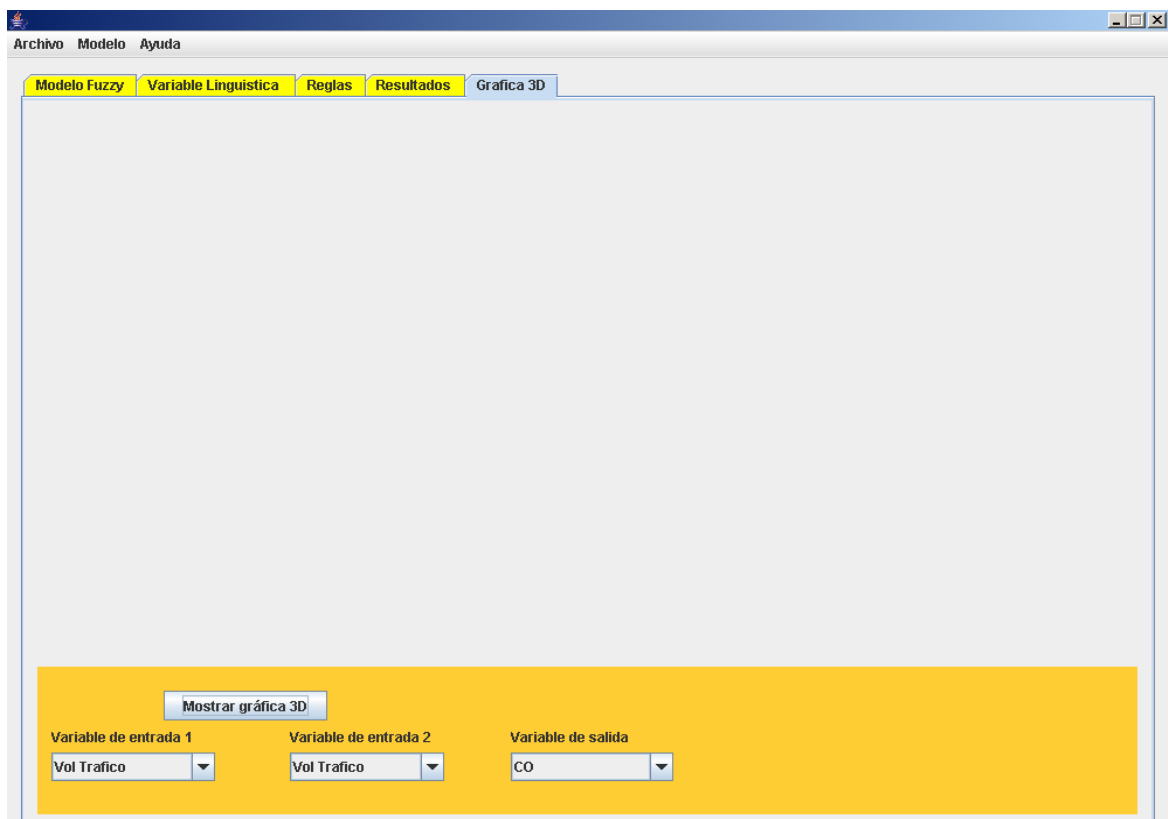
        layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
        .add(org.jdesktop.layout.GroupLayout.TRAILING, layout.createSequentialGroup()
            .addContainerGap(340, Short.MAX_VALUE)
            .add(pnVariables, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
        );
    }

```

4.4 Descripción de la ventana la Gráfica 3D

La ventana diseñada para mostrar la gráfica en 3D es la siguiente:

Figura 32. Ventana principal del grafico en 3D



Los constructores usados fueron:

```
public PanelGrafica3D(ArrayList<VariableLinguistica> variablesEntrada,
    ArrayList<VariableLinguistica> variablesSalida,
    ArrayList<ReglaInferencia> reglasInferencia) {
    this.entradas=variablesEntrada;
    this.salidas=variablesSalida;
    this.reglas=reglasInferencia;
    initComponents();
    inicializarParametros();
}

public PanelGrafica3D(PanelResultados pnResultados){
    this.pnResultados=pnResultados;
    this.entradas=pnResultados.variablesEntrada;
    this.salidas=pnResultados.variablesSalida;
    this.reglas=pnResultados.reglasInferencia;
    initComponents();
    inicializarParametros();
}
```

Y el método utilizado fue el siguiente:

```
///Inicializa los parámetros de la clase
public void inicializarParametros(){
    cmbVariableEntrada1.removeAllItems();
    cmbVariableEntrada2.removeAllItems();
    cmbVariableSalida.removeAllItems();
    for(VariableLinguistica v: entradas){
        cmbVariableEntrada1.addItem(v);
        cmbVariableEntrada2.addItem(v);
    }
    for(VariableLinguistica v: salidas){
        cmbVariableSalida.addItem(v);
    }

    if(variableEntrada1!=null && variableEntrada2!=null && reglas.size()>0)
    MainFrameView(variableEntrada1,variableEntrada2,variableSalida),
}
```

```

fr=new MainFrame(new
MainFrameView(pnResultados,variableEntrada1,variableEntrada2,variableSalida),
400,400);
}
public void graficarResultados3D(BranchGroup objRoot){
TransformGroup d1;
float x=0;
float y=1;
float z=0;
float nPuntos=100;
ColoringAttributes colorAtrib=new ColoringAttributes();
Appearance appear=new Appearance();
appear.setColoringAttributes(colorAtrib);
Box caja=new Box(0.05f,0.05f,0.05f,appear);
Transform3D trasladar=new Transform3D();
for (float j=variableEntrada1.xInicialRango;j<variableEntrada1.xFinalRango;
j+=(variableEntrada1.xFinalRango-variableEntrada1.xInicialRango)*0.05){
for (float k=variableEntrada1.xInicialRango;k<variableEntrada2.xFinalRango;
k+=(variableEntrada1.xFinalRango-variableEntrada1.xInicialRango)*0.05){
variableEntrada1.setValorLogico(j);
variableEntrada2.setValorLogico(k);
x=j;
y=variableSalida.xCentroide;//k;
pnResultados.calcularXCentroDeArea();
z=k;//variableSalida.xCentroide;
colorAtrib=new ColoringAttributes();
appear=new Appearance();
colorAtrib.setColor((float)((x-variableEntrada1.xInicialRango)/(variableEntrada1.xFinalRango-
variableEntrada1.xInicialRango)),
(float)((y-variableSalida.xInicialRango)/(variableSalida.xFinalRango-variableSalida.xInicialRango)),
(float)((z-variableEntrada2.xInicialRango)/(variableEntrada2.xFinalRango-
variableEntrada2.xInicialRango)));

trasladar.set(new Vector3f(0f,0.1f,0f));
d1=new TransformGroup(trasladar);
appear.setColoringAttributes(colorAtrib);

```

```
        caja=new Box(x,y,z,appear);
        d1.addChild(caja);
        objRoot.addChild(d1);
    }
}
}
```

5. PRUEBAS DEL PROTOTIPO

Las pruebas se realizaron para verificar la calidad y el desempeño del prototipo de sistema experto. Para realizar las pruebas se utilizó el último prototipo realizado de SEXDP y para comparar los resultados obtenidos se utilizaron los mismos datos de entrada y salida con el toolbox/Fuzzy Logic de Matlab 7.0.

Las variables utilizadas para realizar la prueba del prototipo son las siguientes:

Variables de entrada:

- Volumen del Tráfico
- Intensidad Solar
- Velocidad del viento
- Temperatura

Variables de salida:

- Monóxido de Carbono (CO)
- Ozono (O₃)

5.1 DESCRIPCIÓN DE LAS VARIABLES DE ENTRADA Y SALIDA¹⁶

5.1.1 Volumen del Tráfico

Para acceder a datos para esta variable se consultó un proyecto de grado que se realizó en el año 2005, debido a que no existen estudios recientes que faciliten esta información, el proyecto se llama “FASE II: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN GEOGRÁFICA DE SOPORTE PARA EL SISTEMA INTEGRAL DE TRANSPORTE MASIVO EN EL ÁREA

¹⁶ Jaimes Becerra, Marcela, castillo, Edgar Fernando, Chacón Velasco, Jorge Luís: Artículo de Modelo Difuso de la Dispersión del Aire en Bucaramanga.

METROPOLITANA DE BUCARAMANGA¹⁷”. El conteo se realizó en periodos donde se esperaría encontrar las horas pico, es decir, un volumen vehicular critico o máximo para intervalos de tiempo, los cuales fueron: En la mañana de 6:00 a.m. a 8:30 a.m., al mediodía de 11:30 a.m. a 3 p.m. y en la tarde de 5:30 p.m. a 8 p.m.

El procedimiento para realizar estos conteos consiste en tomar intervalos de 15 minutos durante el tiempo establecido, por cada sentido y teniendo en cuenta los vehículos que giran y los que continúan. La hora que presenta el volumen máximo de vehículos es la que se establece como la hora pico.

Finalmente esta variable de entrada se subdividió en 3 conjuntos borrosos a saber: Bajo o Descongestionado, Medio o Semicongestionado, y Alto o Congestionado.

5.1.2 Intensidad Solar

Esta variable de entrada fue subdividida en 3 conjuntos borrosos a saber: Noche o Bajo, Aurora o Medio y Mediodía o Alto. Para esto el modelo se baso de nuevo en la CDMB¹⁸. Las unidades que se tomaron para medir esta variable es Watios/m². El subconjunto noche lo tomamos con una función de pertenencia del tipo campana de gauss pero muy pequeña, ya que la radiación solar siempre permanece nula durante las horas nocturnas, y para los otros subconjuntos también se hace uso de una función de pertenencia del tipo campana de gauss, mientras en Aurora existe menor grado de pertenencia en raciones altas cuyo comportamiento sucede en horas tempranas o en el atardecer, es decir entre 6:00 a.m. y 8:00 a.m. como también entre 4:00 p.m. y 7:00 p.m; para el mediodía una

¹⁷ HUERTAS BECERRA, RAMIRO ANDRÉS, JEREZ QUINTERO, IVÁN FERNANDO

¹⁸ CDMB: <http://www.cdmb.gov.co/conozca/páginas/ibuca.htm>

función del mismo tipo con un máximo grado de pertenencia a este conjunto una radiación correspondiente a un intervalo entre 800 watos/m² y 1000 watos/ m².

5.1.3 Velocidad del Viento

En este proyecto¹⁹ se muestra una tabla de datos tomados por la estación de monitoreo (centro) ubicada en la terraza de Mercadefam de la carrera 15.

Tabla 8. Datos de la estación centro

HORA	VELOCIDAD DEL VIENTO m/s
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

Datos tomados por la estación centro de la CDMB en el año 2003

¹⁹ Amado, Juan Pablo. Castro, Juan Carlos y otros (2004). Estudio Comparativo de Contaminación atmosférica por la operación de un S.I.T.M en Bucaramanga. Universidad Industrial de Santander. Bucaramanga

Aunque estos datos no son los más recientes se usaron en el desarrollo del modelo porque ya tienen unas modificaciones que son necesarias, debido a que en la zona existen elementos altos (los edificios) que provocan que la velocidad del viento disminuya debido a la presión. Finalmente esta tabla muestra el cálculo de la velocidad a 2 metros de altura. La cual es la recomendable para estudios en calidad del aire, ya que a esta altura es en la cual nosotros respiramos. Finalmente a esta variable de entrada se subdividió en 3 conjuntos borrosos, a saber: Lento o Bajo, Moderado o Medio y Rápido o Alto.

5.1.4 Temperatura

Para esta variable de entrada los datos fueron tomados de la CDMB²⁰ y se subdividió en 3 conjuntos borrosos, a saber: Frío o Bajo, Templado o Medio, y Calor o Alto. Para describir los grados de pertenencia a frío utilizamos la función gaussiana, esta misma función se utiliza para los otros dos conjuntos borrosos, teniendo en cuenta los datos a utilizar suministrados por la CDMB.

5.2 DESCRIPCIÓN DE LOS GASES CONTAMINANTES

Los principales gases contaminantes que se monitorean actualmente son:

- Monóxido de Carbono (CO)
- Óxidos de Nitrógeno (NOx)
- Ozono (O3)
- Material Particulado (PTS, PM10 y PM2,5)
- Dióxido de Azufre (SO2)

²⁰ CDMB: <http://www.cdmb.gov.co/conozca/páginas/ibuca.htm>

A continuación se presenta una tabla con algunos contaminantes y sus respectivas unidades.

Tabla 9. Unidades de los contaminantes

µg/m³: microgramos por metro cúbico; ppb: partes por billón; ppm: partes por millón.

CONTAMINANTE	PERIODO	NORMA	UNIDAD
Partículas Suspendidas PM10	24 horas	134	ug/m ³
Óxidos de Azufre SO2	24 horas	136	ppb
Óxidos de Nitrógeno NO2	1 hora	114	ppb
Monóxido de Carbono CO	8 horas	31	Ppm
Ozono O3	1 hora	77	Ppb

<http://www.cdmb.gov.co/conozca/páginas/ibuca.htm>

A continuación se da una breve explicación de cada uno de los contaminantes con el fin de conocer como se originan y que efectos producen en la salud, esto proporciona una idea de la importancia de controlar su presencia en el medio ambiente en que vivimos.

5.2.1 Monóxido de Carbono (CO)

El monóxido de carbono es un contaminante primario (emitido directamente por las fuentes) y es bastante estable en la atmósfera, por lo cual no reacciona con otros compuestos. No presenta color, sabor ni olor; es ligeramente menos denso que el aire, constituido por un átomo de carbono y uno de oxígeno en su estructura molecular. En condiciones normales no ocasiona ningún daño, sin embargo, en concentraciones altas, afecta seriamente el sistema respiratorio. Su peligro es mayor en aquellas personas que padecen enfermedades cardiovasculares, angina de pecho o enfermedad vascular periférica;

concentraciones elevadas de CO en un área cerrada, pueden causar la muerte por insuficiencia cardíaca o sofocación.

Cuando usamos combustibles (como la gasolina de tu automóvil), producimos CO. Puede ser que estés respirando altos niveles de CO en los alrededores de calles o intersecciones muy transitadas.

Las altas concentraciones de CO pueden existir en ciudades con mucho tráfico. Gente que pasa los días en las calles (conductores de autobuses, camiones y patrullas, inspectores de vehículos, encargados de estacionamientos, peatones y ciclistas, trabajadores que reparan las calles y vendedores ambulantes) pueden respirar más CO. Los conductores de automóviles también están expuestos al CO proveniente del tráfico y posiblemente, de las emisiones de su vehículo o del humo de cigarro. Cuando el automóvil está detenido, los niveles de CO dentro del vehículo pueden incrementar.

5.2.2 Óxidos de Nitrógeno (NOX)

Óxidos de nitrógeno es el nombre que se aplica a varios compuestos químicos gaseosos formados por la combinación de oxígeno y nitrógeno. En función de la valencia atómica que utilice el nitrógeno reciben distintos nombres y tienen distintas formulaciones, siendo los más frecuentes el óxido nitroso y el óxido nítrico.

El monóxido de nitrógeno y el dióxido de nitrógeno constituyen dos de los óxidos de nitrógeno más importantes toxicológicamente; ninguno de los dos es inflamable.

El monóxido de nitrógeno es un gas a temperatura ambiente de olor dulce penetrante, fácilmente oxidable a dióxido de nitrógeno. Mientras que el dióxido de nitrógeno tiene un fuerte olor desagradable. El dióxido de nitrógeno es un líquido a

temperatura ambiente, pero se transforma en un gas pardo-rojizo a temperaturas sobre los 21 °C.

Los óxidos de nitrógeno son liberados al aire desde el escape de vehículos motorizados (sobre todo diésel y de mezcla pobre), de la combustión del carbón, petróleo o gas natural, y durante procesos tales como la soldadura por arco, galvanoplastia, grabado de metales y detonación de dinamita. También son producidos comercialmente al hacer reaccionar el ácido nítrico con metales o con celulosa.

5.2.3 Ozono (O₃)

El ozono presente en la capa baja de la atmósfera se considera como uno de los contaminantes de mayor preocupación en la actualidad, ya que es altamente oxidante y afecta a los tejidos vivos; se asocia con diversos padecimientos en la salud humana, es un agente irritante para el sistema respiratorio, produce tos, dolor al respirar e inflamación del tejido pulmonar. El ozono se clasifica como agente contaminante secundario, pues no es emitido directamente por la fuente a la atmósfera, sino que se genera por la transformación fotoquímica de los hidrocarburos con los óxidos de nitrógeno, en presencia de la luz solar.

5.2.4 Material Particulado (PST, PM₁₀ y PM_{2,5})

En contaminación atmosférica se reconoce como partícula a cualquier material sólido o líquido con un diámetro que oscila entre 0.0002 y 500 micrómetros (µm), en conjunto se designan como partículas totales en suspensión o PST. Las fuentes de emisión de partículas pueden ser naturales o antropogénicas.

Entre las fuentes naturales se encuentran: erosión del suelo, material biológico fraccionado, erupciones volcánicas, incendios forestales, etc. Entre las fuentes

antropogénicas se encuentran: combustión de productos derivados del petróleo, quemas en campos agrícolas y diversos procesos industriales. El material particulado lo podemos clasificar por su tamaño en:

- **Partículas sedimentables ($>10\ \mu\text{m}$):** son partículas que por su peso tienden a precipitarse con facilidad, razón por la cual permanecen suspendidas en el aire en períodos cortos de tiempo. Por lo general no representan riesgos significativos para la salud, sin embargo la exposición continua a altas concentraciones puede causar irritación de garganta y mucosas.

- **Partículas menores a 10 micrómetros o PM10 ($\leq 10\ \mu\text{m}$):** son partículas de diámetro equivalente o menor a $10\ \mu\text{m}$. Se consideran perjudiciales para la salud debido a que no son retenidas por el sistema de limpieza natural del tracto respiratorio, causando problemas en dicho sistema que pueden ir desde la disminución de la función pulmonar hasta producir enfermedades crónicas respiratorias que llevan a la muerte.

- **Partículas menores a 2,5 micrómetros o PM2,5 ($\leq 2,5\ \mu\text{m}$):** es la fracción fina del material particulado y está compuesta por partículas de diámetro equivalente o menor a $2,5\ \mu\text{m}$. Es considerada la fracción más peligrosa para la salud humana debido a que existe un mayor grado de penetración y permanencia de estas partículas en el aparato respiratorio, sobretodo en los alvéolos, lo que combinado con la composición química de las mismas, las hace más peligrosas.

Esta fracción está compuesta mayoritariamente por derivados de carbono y aerosoles secundarios, que no son emitidos directamente a la atmósfera, sino que se forman fundamentalmente por reacciones químicas de sustancias primarias, como óxidos de nitrógeno (NO_x), dióxido de azufre (SO_2), amoníaco (NH_3) y compuestos orgánicos volátiles (COV).

Puede considerarse un factor de muerte prematura en la población; de los elementos presentes en la fracción fina, los más nocivos para la salud son los

derivados de carbono, conocidos como “carbono orgánico”, por tener acción cancerígena de algunos de sus compuestos y propiedades tóxicas en otros. El “carbono elemental”, es también relevante ya que es el portador en el que se absorben gases como SOx y NOx que, junto con la humedad atmosférica, generan la acidez del material particulado.

5.2.5 Dióxido de Azufre (SO₂)

Los óxidos de azufre son gases incoloros que se forman al quemar azufre. El dióxido de azufre (SO₂) es el contaminante criterio que indica la concentración de óxidos de azufre en el aire. La fuente primaria de óxidos de azufre es la quema de combustibles fósiles, en particular el carbón. Se ha denominado al dióxido de azufre como un contaminante que "pasa a través de" porque la cantidad de dióxido de azufre emitido al aire es casi la misma cantidad presente en el combustible.

Se ha encontrado que los óxidos de azufre perjudican el sistema respiratorio, especialmente de las personas que sufren de asma y bronquitis crónica. Los efectos de los óxidos de azufre empeoran cuando el dióxido de azufre se combina con partículas o la humedad del aire. El efecto que causa mayor preocupación es la contribución de óxidos de azufre en la formación de lluvia ácida.

5.3 INFORME AMBIENTAL DE BUCARAMANGA

En Bucaramanga la institución encargada de monitorear y analizar la contaminación en el aire es la Corporación Autónoma Regional para la Defensa de la Meseta de Bucaramanga (CDMB), bajo la implementación de la Red de Monitoreo de Calidad del Aire del Área Metropolitana de Bucaramanga que está en funcionamiento desde el año 2001.

• Red de monitoreo de la calidad del aire

La Corporación Autónoma Regional para la Defensa de la Meseta de Bucaramanga, ha dispuesto en su Red de Monitoreo de la Calidad del Aire cinco (5) estaciones que permiten medir la contaminación del aire en zonas de gran importancia desde el punto de vista de mayor impacto a la comunidad, estas zonas son: Chimitá, Centro, Ciudadela Real de Minas, Autopista a Floridablanca y el Norte. Dentro de estas estaciones, la que realiza el muestreo más significativo de la contaminación por fuentes móviles es la ubicada en el centro de Bucaramanga (carrera 15 con calle 34), por ser una de las zonas con mayor tráfico vehicular sobretodo de vehículos pesados, obteniendo datos importantes de contaminación proporcionados por vehículos de transporte público con motores de diesel cuyo combustible es ACPM y por vehículos que usan gasolina para su funcionamiento.

Los datos obtenidos por cada estación son procesados en la Central de Información de la CDMB, permitiendo conocer las concentraciones de los contaminantes. Para conocer el estado actual del contaminante, la CDMB calcula el Índice de la Calidad del Aire, permitiendo clasificar el estado del contaminante como Bueno, Moderado, Regular, Malo o Peligroso.

• Índice de Calidad del Aire (IBUCA)

Es un índice que reporta diariamente la calidad del aire que se respira y su correlación con la salud. Este índice es calculado para los cinco principales contaminantes (ozono (O₃), dióxido de nitrógeno (NO₂), dióxido de azufre (SO₂), monóxido de carbono (CO) y material particulado (PM₁₀)), para los cuales se ha establecido una norma de calidad de acuerdo a la legislación ambiental vigente en Colombia. A continuación se presenta cómo es calculado el IBUCA; la fórmula del indicador es:

$$IBUCA = \frac{C_i}{NCA_i} * 10$$

Donde:

IBUCA: Índice de Calidad del Aire del Área Metropolitana de Bucaramanga.

C1: Es la concentración medida del contaminante a evaluar.

NCA1: Es la norma para cada uno de los contaminantes.

El IBUCA es el valor máximo de la relación entre la concentración de cada Contaminante seleccionado (PM10, SOx, NOx, CO y O3) y la concentración máxima permitida para dicho contaminante multiplicado por 10. En la tabla 28 se ven los valores máximos permitidos para cada contaminante según el Decreto 02 de 1982 expedido por el Ministerio de Salud.

Existe un IBUCA para cada una de las estaciones de monitoreo instaladas en el Área Metropolitana de Bucaramanga. De cada estación se toma la información sobre concentración de los contaminantes y se determina un valor de IBUCA para cada contaminante (PM10, SOX, NOX, CO, O3). El IBUCA de la estación equivale al valor mayor de los obtenidos anteriormente. El más alto de los valores entre las estaciones se convierte en el Índice de Calidad del Aire del día. Por ejemplo, como se ve en la tabla, si la concentración de los diferentes contaminantes en una estación determinada corresponde a:

Tabla 10. Ejemplo de valores medidos en una estación

PM10	SOX	NO2	CO	O3
50 µg/m³	30 ppb	80 ppb	10 ppm	38 ppb

CDMB, <http://www.cdm.gov.co>

Al calcular el IBUCA para cada contaminante (ver tabla abajo), como resultado de dividir la concentración de cada contaminante por su respectiva norma y multiplicar por 10, se tiene:

Tabla 11. Valores de IBUCA correspondientes a cada contaminante según el ejemplo

IBUCA PM10	IBUCA SOX	IBUCA NO2	IBUCA CO	IBUCA O3
3,73	2,2	7,01	3,2	4,93

CDMB, <http://www.cdmb.gov.co>

El IBUCA de la estación corresponde a 7,01 y el contaminante responsable es NO2.

El índice de calidad del aire se puede ver como una regla que va de 0 a 10. El valor más alto del IBUCA corresponde al nivel más alto de contaminación de aire y al efecto más perjudicial en la salud. Por ejemplo, un valor IBUCA de 2,60 representa un estado regular en la calidad del aire y poca probabilidad en el efecto de la salud, mientras que un valor IBUCA por encima de 10 representa una calidad del aire peligroso. Sin embargo, ciertos grupos de personas son particularmente sensibles a efectos dañinos de algunos contaminantes, es decir, que ellos son afectados a niveles más bajos que la población en general.


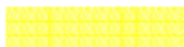



Un valor IBUCA de 2,50 corresponde a la cuarta parte de la norma de calidad local para los contaminantes, lo cual es el nivel que ha establecido la CDMB para proteger la salud de la población; de esta forma, valores por debajo de 2,50 son asociados como satisfactorios. La unidad de medida del índice es adimensional y posee una escala de 0 a 10 que depende del grado de contaminación del aire.

Este indicador está relacionado con la afectación que tiene la contaminación del aire sobre la salud humana.

A continuación en la tabla 12, se presenta la categorización de los valores de IBUCA, el cual tiene asignados unos colores específicos por cada una de las categorías para facilitar su interpretación por parte de la comunidad. Además los

colores permiten determinar rápidamente si los contaminantes en el aire se encuentran en niveles perjudiciales para la salud.

Tabla 12. Categorización de los valores de IBUCA

IBUCA	DESCRIPCIÓN	CARACTERIZACIÓN	COLOR
0 – 1,25	Bueno	La calidad de aire es considerada como satisfactoria y la afectación en la contaminación del aire es pequeña y no evidencia ningún efecto en la salud humana.	
1,26 – 2,5	Moderado	La calidad de aire es aceptable y no tiene ningún efecto sobre la población en general.	
2,51 – 7,5	Regular	Aumento de molestias en personas con padecimientos respiratorios y cardiovasculares; aparición de ligeras molestias en la población en general.	
7,51 – 10	Malo	Agravamiento significativo de la salud en personas con enfermedades cardíacas o respiratorias. Afectación de la población sana.	
Mayor que 10	Peligroso	Alto riesgo para la salud de la población. Aparición de efectos al nivel de daño.	

CDMB, <http://www.cdm.gov.co>

El Ministerio de Ambiente, Vivienda y Desarrollo Territorial expidió la Resolución 0601 del 4 de Abril de 2006, en la cual reglamenta los nuevos valores máximos permitidos para cada contaminante que se muestran en la tabla 13. Sin embargo,

esta reglamentación comienza a regir solo tres meses después de su publicación en el diario oficial.

Tabla 13. Valores máximos permitidos para cada contaminante según la nueva Resolución 0601

Contaminante	Unidad	Límite máximo permisible	Tiempo de Exposición
PST	$\mu\text{g}/\text{m}^3$	100	Anual
		300	24 horas
PM10	$\mu\text{g}/\text{m}^3$	70	Anual
		150	24 horas
SO2	Ppm ($\mu\text{g}/\text{m}^3$)	0.031 (80)	Anual
		0.096(250)	24 horas
		0.287(750)	3 horas
NO2	ppm ($\mu\text{g}/\text{m}^3$)	0.053(100)	Anual
		0.08(150)	24 horas
		0.106(200)	1 hora
O3	ppm ($\mu\text{g}/\text{m}^3$)	0.041(80)	8 horas
		0.061(120)	1 horas
CO	ppm (mg/m3)	8.8(10)	8 horas
		35(40)	1 horas

Ministerio de Ambiente, Vivienda y Desarrollo Territorial.

CONCLUSIONES

- La Lógica difusa demostró ser una herramienta muy fuerte y eficiente en la construcción de prototipos de sistemas expertos para diagnosticar la polución del aire, debido a que esta se presta para ser aplicada en diversos campos o áreas complejas donde no se tienen datos concretos.
- Los diagramas de clases desarrollados con UML fueron de gran utilidad para desarrollar una buena interfaz de usuario y hacerla lo más amigable y didáctica para los usuarios finales.
- Para el manejo del prototipo de sistema experto se requiere un poco de conocimiento en lógica difusa y sobre la polución del aire, debido a que si el usuario quiere resultados o salidas coherentes o parecidas a la realidad, las entradas han de ser lógicas y reales.
- La elección de java como lenguaje de desarrollo del prototipo nos da la posibilidad de seguir trabajando con software libre, debido a que es un lenguaje muy poderoso que permite la creación de aplicaciones y prototipos para cualquier sistema operativo.
- La experiencia que se consigue al interactuar con diferentes tipos de programadores es buena, ya que le ayuda a tener un sentido más amplio para futuros proyectos donde intervengan varios programadores.
- Los prototipos software son una herramienta funcional e importante a la hora de hacer validaciones de modelos o modelados realizados en UML.

- Interacción. El software de varios proveedores puede funcionar como conjunto. Un proveedor utiliza clases de otros. Existe una forma estándar de localizar clases e interactuar con ellas. El software desarrollado de manera independiente en lugares ajenos debe poder funcionar en forma conjunta y aparecer como una sola unidad ante el usuario.
- Independencia del diseño. Las clases están diseñadas para ser independientes del ambiente de plataformas, hardware y software. Utilizan solicitudes y respuestas con formato estándar. Esto les permite ser utilizadas en múltiples sistemas operativos, controladores de bases de datos, interfaces de usuario gráficas, etc. El creador del software no tiene que preocuparse por el ambiente o esperar a que éste se especifique.
- La calidad del aire que se respira en la ciudad se ve afectada por la contaminación producida por las fuentes móviles, no obstante, las autoridades ambientales solo se dedican a monitorear los niveles de los contaminantes en el aire sin preocuparse por establecer planes concretos que permitan bajarlos o por lo menos controlarlos de forma efectiva.

RECOMENDACIONES

- Seguir con el proceso de la lógica difusa no solo en este campo de la contaminación del aire, sino en otros campos donde nadie haya trabajado para ver los grandes resultados que se consiguen.
- Tener la asesoría de profesionales o investigadores en el área de calidad del aire para generar reportes específicos de acuerdo a las necesidades y acordes a la realidad.
- Hay que utilizar una interfaz de usuario gráfica de modo que el usuario apunte a iconos o elementos de un menú desplegado, relacionados con los objetos. En determinadas ocasiones, el usuario puede ver un objeto en la pantalla. Ver y apuntar es más fácil que recordar y escribir.
- Las autoridades ambientales además de disminuir los valores permisibles de emisión de contaminantes para las fuentes móviles, deben establecer programas de control de la contaminación donde se contemplen medidas que incentiven a los ciudadanos para que adquieran una conciencia ambiental que les permita la aceptación de dichas normas y los convierta en parte fundamental de tales programas.

BIBLIOGRAFÍA

BOOCH, G., RUMBAUGH, J., JACOBSON, I. (1999) El Lenguaje Unificado de Modelado. Addison-Wesley.

CDMB Corporación Autónoma Regional para la Defensa de la Meseta de Bucaramanga. SIA Sistema de Información Ambiental. [web]:

<http://www.cdm.gov.co/web/lacdm/>

CEBALLOS, Francisco Javier. JAVA 2 CURSO DE PROGRAMACIÓN, 3ª edición. RA-MA Editorial.

ECO, Umberto. *Cómo se hace una tesis: Técnicas y procedimientos de investigación, estudio y escritura*. Vigésimo cuarta edición, Barcelona, Editorial Gedisa S.A., 2001.

Cox, Earl. T^oHE FUZZY SYSTEM HANDBOOK, primera edición. Editorial AP Profesional.

FOWLER, Martin y SCOTT, Kendall. *UML gota a gota*. Primera edición, México, Editorial Pearson, 1999.

GÓMEZ, Flórez Luís Carlos. *Planeación de proyectos: Un enfoque para Ingeniería de Sistemas e Informática*. Primera edición, Bucaramanga, Ediciones UIS, 2001.

LARMAN, C. UML y Patrones: Una Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado

MARTIN, Robert. UML para Programadores java. Primera edición, España, Editorial Pearson, Prentice Hall, 2004.

PRESMAN, Roger. *Ingeniería del Software: Un enfoque práctico*. Cuarta edición, México, Editorial McGraw Hill, 1998.

ROMERO MERCHÁN, Alberto y VARGAS JEREZ, Pascual Alberto. ORGANIZACIÓN, DISTRIBUCIÓN EN PLANTA, PLANIFICACIÓN DE MONTAJE Y ESTUDIO ECONÓMICO DEL CENTRO DE DIAGNÓSTICO AUTOMOTOR PARA LA DIRECCIÓN DE TRÁNSITO DE BUCARAMANGA. Tesis de grado. 2006

Otras referencias:

<http://www.willydev.net/descargas/Articulos/General/umlTotal.pdf>

Esta página de Internet, incluye una extensa documentación acerca de UML.

<http://www.drcaffeine.com/additionaltopics>

www.monografias.com

Marco teórico:

Canca, J. "Sistemas Expertos". Esi2.us.es. URL:

<http://www.esi2.us.es/~dco/sistemas.htm>

Castro, G. "Tendencias de los Sistemas Expertos". Netmedia.info URL:

http://www.netmedia.info/netmedia/articulos.php?id_sec=32&id_art=2255

Chapa, S. "Arquitectura de Sistemas Expertos". URL:

http://delta.cs.cinvestav.mx/~schapa/red/intro_lm/node46.html

Criado, J. "Sistemas Expertos" Worldonline.es URL:

<http://home.worldonline.es/jmariocr/>

Ibarreta, J. "Sistemas Expertos: Áreas de aplicación". Geocities. URL:

<http://www.geocities.com/javierml.geo/doc/SistemasExpertos.html>

Samper, J. "Sistemas Expertos: El conocimiento al poder". Psicología.com URL:

http://www.psicologia.com/articulos/ar-jsamper01_2.htm

Velarde, J. "Sistemas Expertos". Ucm.es URL:

http://www.ucm.es/info/eurotheo/diccionario/S/sistemas_expertos.htm

http://www.tdx.cesca.es/TESIS_UPC/AVAILABLE/TDX-0207105-105056//04Rpp04de11.pdf

<http://www.lcc.uma.es/~ppgg/FSS/FSS5.pdf>.

ANEXOS

1. REVISION BIBLIOGRAFICA SOBRE TESIS DE SISTEMAS EXPERTOS BASADOS EN LÓGICA DIFUSA PARA LA DETERMINACION DEL AIRE Y AREAS AFINES.

Para la revisión bibliográfica sobre sistemas expertos basados en lógica borrosa solo se consultó la biblioteca de la Universidad Industrial de Santander debido a la facilidad de acceder a las tesis y así poder analizar sobre el contenido de ellas. En la biblioteca de la UIS no se encontró ninguna tesis que cumpla con el objetivo, pero si se encontró tesis sobre Sistemas Expertos, Lógica Difusa o Borrosa y de contaminación por monóxido de carbono pero todo por separado. A continuación se hace un breve resumen sobre las tesis encontradas:

✓ DESARROLLO DE UN PROTOTIPO DE SISTEMA EXPERTO PARA LA SELECCIÓN DE MATERIALES DE CONSTRUCCION EN PLANTAS DE PROCESOS QUIMICOS.

TESIS DE GRADO - GAMBOA QUESADA, RAMIRO; Universidad Industrial de Santander. Escuela de Ingeniería Química, 1990; Villamizar Montero, Christian Armando, dir

Este prototipo de Sistema Experto fue realizado para la selección de materiales de construcción, basados en los métodos existenciales (el del material que soporta las condiciones mas críticas de trabajo, el de emplear materiales que se han utilizado en plantas químicas conocidas, y el método basado en la carta de Bishop - Stern). Por esta razón se realizó una herramienta de programación un sistema que recopile estos conocimientos junto con los “Heurísticos” empleados en cada una de las etapas del proceso de selección.

A través de este proyecto se aclaran los conceptos relacionados con los materiales de construcción y su selección, y se esquematizan los llamados “sistemas expertos”, sus características y el lenguaje de programación empleado para su desarrollo, el prolog, dándose un fundamento teórico sobre lógica.

✓ **MODELO DE ARQUITECTURA DE UN SISTEMA EXPERTO PARA LA EVALUACION DE UN PROSPECTO PETROLIFERO.**

TESIS DE GRADO - CRUZ GUEVARA, LUIS ENRIQUE; Universidad Industrial de Santander 1994. Postgrado en Informática. Corredor Montagut, Martha Vitalia, dir.

Aquí se propone una arquitectura de Sistema Experto para la evaluación geoquímica de un prospecto petrolífero, esta proposición se basa en los conceptos tradicionales de roca de origen, roca acumuladora, roca sello y trampa; y en los conceptos aportados por la Inteligencia Artificial sobre: manejo de incertidumbre, diseño de Sistemas Expertos y Arquitecturas. Es importante establecer que en la prospección de hidrocarburos, esta evaluación se considera como un peldaño en el desarrollo de un Sistema Experto Integrado para la priorización de áreas de explotación.

Este proyecto fue implementado en la estación de trabajo SUNSpart del postgrado de informática de la UIS, utilizando NEXPERT OBJECT, un shell híbrido para construir Sistemas Expertos, que maneja esquemas de representación del conocimiento basados en reglas y frames, además de ser Orientado a Objetos y a Procedimientos.

✓ **SISTEMA EXPERTO QUE APOYA LA SELECCIÓN DE HERAMIENTAS DE CORTE EN PROCESOS DE MECANIZADO SHEC.**

TESIS DE GRADO - Corredor Montagut, Martha Vitalia, dir.; Gonzalez Jaimes, Isnardo, dir.; OROZCO CELIS, ROSS NELCY; Reyes Rivera, Norberto; Universidad Industrial de Santander. 2001. Escuela de Ingeniería Mecánica. Tesis

(pregrado); Universidad Industrial de Santander. Escuela de Ingeniería de Sistemas. Tesis (pregrado)

Es un software para la selección de herramientas de corte en procesos de mecanizado. Esta conformado por un módulo de consulta que permite obtener las herramientas y las condiciones de corte adecuadas para entrada de datos de entrada de maquina herramienta, pieza y condiciones de mecanizado específico y un segundo módulo de actualización que permite conservar y validar conocimientos que se relacionan con los procesos de torneado, fresado y taladro. Se elaboro aplicando las técnicas de Inteligencia Artificial relacionadas con los Sistemas Expertos y sistemas basados en el conocimiento, permitiendo simular los procesos de aprendizaje, memorización, razonamiento, comunicación y acción de un experto humano en la rama del mecanizado.

✓ APLICACIÓN DE LA LÓGICA DIFUSA PARA LA CARACTERIZACION DE YACIMIENTOS PETROLEROS EN EL INSTITUTO COLOMBIANO DEL PETROLEO ECP – ICP.

TESIS DE GRADO - CARRILLO FLOREZ, LILIANA MARIA; Soto Becerra, Rodolfo, tutor; Universidad Industrial de Santander. 2001. Escuela de Ingeniería de Sistemas e Informática. Tesis (pregrado)

Este documento es un vistazo global de un estudio particular de caracterización de yacimientos petroleros basados en Lógica Difusa; y el análisis, diseño, implementación e implantación del software basado en Lógica Difusa llamado “Reservoir Fuzzy Logic” y su integración al software “Oil Field Intelligence” del Instituto Colombiano del Petróleo.

La implementación del software “Reservoir Fuzzy Logic” se ha realizado en Matlab 5.3, aprovechando su motor de Lógica Difusa para Windows 95 y se valida el funcionamiento del software, con los conocimientos de los expertos en petróleos, por medio del estudio de la información del pozo BAA1Z del campo

Cusiana y su comparación con los resultados obtenidos con el software “Mirador/Cusiana Field” y verificación estadística de esta comparación.

✓ CONTROLADOR CON LÓGICA FUZZY BASADO EN MICROCONTROLADOR Y LABVIEW

TESIS DE GRADO, Jaime Eliécer Villalobos Barrera ; Jorge Eliécer Rodríguez Gómez ; director Jaime Guillermo Barrero Pérez. Universidad Industrial de Santander. 2004. Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones. Tesis (pregrado)

Se presenta una herramienta básica en software y hardware para la implementación eficiente, rápida, y económica de un tipo de controlador con Lógica Fuzzy basado en un microcontrolador de gama media o baja. Se trata esencialmente de un algoritmo en lenguaje ensamblador para microcontroladores PIC y MOTOROLA, y que puede extenderse fácilmente para su uso en sistemas basados en Microprocesador o Procesador digital de Señales (DSP).

Además se construye una herramienta software en LabVIEW para MS-Windows para la caracterización del sistema Fuzzy de dos entradas una salida y generación del código en lenguaje ensamblador que realiza control con Lógica Fuzzy en los microcontroladores PIC o Motorola CPU08.

Adicionalmente se describe el diseño del hardware construido para la demostración del control Fuzzy (tanto desde el microcontrolador como desde el computador) así como las principales características de sus componentes.

Ejemplo de aplicación del formato de control Fuzzy y del software FUZZYE3T. Se controla la temperatura de un sistema realimentado simple: Un elemento resistivo de calefacción. Se plantean las características del sistema y se implementa un control Fuzzy; utilizando como dispositivo de procesamiento, primero el computador IBM PC-Compatible y luego un microcontrolador.

✓ SOFTWARE DE APOYO AL CONTROL INTELIGENTE BASADO EN LÓGICA BORROSA -COBOR 1.0-

TESIS DE GRADO - Landinez Cruz, Juan José; REYES FIGUEROA, JUAN CARLOS; Ruiz Díaz, Fernando, dir.; Universidad Industrial de Santander. 2004. Escuela de Ingeniería de Sistemas e Informática. Tesis (pregrado)

COBOR 1.0 es una herramienta para el apoyo y desarrollo de sistemas de control, utilizando la lógica borrosa como una tecnología adaptativa. Mediante el monitoreo de las variables de control, se determina la acción a tomar por el controlador sobre la inferencia de una base de reglas de conocimiento definida a partir de la experiencia dada por un experto en el control de procesos, tomando un soporte formal de los fundamentos teóricos de los conjuntos.

El diseño del software se realizó utilizando el lenguaje Unificado de Modelado (Unified Modeling Language, UML), donde UML es un lenguaje o forma estandarizada de desplegar y escribir la estructura de un software o sistema por medio de conceptos orientados a objetos. Puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

El sistema se construyó en el ambiente de desarrollo DELPHI versión 6.0, que pertenece a la casa Borland y se presenta como una alternativa para el desarrollo de aplicaciones; Además, se destaca de Borland la velocidad, robustez y eficiencia de los compiladores que desarrolla.

Esto con el fin de desarrollar una herramienta que este en capacidad de enfrentarse a problemas de gran envergadura como lo son los controladores borrosos, ya que pueden existir sistemas de muchas entradas y salidas, conllevando a una mayor robustez del programa para soportar este tipo de aplicaciones.

COBOR 1.0 es una herramienta para el diseño y manejo de controladores inteligentes basados en lógica borrosa, debido a la gran aceptación que estos controladores borrosos han tenido últimamente se trato de implementar un sistema

capaz de proveer apoyo al experto de los procesos, para el manejo controlado de estos.

Después de tener las variables de entrada y salida definidas se continúa el proceso normal para la construcción de un controlador borroso, como son la base de reglas, los métodos de desemborronado, y otras muchas características que la herramienta provee al usuario como son:

- 📖 Generación de los conjuntos borrosos.
- 📖 Elección de la función de membresía.
- 📖 Construcción de las reglas borrosas.
- 📖 Método de desemborronado.
- 📖 Matriz de control.
- 📖 Impresión de los datos y el reporte del caso en estudio.

✓ SOFTWARE PARA EL CONTROL BORROSO BASADO EN ALGORITMOS MODERNOS “COBOR 2.0”

TESIS DE GRADO - MENDEZ, ELKIN JAVIER; Ordóñez Camargo, Oscar Mauricio; Reyes Figueroa, Juan Carlos, dir.; Ruiz Diaz, Fernando, dir.; Universidad Industrial de Santander. 2005. Escuela de Ingeniería de Sistemas e Informática. Tesis (pregrado)

Este documento contiene el informe final del proyecto titulado “Software para el control borroso basado en Algoritmos Modernos COBOR 2.0”. Proyecto que ha surgido en el grupo GEMA de la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander, y se inscribe dentro del área de investigación, descubrimiento de conocimientos y tecnologías adaptativas que se encuentran en pleno desarrollo a nivel mundial.

Brinda al usuario un ambiente de trabajo adecuado para el diseño, simulación e

implementación de sistemas de control basado en la lógica borrosa, que permita trabajar cinco variables de control. Permite la definición de los conjuntos borrosos y sus particiones correspondientes a variables de estado y acción.

COBOR 2.0 (Software para el Control Borroso), se realizó bajo el paradigma de la Ingeniería del software llamado Construcción de prototipos, que nos ofrece el mejor enfoque.

El sistema se construyó en el ambiente de desarrollo DELPHI versión 7.0, que pertenece a la casa Borland y se presenta como una alternativa para el desarrollo de aplicaciones; Además, se destaca de Borland la velocidad, robustez y eficiencia de los compiladores que desarrolla.

✓ COMPONENTE DE SISTEMA DE INFERENCIA DIFUSA (FIS) PARA EVOLUCIÓN 3.5

TESIS DE GRADO - Andrade Sosa, Hugo Hernado, dir.; González Pérez, Cesar Eduardo; MACHADO MENDOZA, GESMAN DAVID; Universidad Industrial de Santander. 2006. Escuela de Ingeniería de Sistemas e Informática. Tesis (pregrado)

En este documento se presentan en detalle las actividades desarrolladas en el proceso de construcción e integración a EVOLUCION, de un módulo software, que permita el trabajo con Sistemas de Inferencia basados en Lógica Borrosa (FIS por sus siglas en Ingles), como si tratase de un elemento mas del diagrama Forrester. Dicho software esta dotado de dos componentes, un editor que permite al usuario la creación y puesta a punto de Sistemas de Inferencia basados en Lógica Borrosa tipo Mandani, además de un componente de nombre FIS, el cual es el encargado de realizar los cálculos pertinentes a este elemento. Por su diseño, el módulo puede ser usado en el desarrollo de otras aplicaciones que requieren implementar Sistemas de Inferencia basados en Lógica Borrosa tipo Mandani.

Para llevar a cabo la formación en el desarrollo de software orientado a objetos, utilizando UML como medio para expresar los planos del software y lenguaje programación DELPHI 7 como herramienta de desarrollo del mismo.

✓ MODELO DE UNA BASE DE DATOS PARA EL SISTEMA EXPERTO BASADO EN LÓGICA DIFUSA QUE PERMITE PRONOSTICAR EL ÍNDICE DE CONTAMINACIÓN DEL AIRE EN EL ÁREA METROPOLITANA DE BUCARAMANGA

TESIS DE GRADO - Cachón Velasco, Jorge Luís, dir.; JACANAMIJOY TISOY, YENNY CRUZ; Jaimes Becerra, Marcela, dir.; Universidad Industrial de Santander. 2006. Escuela de Ingeniería de Sistemas e Informática. Tesis (pregrado)

El modelo de base datos se implementa en el Sistema de Información para Medir la Calidad del Aire – SIMCA –, aplicativo que funciona en una arquitectura basada en la web.

El presente proyecto está asociado al desarrollo del Sistema Experto Difuso para la Polución del Aire – SEXPD –, que el grupo GIEMA²¹ de Ingeniería Mecánica, el CEIAM²² de Ingeniería Química vienen desarrollando, y se encuentra financiado por el DIF. El macroproyecto pretende crear una primera versión del SEXDP, que contribuya a medir la contaminación del aire originada por las fuentes móviles.

Para el desarrollo de la herramienta se implementó tecnología de punta, como los struts para el uso del patrón de arquitectura MVC en Java. Los datos empleados para la evaluación de la herramienta y su respectivo indicador de calidad del aire fueron facilitados por la CDMB. La implantación del aplicativo web en las diferentes entidades ambientales locales de Bucaramanga, facilitará la investigación en calidad del aire, debido al fácil acceso y extracción de la información.

²¹ GIEMA: Grupo de Investigación en Energía y Medio Ambiente

²² CEIAM: Centro de Estudios e Investigaciones Ambientales

✓ ORGANIZACIÓN, DISTRIBUCIÓN EN PLANTA, PLANIFICACIÓN DE MONTAJE Y ESTUDIO ECONÓMICO DEL CENTRO DE DIAGNÓSTICO AUTOMOTOR PARA LA DIRECCIÓN DE TRÁNSITO DE BUCARAMANGA. TESIS DE GRADO. ROMERO MERCHÁN, ALBERTO y VARGAS JEREZ, PASCUAL ALBERTO. UNIVERSIDAD INDUSTRIAL DE SANTANDER. ESCUELA DE INGENIERÍA MECÁNICA. 2006. Ph.D. JORGE LUIS CHACÓN VELASCO

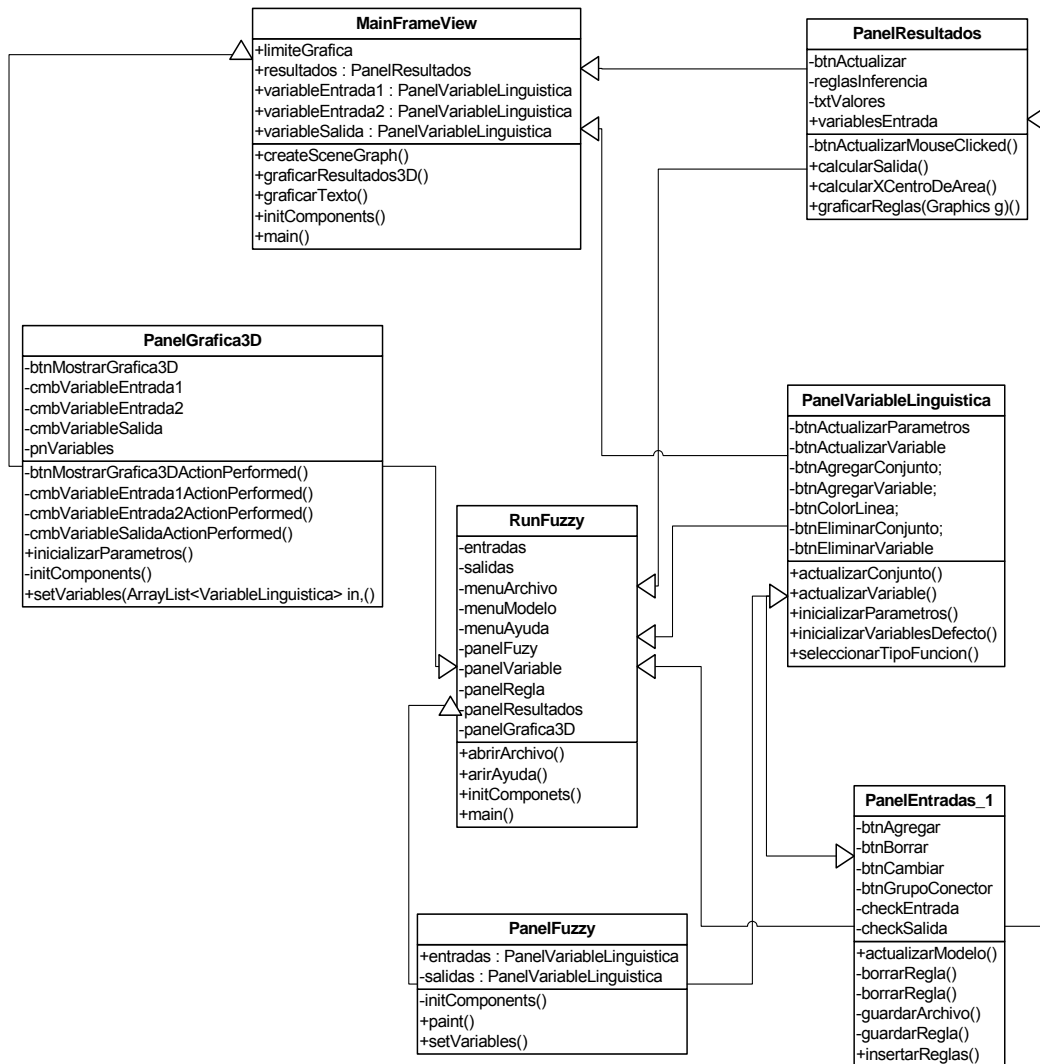
Este trabajo de grado muestra la necesidad de la creación de un Centro de Diagnóstico Automotor (CDA) para la Dirección de Tránsito de Bucaramanga (DTB), donde se realizarán las pruebas técnico-mecánicas y ambientales a los vehículos automotores; esto con el fin de reducir los altos niveles de contaminación del aire y contribuir a disminuir la accidentalidad en las carreteras. Inicialmente se encuentra un compendio de normas relacionadas con la generación de ruido, emisión de gases y partículas contaminantes provenientes de las fuentes móviles, las cuales fueron tomadas de la Agencia de Protección Ambiental (EPA) de los Estados Unidos y la Unión Europea, comparándolas con las actualmente vigentes en Colombia. Además, se presentan normas internacionales y locales relacionadas con el aspecto técnico-mecánico de los vehículos, realizando un paralelo que muestra las características de cada una de ellas. Posteriormente se encuentra el análisis de diversos estudios ambientales que dan una idea de la importancia de monitorear y controlar la contaminación producida por las fuentes móviles. Se continúa con la descripción de los equipos e instrumentos necesarios para la revisión de los vehículos y se indica su ubicación en forma de líneas de inspección; con base en esto se procede a hacer la distribución en planta del CDA, teniendo en cuenta además, la infraestructura disponible en la DTB.

✓ SIMULACION DE CONTAMINACION POR MONOXIDO DE CARBONO DE FUENTES MOVILES EN LA CARRERA 15 CON CALLE 36 DE BUCARAMANGA.

TESIS DE GRADO - ACEVEDO DUARTE, LEONARDO, dir.; MALDONADO MOJICA, RICARDO ARTURO; MARTINEZ, ALVARO, dir.; ORDONEZ SOLANO, IVAN, Universidad Industrial de Santander. 1995. Escuela de Química.

Para el desarrollo de la simulación en la zona seleccionada, se utilizaron programas de modelamiento de contaminación atmosférica, aprobado por la agencia de protección ambiental de los EEUU (EPA), se corrieron las versiones comerciales de trinity Consultants, Inc. de los modelos MOBILE5 y CALINE4, los cuales fueron adquiridos por ECOPETROL y amablemente facilitados por el Instituto Colombiano del Petróleo. Estos modelos se usaron separadamente, pero los resultados obtenidos con uno de ellos constituyen los datos de entrada al otro modelo. Con el modelo de MOBILE5 se estimaron los factores de emisión de los diferentes tipos de vehículos que circulan en Bucaramanga y especialmente en la intercepción de la carrera 15 con calle 36. una vez calculado los factores de emisión, se usa el modelo CALINE4 para realizar la simulación global de la zona escogida y determinar el nivel de contaminación producida por monóxido de carbono.

DIAGRAMA DE CLASES DE LA INTERFAZ GRAFICA DEL USUARIO



MANUAL DE USUARIO PARA LA INSTALACIÓN Y EL MANEJO DEL SOFTWARE

1. GENERALIDADES DEL SOFTWARE

SEXDP es el primer prototipo software realizado para diagnosticar la polución del aire originada por las fuentes móviles en el área metropolitana de Bucaramanga utilizando la Lógica Borrosa como una tecnología adaptativa. Los datos que se utilizaran serán tomados de la página de la CDMB (Corporación Autónoma Regional para la Defensa de la meseta de Bucaramanga) o datos que contenga el experto sobre las variables de estudio. Estas variables de estudio serán escogidas por el experto dependiendo de las salidas que quiera.

El prototipo se desarrollo en Java, debido a que este es un lenguaje muy poderoso y se puede adaptar a cualquier sistema operativo. También se utilizó para su desarrollo una herramienta llamada NetBeans 5.5, ya que esta presenta una excelente presentación y facilidad para trabajar con los objetos deseados. NetBeans, distribuido gratuitamente por la propia Sun (Creadora de Java), es un entorno completísimo al que no le falta absolutamente nada, pero tiene un problema enorme: Está hecho en Java. Esto está bien para demostrar la potencia de java a la hora de realizar aplicaciones de una calidad tan excepcional como NetBeans, pero precisamente, al estar hecho en Java sufre del mayor de los inconvenientes: Ocupa mucho espacio y es extremada e intolerablemente lento. Esta herramienta se encuentra disponible en la página Web <http://www.netbeans.org>, y también es necesario una librería llamada Java 3D, la cual es indispensable para realizar la gráfica en tres dimensiones. Otra herramienta que se utilizó fue JCreator pro, esta tiene ciertas ventajas como Implementación de colores en código, numeración de líneas, sencillez, robusto,

rápido, come pocos recursos, fácil manejo. Jcreator seguramente sea una utilidad que poca gente conozca ya que es una herramienta para la programación en Java. Simplemente por la sencillez de la misma y que esta echa explícitamente para eso, para la programación en Java. JCreator es un entorno de desarrollo integrado (IDE) muy peculiar, ya que se basa en una filosofía radicalmente distinta al resto de IDEs para Java, y es la sencillez.

2. REQUERIMIENTOS DEL HARDWARE Y SOFTWARE.

Procesador: Pentium III 1.2 GHz

Memória: 512 MB de Ram

Sistema Operativo: Windows xp

3. INSTALACION DEL SOFTWARE

Para el correcto funcionamiento del prototipo en el computador se deberá instalar la maquina virtual de java. La **máquina virtual de Java** (en inglés *Java Virtual Machine*, **JVM**) es un programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java.

El código binario de Java no es un lenguaje de alto nivel, sino un verdadero código máquina de bajo nivel, viable incluso como lenguaje de entrada para un microprocesador físico.

La gran ventaja de la máquina virtual java es aportar portabilidad al lenguaje de manera que desde SUN se han creado diferentes máquinas virtuales java para diferentes arquitecturas y así un programa .class escrito en un Windows puede ser

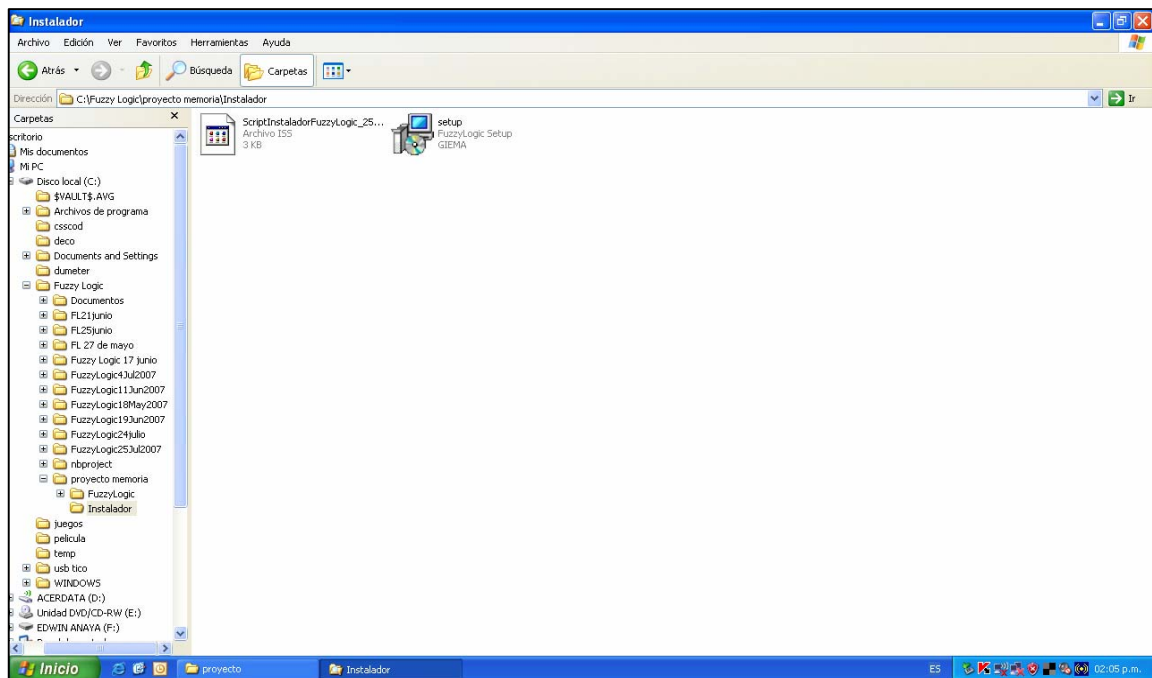
interpretado en un entorno Linux. Tan solo es necesario disponer de dicha máquina virtual para dichos entornos.

La máquina virtual de Java puede estar implementada en software, hardware, una herramienta de desarrollo o un Web browser; lee y ejecuta código precompilado bytecode que es independiente de la plataforma multiplataforma. La JVM provee definiciones para un conjunto de instrucciones, un conjunto de registros, un formato para archivos de clases, la pila, un heap con recolector de basura y un área de memoria. Cualquier implementación de la JVM que sea aprobada por SUN debe ser capaz de ejecutar cualquier clase que cumpla con la especificación. Este programa puede ser descargado desde la página de Sun Microsystems.

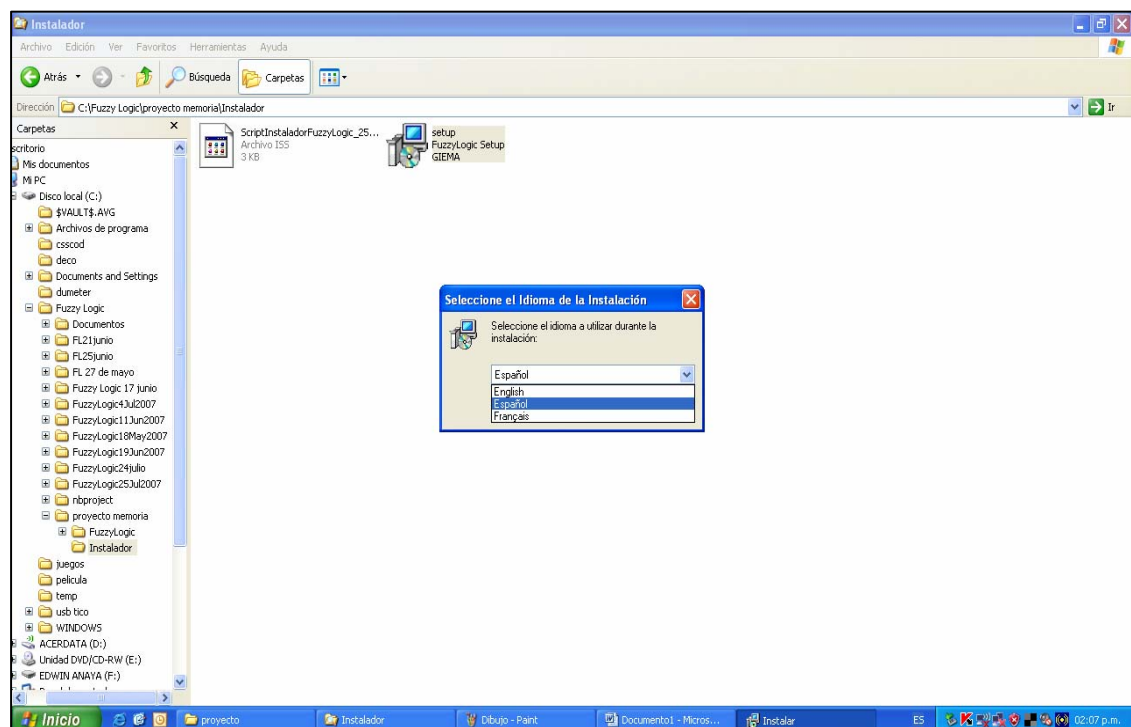
Otra aplicación a tener en cuenta a antes de la instalación del prototipo es haber instalado el paquete J3D, es una aplicación para poder ver las gráficas en tres dimensiones que mostrara el prototipo. Si no se instala este paquete de java no sera posible visualizar las graficas y por tanto el prototipo no se ejecutara.

El prototipo ha sido diseñado para instalarse en el computador, solo hay que seguir la secuencia de instrucciones que se muestran durante el proceso de instalación. A continuación se mostrara paso a paso el proceso de la instalación:

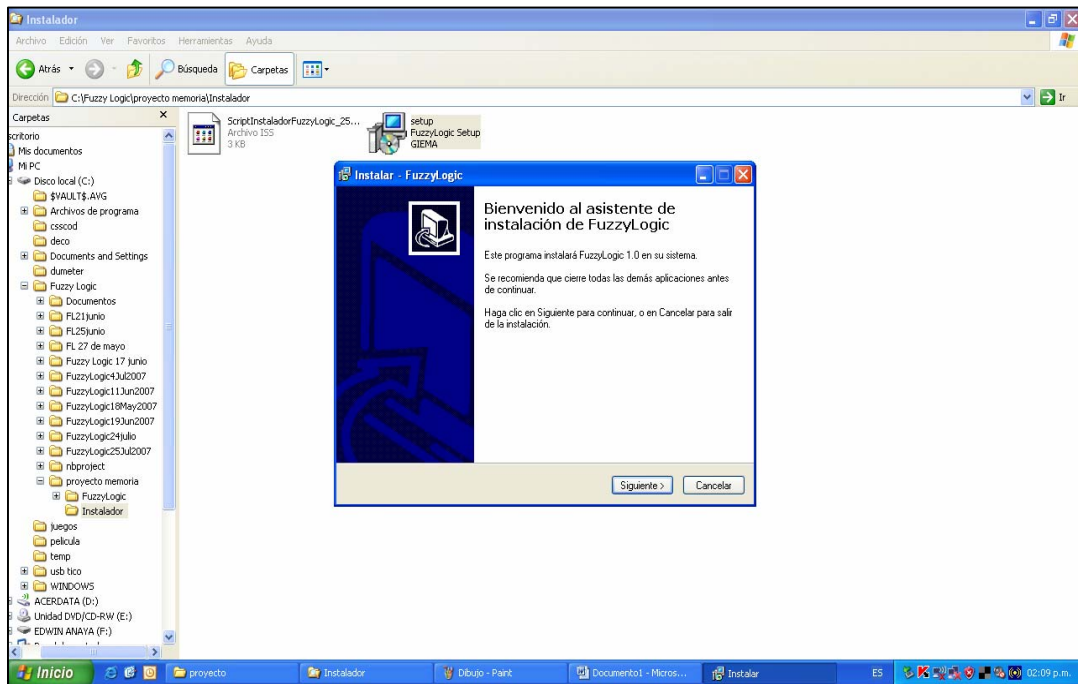
1. Se ubica el archivo que instalará el prototipo, el cual se llama setup, y se procede a seguir todas las instrucciones que allí le aparecen. A este archivo se le da doble click y se procede a instalar el prototipo.



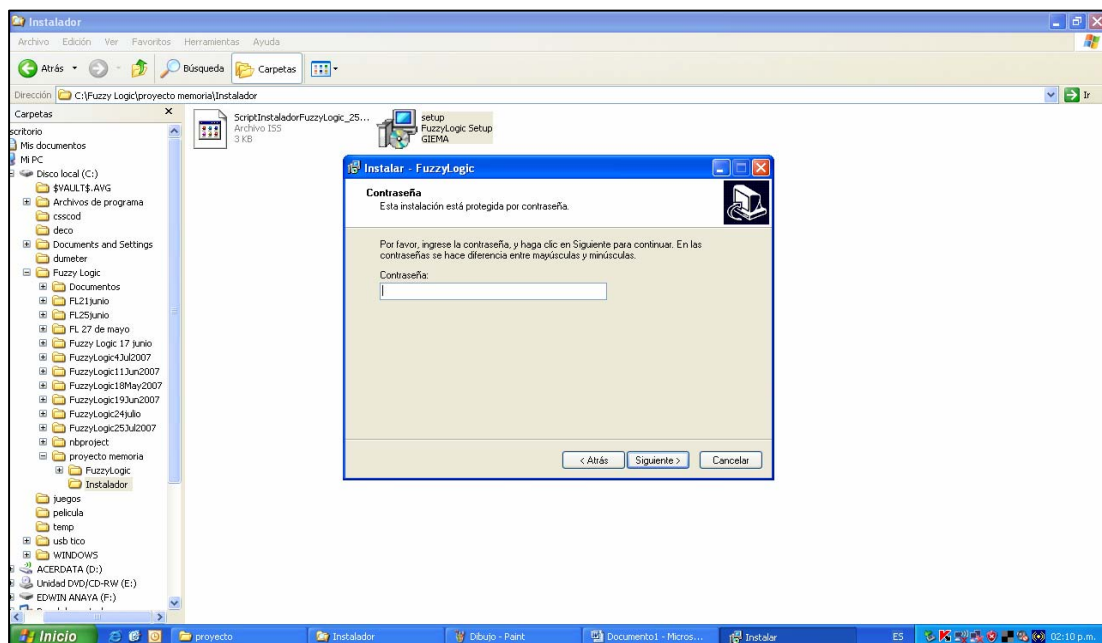
2. Después de dar doble click se procede a seleccionar e idioma de instalación, para este caso hay tres idiomas a seleccionar: español, ingles y francés.



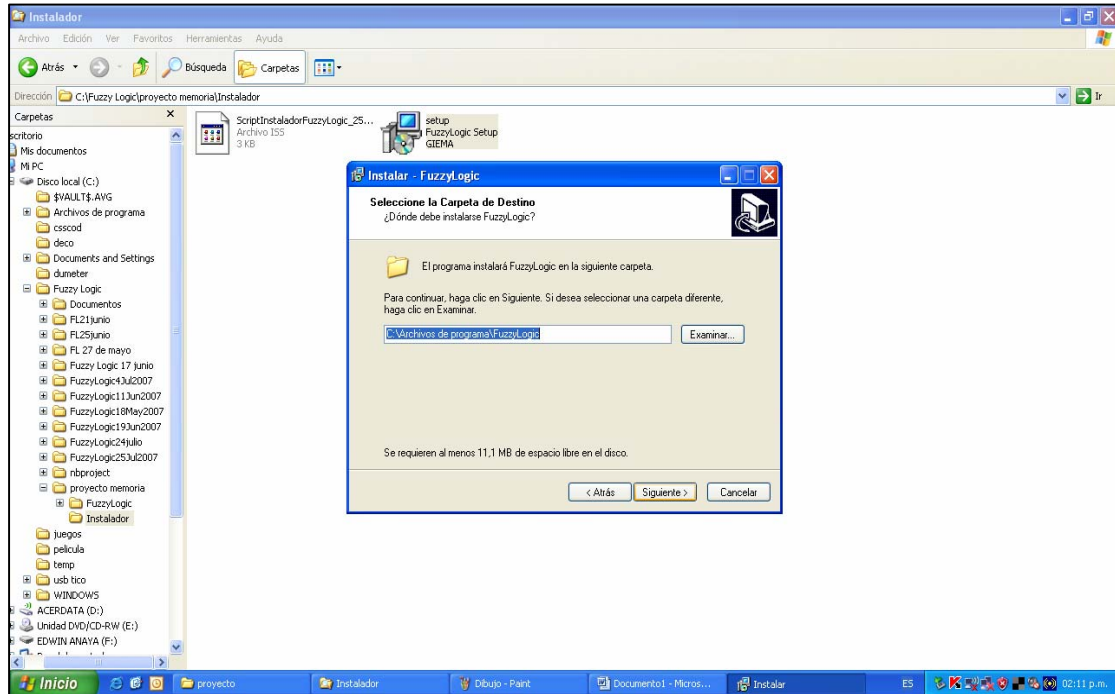
3. Después de seleccionar el idioma se da aceptar e inmediatamente aparece otra ventana que da la bienvenida al proceso de instalación y a continuación se da siguiente.



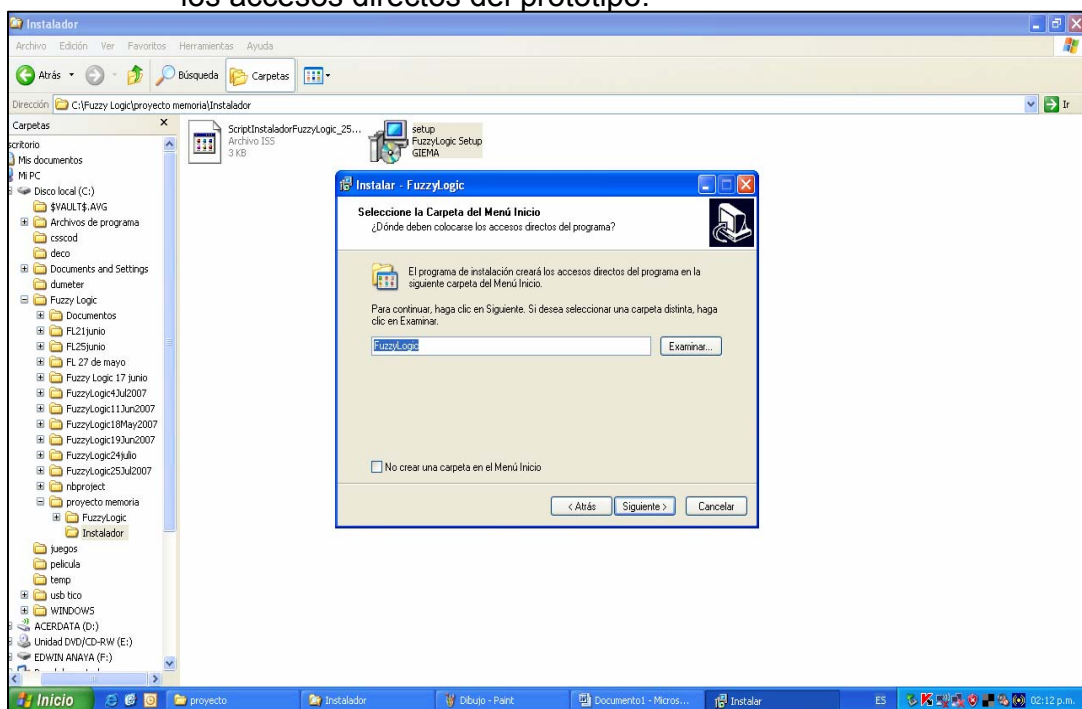
4. luego se procesa a introducir la contraseña que se ha generado para poder instalar el prototipo. La contraseña es 12345.



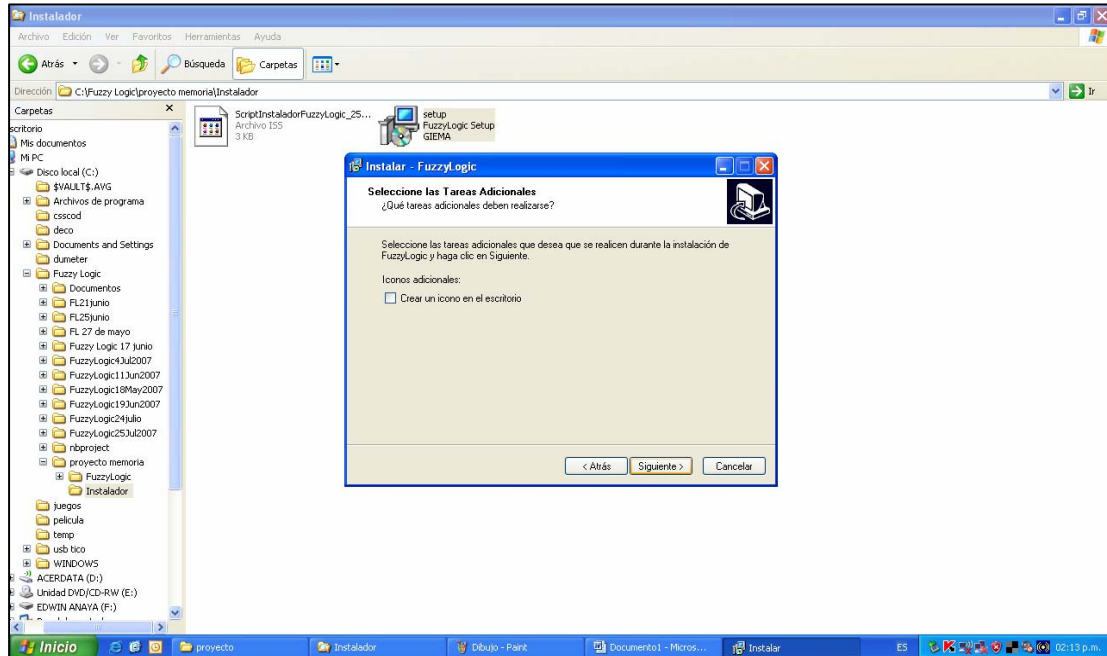
5. después de ingresar la contraseña se oprime el botón siguiente y saldrá una ventana que le indicara donde instalara la aplicación.



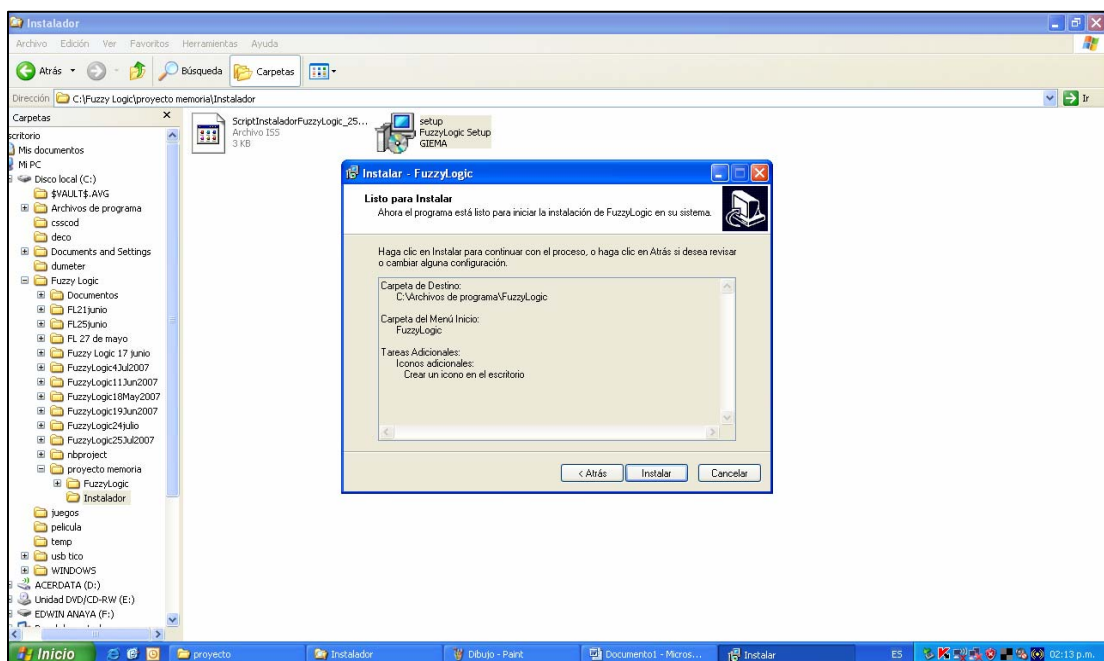
6. después se creara una carpeta en el menú inicio donde guardara los accesos directos del prototipo.



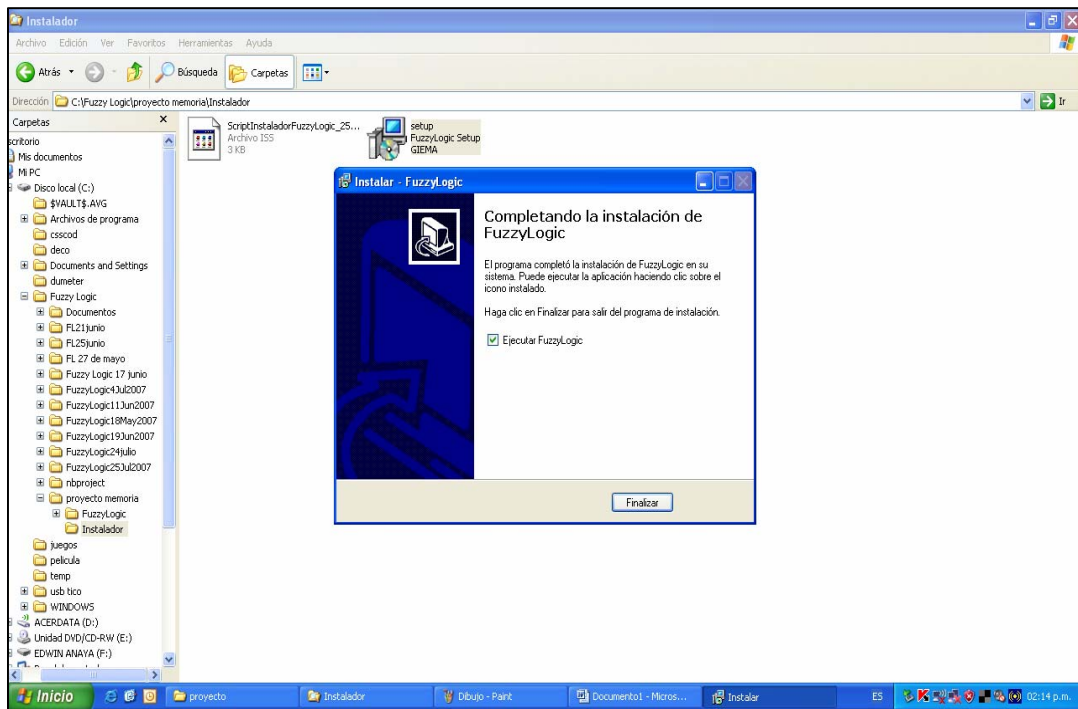
7. Luego se selecciona si desea crear un icono de acceso directo en el escritorio.



8. después de darle click en el botón siguiente aparece otra ventana con una descripción de lo que llevas hasta el momento, solo se debe dar click en el botón instalar para empezar la instalación.

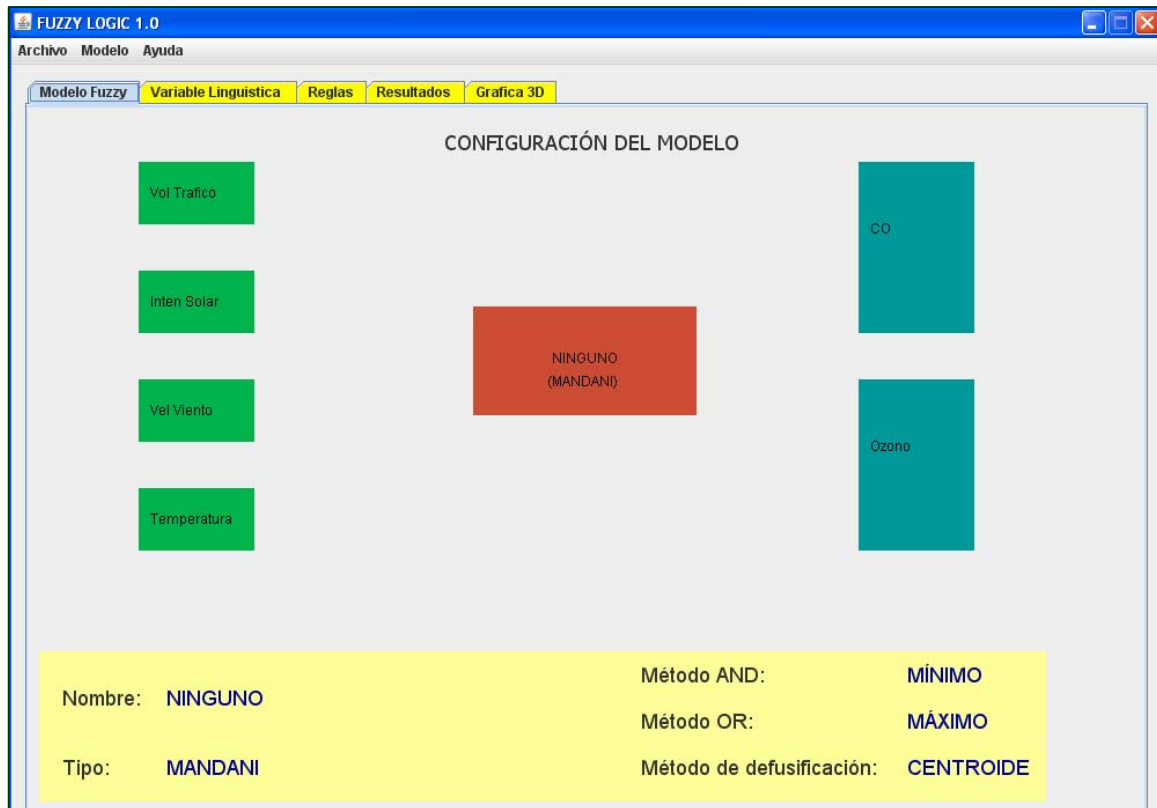


9. al final aparece una ventana indicando que ha finalizado el proceso de instalación y además se muestra un espacio para seleccionar si quiere abrir el prototipo de una vez. Este es todo el proceso que se debe seguir para el proceso de la instalación.

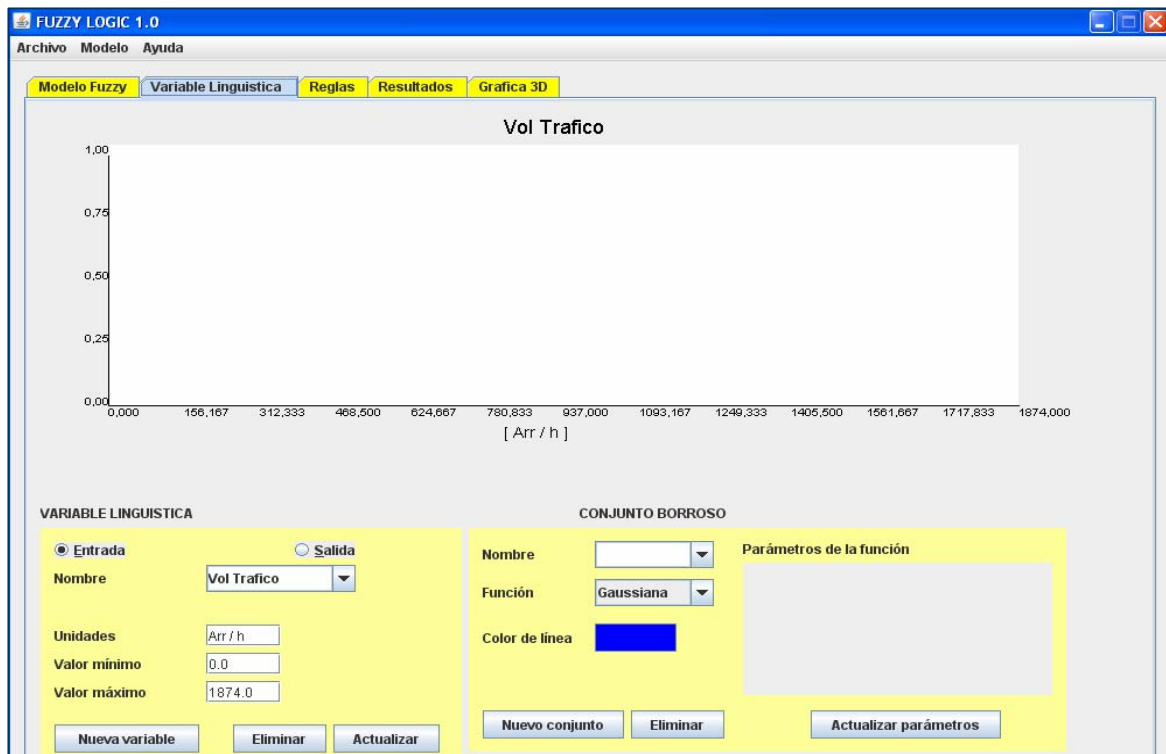


4. INTERFACES DEL PROTOTIPO Y PRESENTACIÓN DE LOS MENÚS.

Al abrir el prototipo aparece una interfaz principal de trabajo la cual contiene los tabs donde el usuario interactuara de una manera organizada siguiendo el orden de los mismos. El primer tab aparece la configuración del modelo, ahí aparece por defecto cuatro variables de entrada y dos de salida, ya queda a criterio del usuario si trabaja con estas o crea unas nuevas.



El prototipo presenta el siguiente entorno de trabajo:



A partir de esta pantalla empieza el desarrollo de la herramienta.

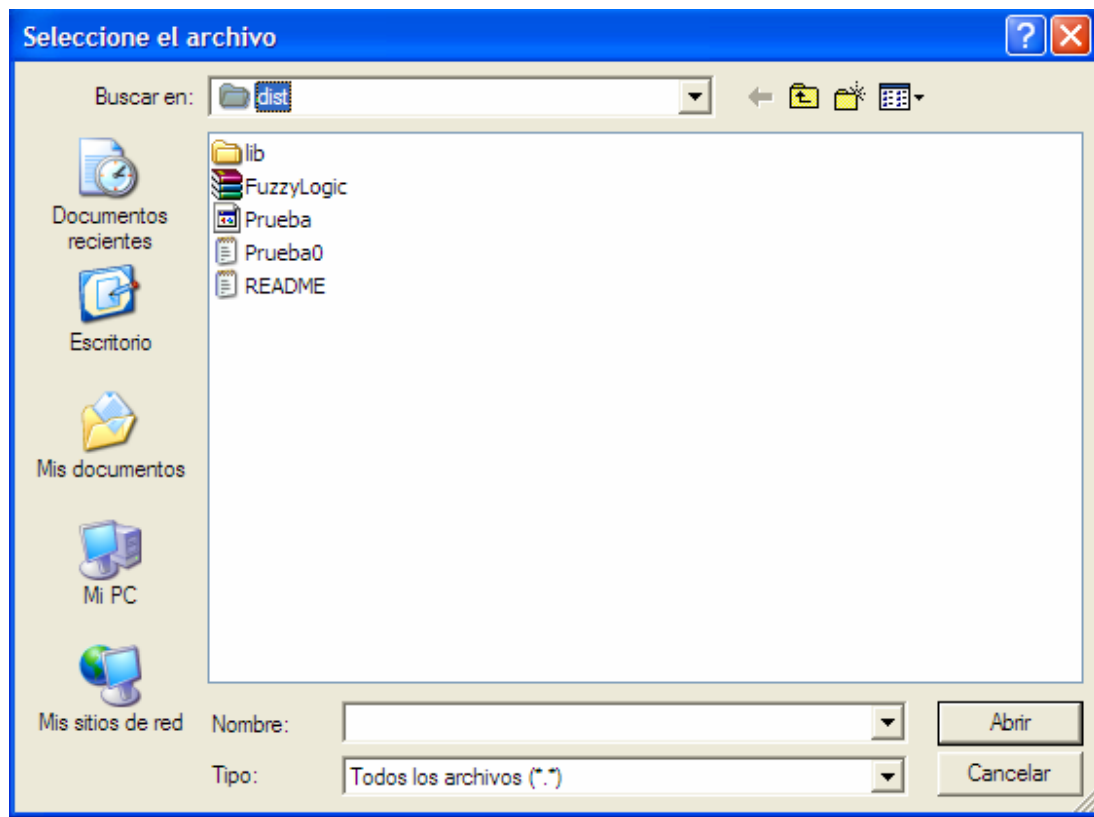
El tab que aparece es el de la variable lingüística, este se puede considerar como el más importante debido a que aquí es donde se recopilará toda la información de las variables lingüísticas y los conjuntos borrosos.

En la parte superior aparecen los menús básicos para el funcionamiento de la herramienta.

1. Archivo
2. Modelo
3. Ayuda

1. Archivo. Es la primera opción de la barra de menús y nos presenta el siguiente submenú.

abrir. Esta opción permite abrir un proyecto ya creado, que se encuentre almacenado en el disco duro.



Guardar. Es la opción que permite guardar un proyecto nuevo o uno ya modificado, se puede guardar donde el usuario desee.

Salir. Es la opción que termina la ejecución del programa.

2. Modelo. En este menú aparecen todos los tabs que aparecen debajo de los menús.

3. Ayuda. En este menú se encuentra la opción de ayuda y acerca de.

3.1 Ayuda. Es la opción que permite mostrar la ayuda al usuario de manera que se pueda consultar aspectos relativos al correcto funcionamiento del prototipo.

3.2 Acerca de: Esta opción presenta el nombre de los desarrolladores del prototipo.

5. EJEMPLO DETALLADO DE UN PROYECTO EN EL PROTOTIPO

Las variables utilizadas para realizar el ejemplo del prototipo son las siguientes:

Variables de entrada:

- Volumen del Tráfico
- Intensidad Solar
- Velocidad del viento
- Temperatura

Variables de salida:

- Monóxido de Carbono (CO)
- Ozono (O₃)

Los datos para estas variables son recopilados de la página de la CDMB (Corporación Autónoma Regional para la Defensa de la meseta de Bucaramanga) y de tesis de investigación.

Volumen del Tráfico:

- Valor mínimo: 0
- Valor máximo: 1874

Valores de la función de pertenencia gaussiana:

- Baja: $C = 0.8$ $\text{Sigma} = 255$
- Media: $C = 800$ $\text{Sigma} = 380$
- Alta: $C = 1836$ $\text{Sigma} = 344$

Intensidad Solar:

- Valor mínimo: 16
- Valor máximo: 920

Valores de la función de pertenencia Curva de Bell:

- Baja: $A = 83$ $B = 2$ $C = 17$
- Media: $A = 326$ $B = 5$ $C = 211$
- Alta: $A = 303$ $B = 3$ $C = 812$

Velocidad del Viento:

- Valor mínimo: 0.21
- Valor máximo: 3.5

Valores de la función de pertenencia gaussiana:

- Baja: $C = 0.2$ $\text{Sigma} = 0.6$
- Media: $C = 1.9$ $\text{Sigma} = 0.7$
- Alta: $C = 3.5$ $\text{Sigma} = 0.5$

Temperatura:

- Valor mínimo: 19
- Valor máximo: 26

Valores de la función de pertenencia trapezoidal:

- Bajo: $A=17$; $B=18$; $C=20$; $D=21$
- Medio: $A=20$; $B=21$; $C=23$; $D=24$
- Alto: $A=23$; $B=24$; $C=26$; $D=27$

Monóxido de Carbono (CO):

- Valor mínimo: 1
- Valor máximo: 4

Valores de la función de pertenencia gaussiana:

- Baja: $C = 0.9$ $\text{Sigma} = 0.5$

- Media: $C = 2.6$ $\text{Sigma} = 0.5$
- Alta: $C = 4$ $\text{Sigma} = 0.6$

Ozono (O3):

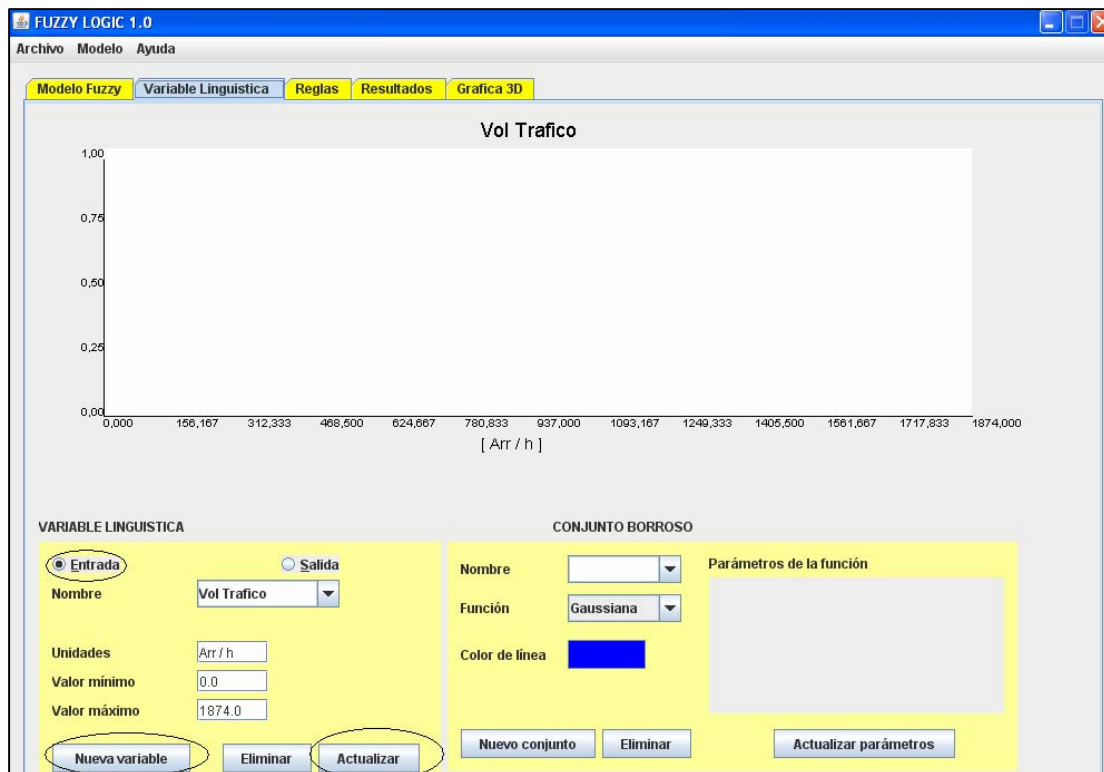
- Valor mínimo: 0
- Valor máximo: 38

Valores de la función de pertenencia gaussiana:

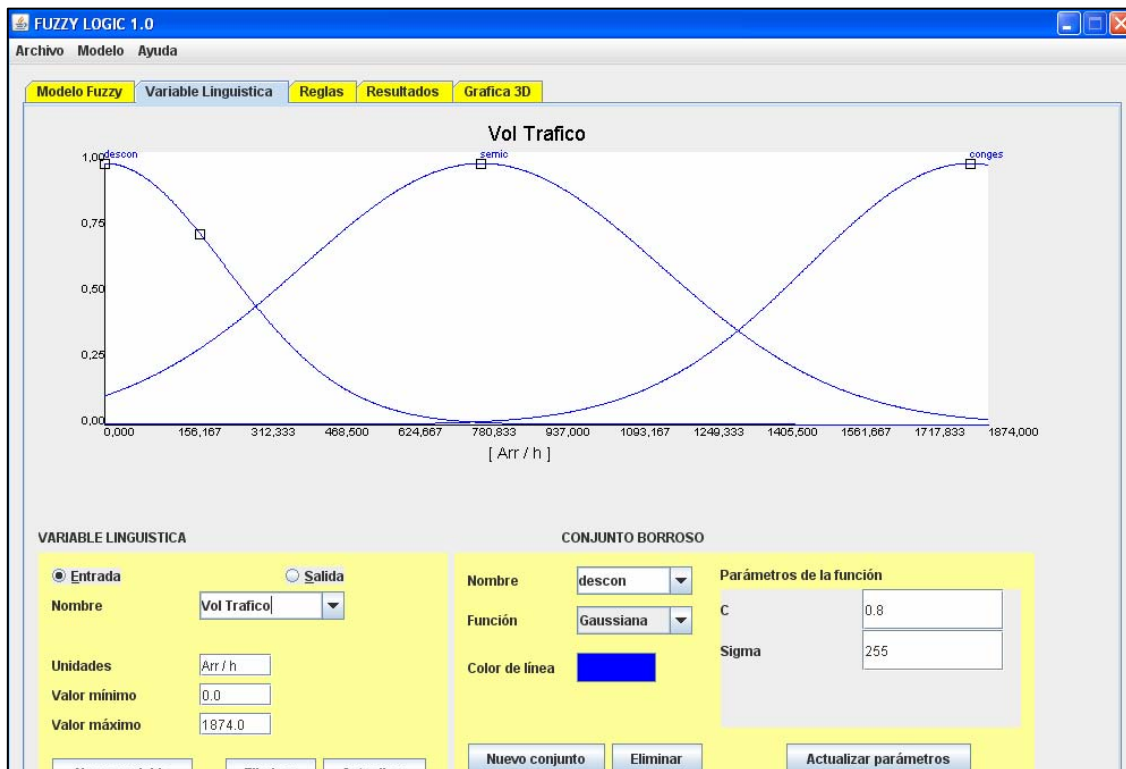
- Baja: $C = 0$ $\text{Sigma} = 7$
- Media: $C = 17$ $\text{Sigma} = 7.6$
- Alta: $C = 38$ $\text{Sigma} = 7$

Ya teniendo la información sobre las variables de entrada y salida se procede a ingresar estos datos en el prototipo.

En la primera pantalla que se muestra del prototipo se incluirán todos los datos de las variables (entrada y salida), se da click en el botón de nueva variable y luego se selecciona si es una variable de entrada o de salida, se introducen los respectivos datos y luego se presiona actualizar. Todos los datos se recibirán por teclado, no habrá nada para seleccionar o escoger, excepto en el tipo de función pero en los conjuntos borrosos.

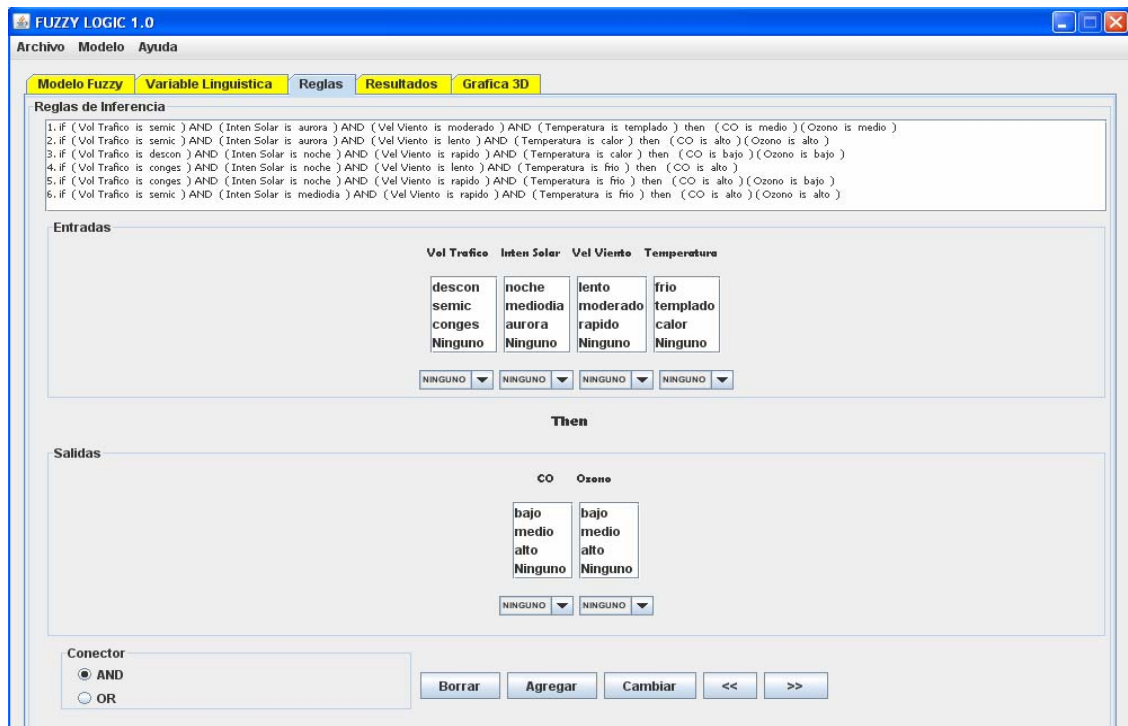


Luego de introducir los datos de la variable lingüística se procede a digitar los conjuntos borrosos para la variable creada. Aquí solo se digitará el nombre del conjunto borroso, el tipo de la función ya sea Gaussiana, Campana de Bell, Sigmoidal, Trapezoidal o Triangular. Y por último se escoge el color de la gráfica que representa el conjunto borroso.



Este procedimiento se realiza para cada una de las variables de entrada y salida, cada variable lingüística tiene como mínimo tres conjuntos borrosos.

La recopilación de los datos o información se hace en el primer tab de variable lingüística y luego se pasa al segundo tab que se llama reglas.

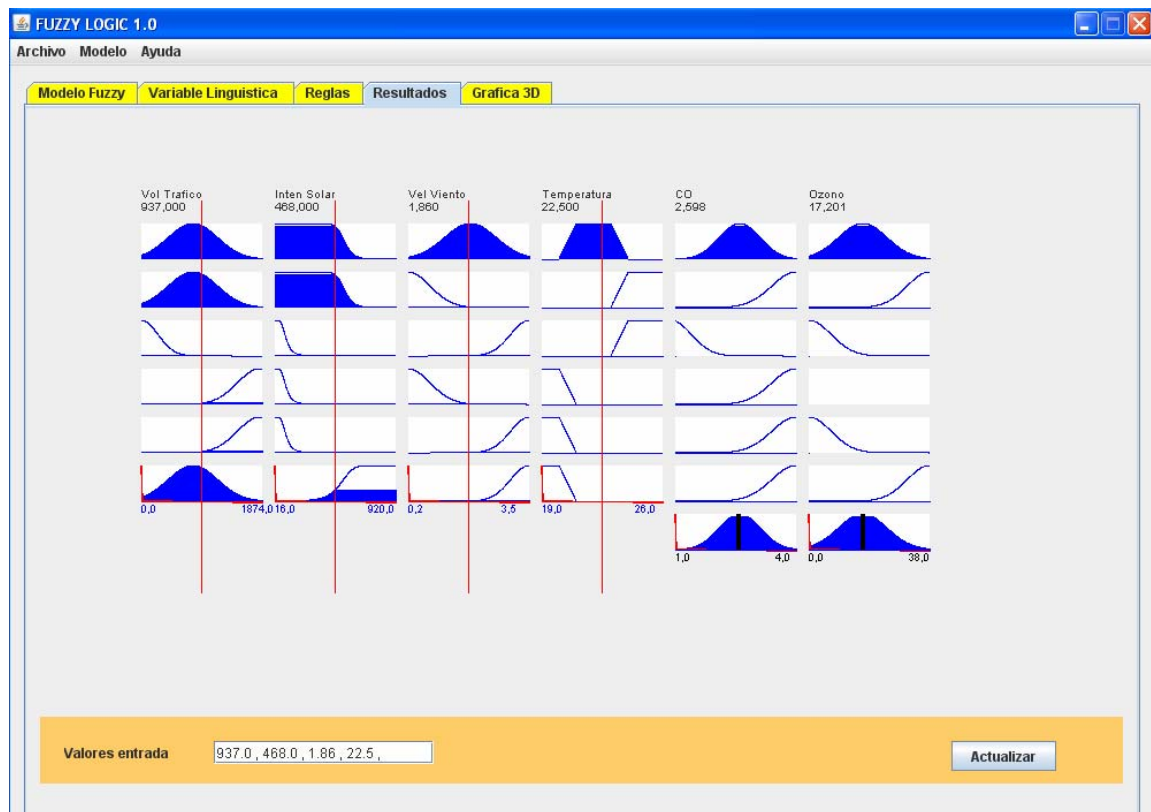


La anterior gráfica presenta una breve descripción de lo que es el tab de las reglas. En la parte superior esta el visualizador de las reglas, ahí es donde estarán todas las reglas que el usuario del prototipo va a agregar, estas reglas tienen que ser hechas con la ayuda de un experto en el tema.

En la sección de entradas se encuentra el conjunto de todas las variables de entrada con sus respectivos conjuntos borrosos. Y en la sección de salidas se encuentra todas las variables de salida con sus respectivos conjuntos borrosos. Estas dos secciones permiten visualizar un poco mejor todas las variables de entrada y salida, ya que facilita el trabajo para adicionar las reglas.

En este tab de reglas no existe un límite alguno para definir la cantidad de reglas, este criterio ya depende del problema que se este tratando y del experto.

El tab de resultados presenta de una manera gráfica cada una de las reglas adicionadas, ahí el usuario podrá analizar de una manera visual y mas fácil cada una de las reglas y así sacar las conclusiones necesarias para el problema planteado.



Los resultados de este ejemplo fueron comparados con los del toolbox de matlab y son muy similares con respecto a las gráficas y los resultados.