

**SOFTWARE DE APOYO PARA EL CONTROL INTELIGENTE BASADO EN LOGICA  
BORROSA "COBOR 1.0"**

**JUAN CARLOS REYES FIGUEROA  
JUAN JOSE LANDINEZ CRUZ**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERIAS FISICOMECAÑICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA  
2004**

**SOFTWARE DE APOYO PARA EL CONTROL INTELIGENTE BASADO EN LOGICA  
BORROSA "COBOR 1.0"**

**JUAN CARLOS REYES FIGUEROA  
JUAN JOSE LANDINEZ CRUZ**

**Trabajo de grado presentado como requisito  
parcial para optar a los títulos de:  
INGENIEROS DE SISTEMAS**

**DIRECTOR  
FERNANDO RUIZ DIAZ  
INGENIERO DE SISTEMAS, MsC**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERIAS FISICOMECAÑICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA  
2004**

## TABLA DE CONTENIDO.

	Pagina
INTRODUCCION	1
1 PRESENTACION DEL PROYECTO	2
1.1 JUSTIFICACION	2
1.2 OBJETIVO	4
1.2.1 OBJETIVO GENERAL	4
1.2.2 OBJETIVOS ESPECIFICOS	4
2 MARCO TEORICO	6
2.1 MARCO TEORICO GENERAL	6
2.1.1 SISTEMA DE CONTROL	6
2.1.1.1 SISTEMAS DE CONTROL EN LAZO ABIERTO	7
2.1.1.2 SISTEMAS DE CONTROL EN LAZO CERRADO O REALIMENTADOS	8
2.1.1.3 CLASIFICACION DE LAS TECNICAS DE CONTROL	9
2.2 MARCO TEORICO ESPECÍFICO	10
2.2.1 LOGICA CLASICA vs. LOGICA BORROSA	10
2.2.1.1 LOGICA CLASICA DE CONJUNTOS	11
2.2.1.1.1 CONJUNTOS CLASICOS	12
2.2.1.1.2 OPERACIONES CON CONJUNTOS	16
2.2.1.2 LOGICA BORROSA	18
2.2.1.2.1 CONJUNTOS BORROSOS	20
2.2.1.2.2 OPERACIONES CON CONJUNTOS BORROSOS	23
2.2.1.2.3 CARACTERISTICAS DE UN CONJUNTO BORROSO	27
2.2.1.2.4 RELACIONES ENTRE CONJUNTOS BORROSOS	27

2.2.1.2.5 PRINCIPIO DE EXTENSION	29
2.2.1.2.6 RAZONAMIENTO APROXIMADO	29
2.2.1.2.7 VARIABLE LINGUISTICAS	30
2.2.1.2.8 MODIFICADORES LINGUISTICOS	31
2.2.1.2.9 PROPOSICIONES BORROSAS	31
2.2.1.2.10 REGLAS BORROSAS	33
2.2.1.2.11 RAZONAMIENTO EN LOGICA BORROSA	34
3 SISTEMAS DE CONTROL INTELIGENTES	36
3.1 CONTROL INTELIGENTE	37
3.1.1 SISTEMAS DE CONTROL NEURO-BORROSOS	37
3.1.1.1 REDES NEURONALES	38
3.1.1.2 SISTEMAS NEURO-BORROSOS	43
3.1.1.2.1 LIMITACIONES DE LOS SISTEMA NEURO-BORROSOS	44
3.1.2 SISTEMAS DE CONTROL GENETICO-BORROSOS	46
3.1.2.1 INTRODUCCION SOBRE ALGORITMOS GENETICOS	46
3.1.2.2 SISTEMAS GENETICO-BORROSOS	49
3.1.2.2.1 SBRBs CON APRENDIZAJE GENETICO EN LA BASE DE REGLAS	51
3.1.3 SISTEMAS DE CONTROL BORROSO	52
3.1.3.1 SISTEMAS BASADOS EN REGLAS BORROSAS	54
3.1.3.1.1 TIPOS DE SISTEMAS BASADOS EN REGLAS BORROSAS	56
3.1.3.2 CONTROLADORES BORROSOS	58
3.1.3.2.1 ESTRUCTURA DE UN CONTROLADOR BORROSO	59
3.1.3.3 FASES DEL CONTROL BORROSO	60
3.1.3.3.1 EMBORRONADO	60
3.1.3.3.1.1 INTERFAZ DE EMBORRONADO	61
3.1.3.3.2 DESEMBORRONADO	63
3.1.3.3.3 BASE DEL CONOCIMIENTO	66
3.1.3.3.4 BASE DE DATOS	67
3.1.3.3.5 BASE DE REGLAS	67
3.1.3.4 ANALISIS DE CONTROLADORES BORROSOS	69
3.1.3.5 ERRORES EN LA SEMANTICA	70

4. DISEÑO Y SISTEMA COBOR 1.0	79
4.1 DISEÑO	79
4.1.1 DIAGRAMA DE CLASES	79
4.1.1.1 DIAGRAMA DE CLASES DEL PROYECTO COBOR 1.0	80
4.1.1.1.1 PROTOTIPO No 1	80
4.1.1.1.2 PROTOTIPO No 2	80
4.1.1.1.3 PROTOTIPO No 3	81
4.1.1.1.4 PROTOTIPO No 4	82
4.1.1.2 DIAGRAMA DE CLASES FINAL	83
4.1.2 DIAGRAMA DE CASOS DE USO.	84
4.1.2.1 DIAGRAMA DE CASOS DE USO PROYECTO COBOR 1.0.	84
4.1.3 PRUEBAS DE INGENIERÍA DE SOFTWARE.	98
4.1.3.1 PRUEBAS ALFA.	98
4.2 SISTEMA COBOR 1.0.	99
4.2.1 ARQUITECTURA DE COBOR 1.0.	99
4.2.2 CARACTERÍSTICAS DEL SISTEMA	100
4.2.3 CARACTERÍSTICAS DE LOS COMPONENTES.	100
4.2.4 PARÁMETROS ESPECÍFICOS.	102
4.2.5 REQUERIMIENTOS DE HARDWARE Y SOFTWARE.	103
5. APLICACIONES.	104
5.1. SISTEMAS DE CONTROL BASADOS EN REGLAS BORROSAS.	104
5.2. APLICACIONES EN LA INGENIERÍA.	105
5.2.1 SISTEMA DE CONTROL DEL CALOR DE UNA CASA.	116
5.2.2 ROBOTS MÓVILES Y NAVEGACIÓN AUTÓNOMA.	108
5.2.3 PLANTA DE CALCINACIÓN DE LA INDUSTRIA DEL NÍQUEL.	112
5.2.4 CONTROL BORROSO DE UN APARATO DE AIRE ACONDICIONADO.	113
CONCLUSIONES.	119
RECOMENDACIONES FINALES.	122
BIBLIOGRAFIA.	123
REFERENCIA BIBLIOGRAFICA.	126
ANEXO	128

## LISTADO DE FIGURAS

	Pagina
Figura 1. Significancia y Precisión en el Mundo real.	3
Figura 2. Sistema de Control.	6
Figura 3. Sistema de Control de Lazo Abierto.	7
Figura 4. Sistema de Control de Lazo Cerrado.	8
Figura 5. Lógica Borrosa vs. Lógica Clásica.	10
Figura 6. Diagrama de Venn.	17
Figura 7. Función característica de un conjunto clásico.	21
Figura 8. Conjunto Gente Joven.	23
Figura 9. Intersección de un Conjunto Borroso.	24
Figura 10. Unión de un Conjunto Borroso.	24
Figura 11. Complemento de un Conjunto Borroso.	25
Figura 12. Principio de Extensión.	28
Figura 13. Variables Lingüísticas.	30
Figura 14. Modelo de una Neurona Biológica.	40
Figura 15. Modelo de una Neurona Artificial.	40
Figura 16. Perceptron.	41
Figura 17. Función XOR Generada.	43
Figura 18. Sistema Neuro Borroso.	44
Figura 19. Estructura de Anfis.	45
Figura 20. Ciclo de Evolución.	49
Figura 21. Diseño de un Sistema Genético Borroso.	49
Figura 22. SBRBs con Aprendizaje Genético de la Base de Reglas.	51

Figura 23 Estructura de un Controlador Borroso.	60
Figura 24 Función Triangular.	62
Figura 25 Función Trapezoidal.	62
Figura 26. Función Gaussiana.	62
Figura 27 Conjunto Borrosos Tipo Singleton.	63
Figura 28 Conjunto Borroso Tipo Triangular.	64
Figura 29 Método del Centro de Área.	64
Figura 30 Agregación de Particiones Activas.	65
Figura 31 Método del Centro de Mayor Área.	66
Figura 32 Método del Primer y Ultimo Máximo.	67
Figura 33 Base de Datos.	68
Figura 34 Consistencia de Reglas.	71
Figura 35 Completitud.	72
Figura 36 Continuidad.	73
Figura 37 Trayectoria de Disparo de Reglas Sistema Inestable.	76
Figura 38 Trayectoria de Disparo de Reglas Sistema Estable.	76
Figura 39 Errores en la Semántica "Intervalo"	77
Figura 40 Errores en la Semántica "Cobertura"	78
Figura 41 Normalidad en los Conjuntos."Grados de Pertenencia".	78
Figura 42 Normalidad en los Conjuntos."Extremos del Universo".	79
Figura 43 Distinguibilidad de los Conjuntos.	79
Figura 44. Diagrama de Clases del Prototipo No 1.	81
Figura 45. Diagrama de Clases del Prototipo No 2.	82
Figura 46. Diagrama de Clases del Prototipo No 3.	82
Figura 47. Diagrama de Clases del Prototipo No 4.	83
Figura 48. Diagrama de Clases Final.	84
Figura 49. Diagrama de casos de uso de la aplicación COBOR 1.0	85
Figura 50. Diagrama de Casos de Uso: Variable Borrosas	88
Figura 51. Diagramas Casos de Uso: Motor de Inferencia.	92
Figura 52. Diagramas Casos de Uso: Analizador.	94
Figura 53. Control de Calor de una Casa.	108

Figura 54. Robot Autónomo PILAR.	111
Figura 55. Horno de Calcinación.	114
Figura 56. Esquema de bloques	115
Figura 57. Controlador de Aire Acondicionado.	117
Figura 58. Elementos de la cabina de aire acondicionado.	119
Figura 59 Entorno de trabajo Cobor 1.0.	128
Figura 60 Ventana Nuevo Proyecto	129
Figura 61. Cuadro de dialogo abrir proyecto existente	130
Figura 62. Cuadro de dialogo Guardar Proyecto como	131
Figura 63. Ventana Información del proyecto	131
Figura 64. Mensaje de confirmación Guardar Cambios	132
Figura 65. Ventana nueva Variable	133
Figura 66. Ventana Editar Variable	134
Figura 67. Ventana Eliminar Variable	134
Figura 68. Ventana Crear Todas las Reglas	135
Figura 69. Ventana Crear Reglas Una por Una	136
Figura 70. Ventana Calcular	137
Figura 71. Ventana Paso a Paso	138
Figura 72. Ventana Matriz de Acciones	139
Figura 73. Ventana Resultado Grafico	140

## RESUMEN

TITULO:

“SOFTWARE DE APOYO AL CONTROL INTELIGENTE BASADO EN LOGICA BORROSA  
-COBOR 1.0 –“

AUTORES:

JUAN CARLOS REYES FIGUEROA  
JUAN JOSE LANDINEZ CRUZ.

PALABRAS CLAVES:

Control borroso, Diseño de controladores borrosos, Ayudas para el diseño de Controladores borrosos.

CONTENIDO:

En el campo específico de los sistemas de control, es frecuente el caso de procesos o plantas en las que controladores diseñados según otros patrones (clásicos, adaptables.) no dan los resultados deseados. Se trata de procesos de elevada complejidad y, en la inmensa mayoría de los casos operadores humanos diestros y experimentados (!expertos!) alcanzan resultados satisfactorios en su dirección. Cuando se investiga el proceder de esos expertos, se concluye de inmediato que, tanto la información acerca del estado del proceso como sus acciones de control, las procesan y manejan de un modo básicamente cualitativo, usando su intuición, habilidad heurística y experiencia.

La importancia de la lógica borrosa desde el punto de vista de la teoría del control de procesos, es la de proveer soporte cuándo se quiere traducir el conocimiento heurístico experimentado, expresado en frases lingüísticas imprecisas a algoritmos numéricos. Esto ya que esta basada en la teoría de conjuntos que posibilita imitar el comportamiento de la lógica humana. El término "borroso" procede de la palabra inglesa "fuzzy" que significa "confuso, difuso, indefinido.

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.

## **SUMMARY**

**TITLE:**

SUPPORT SOFTWARE TO THE INTELLIGENT CONTROL BASED ON FUZZY LOGIC  
- COBOR 1.0 -

**AUTHORS:**

JUAN CARLOS REYES FIGUEROA  
JUAN JOSE LANDINEZ CRUZ.

**KEY WORDRS:**

Control, Fuzzy Control Design, Fuzzy Control Design Aided.

**CONTENT:**

In the specific field of the control systems, it is common to find processes in those that controllers designed according to other patterns (classic, adaptive) doesn't work properly or definitely doesn't work. Those are processes of high complexity where the human operators that are called experts in the immense majority of the cases skillful and experienced make a great roll, reaching satisfactory results managing properly almost all situations. And if we make an investigation on the proceedings carried out by those experts, surely we concludes immediately that, so much the information about the state of the process as their control actions, are processed and managed in a qualitative way, using their intuition, heuristic ability and experience.

In The control processes theory the fuzzy logic theory is important because it gives a support when it is wanted to translate the experienced heuristic knowledge, expressed in imprecise linguistic sentences to numeric algorithms. It is because the fuzzy logic is based on the terms of the sets theory that facilitates to imitate the behavior of the human logic.

COBOR 1.0 is a software tool that gives a support at time to design and develop control systems, using the fuzzy logic theory as an adaptative technique. First of all the monitored variables came into the software, then it are converted into fuzzy variables, finally the action is determined to take a decision inferred by the result of an inference on an data rule base, that was defined by the process expert.

## INTRODUCCION

El presente documento contiene el informe final del proyecto titulado "Software de apoyo para el control inteligente basado en Lógica Borrosa COBOR 1.0". Este proyecto ha surgido en el grupo SISTEMIC 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 en la actualidad se encuentran en pleno desarrollo a nivel mundial.

El problema del control en los procesos de ingeniería ha sido uno de los factores fundamentales en el desarrollo de nuevas tecnologías de razonamiento aproximado, esto con el fin de reemplazar o en su defecto dar apoyo a los responsables del control de procesos. Dado que el diagnostico de los expertos no siempre va a ser el mas acertado en procesos de este tipo, el objetivo es crear una herramienta capaz de dar una aproximación a la solución en los casos donde se requiera una vasta experiencia y sapiencia de las personas encargadas del control de procesos.

Ya que la lógica borrosa es una opción para realizar control en los procesos de alta complejidad que se manejan en las diferentes áreas de las ingenierías, se pretende crear un software interdisciplinario, que se pueda utilizar en los diferentes campos de la ingeniería y que además muestre el proceso que se sigue para obtener la matriz de acciones que es el corazón del controlador Lógico Borroso.

## **1. PRESENTACION DEL PROYECTO.**

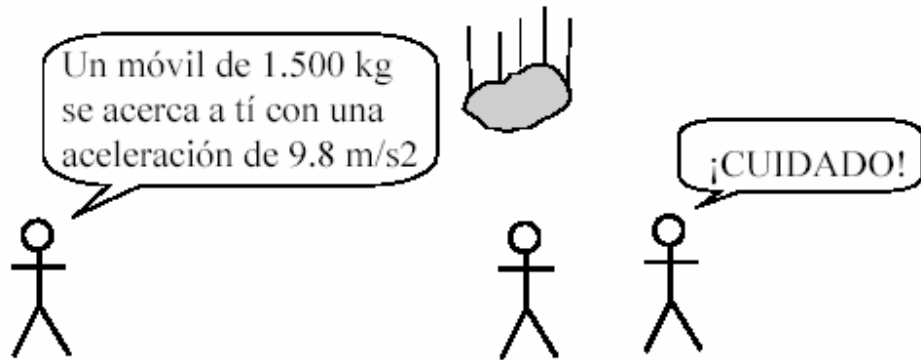
### **1.1 JUSTIFICACION.**

En los años recientes se han venido incorporando sistemas de automatización, debido a la necesidad de rebajar costos y hacer sencillas las tareas para el operador de procesos complejos. En un principio surgieron los controladores convencionales, usados aún en la industria, que presentan el problema, que necesitan un modelo cuantitativo del proceso que se va a automatizar lo cual no siempre es posible, especialmente en procesos complejos, en los que el experto juega un papel muy importante usando su intuición, su habilidad heurística y su experiencia. Una de las técnicas efectivas para resolver este problema, es la teoría de los conjuntos borrosos que ha hecho posible establecer una modalidad del llamado "Control Inteligente".

La importancia de la lógica borrosa desde el punto de vista de la teoría del control de procesos, es la de proveer un buen soporte cuando se quiere traducir el conocimiento heurístico experimentado y expresado en frases lingüísticas imprecisas, a algoritmos numéricos.

La lógica borrosa es un área fascinante de la investigación, ya que hace un buen trabajo compaginando significancia y precisión. En la figura 1 vemos un claro ejemplo de lo que significan estas dos palabras.

Figura 1 Significancia y precisión en el mundo real.



De las muchas formas de hacer funcionar la caja negra, frecuentemente la mejor opción es la lógica borrosa. Como alguna vez dijo Lotfi Zadeh, considerado el padre de la lógica borrosa: "En casi todos los casos se puede construir el mismo producto sin lógica borrosa, pero ésta es más rápida y barata."

## **1.2 .OBJETIVO.**

### **1.2.1 OBJETIVO GENERAL.**

- Facilitar a los expertos en control de procesos complejos definir un procedimiento de control en línea soportado en una matriz de acciones a realizar generada a partir de operaciones lógicas difusas.
- Brindar 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 al menos dos variables de control y una de acción. Se intentará incluir más de una variable de acción, proponiendo un criterio de decisión para definir la operación a tomar en más de dos dimensiones.
- Ilustrar didácticamente el algoritmo de solución basado en Lógica Borrosa aplicada al control de procesos, para la comprensión y entendimiento de los usuarios que utilizarán la herramienta

### **1.2.2 OBJETIVOS ESPECIFICOS.**

Diseñar e implementar una herramienta software que permita a los expertos en control crear el controlador de procesos basado en lógica borrosa con las siguientes características:

- Que facilite la definición de los conjuntos borrosos y sus particiones correspondientes a variables de estado y de acción.

- Que permita visualizar y modificar los conjuntos definidos y almacenados con anterioridad.
- Que facilite la construcción de una base de conocimiento definida por reglas de la forma IF [condición] THEN [acción], donde la condición se forme con expresiones lógicas del tipo NOT, AND, OR, para todas las variables.
- Que produzca una matriz de control n-dimensional, donde n es el número de variables de estado a monitorear.
- Que ilustre las operaciones borrosas que soportan la construcción de la matriz de control.
- Que muestre el proceso de solución paso a paso de forma que el interesado pueda ver la totalidad del proceso para llegar a la solución final.

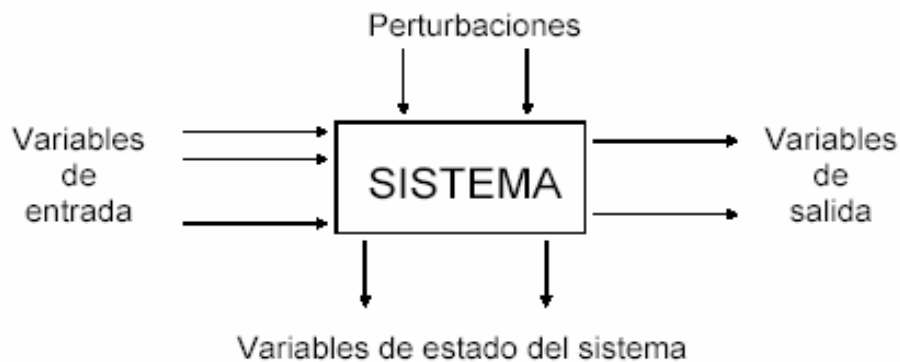
## 2. MARCO TEORICO.

### 2.1. MARCO TEORICO GENERAL.

#### 2.1.1 SISTEMA DE CONTROL.

Un sistema de Control es un Conjunto o combinación de componentes que actúa conjuntamente y que cumple un determinado objetivo.

Figura 2 Sistema de Control.



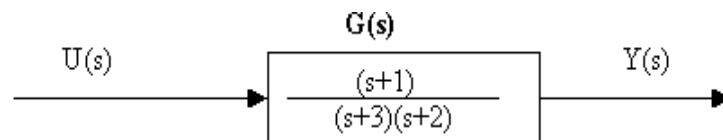
Las ventajas de utilizar un sistema de control son las siguientes:

- Las técnicas de control automático tienen un campo prácticamente ilimitado de aplicación.

- Es útil contar con sistemas capaces de mantener todos los parámetros “controlados” sin la intervención humana.
- En ocasiones se consigue optimizar la evolución del proceso.
  - ✓ Situaciones de elevada complejidad
  - ✓ Situaciones en las que se debe operar en tiempo de respuesta corto
- Pueden eliminar fallos (distracciones, cansancio, tensión...).

### 2.1.1.1 SISTEMA DE CONTROL EN LAZO ABIERTO.

Figura 3 Sistema de Control de Lazo Abierto.



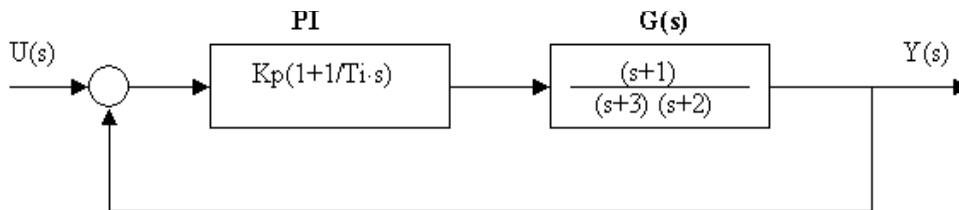
Son aquellos en la que la única señal que ejerce una acción de control sobre el sistema es la que entra al sistema (**señal de referencia o consigna**), en estos casos la salida no actúa sobre la entrada, y la salida no influye en la acción de control.

- La acción de control es independiente de la salida.
- No hay realimentación.
- A cada entrada de referencia le corresponde una condición de operación fija.
- La precisión depende de la calibración.
- Problemas con las perturbaciones.

Este tipo de controladores genera secuencias de instrucciones como respuesta a las distintas órdenes o variables de entrada.

### 2.1.1.2 SISTEMAS DE CONTROL EN LAZO CERRADO O REALIMENTADOS.

Figura 4 Sistema de Control de Lazo Cerrado.



Son aquellos en los que la acción de control depende tanto de la entrada de referencia como del valor de la salida. En estos casos, la salida del sistema actúa sobre la entrada para mantener su valor dentro de los límites fijados. Cuando la salida ejerce influencia sobre la entrada, se puede decir que hay realimentación. Los sistemas de control de lazo cerrado son sistemas realimentados. Mediante la realimentación se corrigen las variaciones existentes entre el valor real de la salida y el valor deseado.

- Mantiene una relación entre la entrada y la salida, comparándolas y utilizando la diferencia como medio de control.
- Comportamiento adecuado en presencia de perturbaciones.
  - ✓ Sistema de control de la temperatura de una habitación.
  - ✓ Sistema de control de la velocidad de un vehículo.
- Tipos de sistemas de control realimentados:
  - ✓ Reguladores (termostato).
  - ✓ Servomecanismos (brazo de robot).

### 2.1.1.3 CLASIFICACIÓN DE LAS TÉCNICAS DE CONTROL.

Entre las principales técnicas de control tenemos:

### **Sistemas de control continuo.**

- Sistemas que operan con señales continuas.
- Normalmente la función de control se implementa con circuitos electrónicos.

### **Sistemas de control digital.**

- Utilizan tecnología digital (Mayor flexibilidad en diseño).
- Un controlador digital para plantas continuas necesita conversión analógica-digital (y viceversa).
- “Problemas” de retardos y longitud de palabra inapreciables con las mejoras en la tecnología digital.
- Mayor capacidad para almacenar y manipular datos.
- Permite la inclusión de procesos de aprendizaje, control adaptativo, conocimiento experto y otros conceptos avanzados.

### **Sistemas de eventos discretos.**

(Control secuencial, control lógico programable, control dinámico de eventos discretos).

Acciones de control determinadas como respuesta a las características secuenciales y combinaciones observadas de un conjunto de órdenes y condiciones sensoriales.

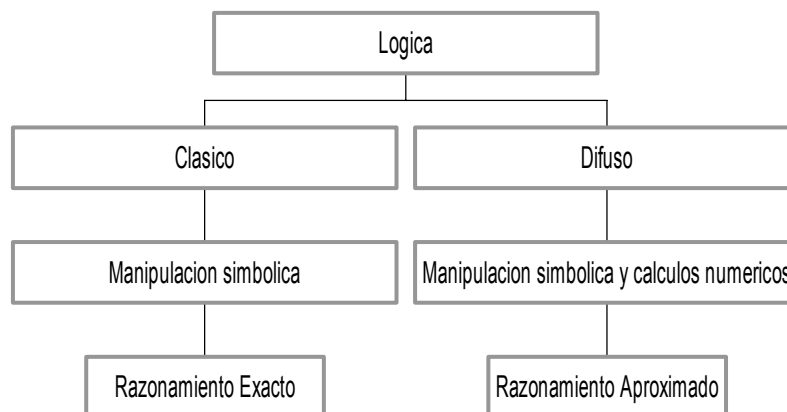
- Entradas y realimentación suelen ser binarias.
- Salida también suele ser binaria.
- Se diseñan mediante el desarrollo de una tabla de transición de estados.

## 2.2 MARCO TEORICO ESPECÍFICO.

### 2.2.1 LOGICA CLÁSICA vs. LOGICA BORROSA.

Aunque existen mucha similitud entre la lógica tradicional y la lógica borrosa en la parte operacional (Figura 5), también existen grandes diferencias que han dado pie para nuevos aportes que han significado grandes avances en todos los campos de la ingeniería.

Figura 5 Lógica Clásica vs. Lógica Borrosa.



La Lógica Borrosa es una técnica de la inteligencia computacional que permite trabajar información con alto grado de imprecisión. En esto se diferencia de la lógica convencional que trabaja con información bien definida y precisa.

El concepto de Lógica Borrosa fue concebido por Lofti Zadeh, un profesor de la Universidad de California en Berkeley, quien inconforme con los conjuntos clásicos

("crisp-sets") que sólo permiten dos opciones, la pertenencia o no de un elemento a dicho conjunto, la presentó como una forma de procesar información permitiendo pertenencias parciales a unos conjuntos, que en contraposición a los clásicos los denominó Conjuntos Borrosos ("fuzzy-sets"). El concepto de conjunto borroso fue expuesto por Lofti Zadeh en un artículo hoy clásico en la literatura de la lógica borrosa, en el año de 1965 titulado "Fuzzy Sets" y fue publicado en la revista "Information and Control". El mismo Zadeh publica en 1971 el artículo, "Quantitative Fuzzy Semantics", en donde introduce los elementos formales que acabarían componiendo el cuerpo de la doctrina de la lógica borrosa y sus aplicaciones tal como se conocen en la actualidad.

Pocos años después (1974), el británico Ebrahim Mandami demuestra la aplicabilidad de la lógica borrosa en el campo del control, desarrollando el primer sistema de control borroso (Fuzzy Control) práctico, la regulación de un motor de vapor. Las aplicaciones de la lógica borrosa en el control no se pudieron implementar con anterioridad a estos años debido a la poca capacidad de cómputo de los procesadores de la época.

### **2.2.1.1 LOGICA CLÁSICA DE CONJUNTOS.**

#### **Historia.**

La lógica de conjuntos fue creada por George Cantor, aunque George Boole dio los primeros pasos en su libro "Investigations of the laws of thought". El concepto de infinito fue tratado por Zenón de Elea y sus célebres paradojas. Cantor publicó varios artículos entre 1867 y 1871 sobre teoría de números de gran calidad, pero nada indicaba que su autor cambiaría el curso de la matemática. Tiempo después empezó a trabajar en series trigonométricas y aquí aparecen las primeras ideas sobre teoría de conjuntos. Cantor demostró que los números reales algebraicos se podían poner en correspondencia uno a uno con los números naturales pero que esto no se podía hacer con los números reales

(que incluyen, además de los reales algebraicos, los trascendentes). El primer intento de axiomatizar la Teoría de Conjuntos la hizo Zermelo en 1908. Después lo intentaron Fraenkel, Von Neumann, Bernays y Godel. Godel mostró las limitaciones de cualquier teoría axiomática.

La mejor prueba de que la teoría de conjuntos no ha logrado unificar a las matemáticas es que éstas se han ramificado en áreas muy diferenciadas, como la aritmética, el álgebra, la trigonometría y geometría. También se han separados distintos campos como el cálculo, la topología, la teoría de conjuntos, la teoría de los números y la estadística.

#### **2.2.1.1.1 CONJUNTOS CLÁSICOS.**

##### **Notación.**

Ordinariamente se usan letras mayúsculas para representar los conjuntos que incluiremos sus elementos dentro de llaves separados por comas,  $\{ \}$ . El símbolo  $\in$  significa (es elemento de). Análogamente,  $\notin$  significa (no es elemento de).

##### **Conjuntos Iguales.**

Se dice que dos conjuntos S y T son iguales si cada elemento de S es elemento de T y viceversa. Se escribe  $S = T$ . Se usa el signo de igualdad para indicar que dos símbolos representan al mismo conjunto.

##### **Conjuntos Vacíos.**

Un conjunto sin elementos recibe el nombre de conjunto vacío o conjunto nulo y se representa por  $[ ]$  o por  $\emptyset$ . Es útil tener el concepto de un conjunto sin elemento.

### **Subconjuntos.**

Se dice que un conjunto  $S$  es subconjunto  $T$ , si todos los elementos de  $S$  lo son  $T$ . El símbolo  $\subseteq$  se lee (es subconjunto de).

### **Conjuntos Equivalentes.**

Cuando los elementos de un conjunto se corresponden con los de un segundo conjunto de modo que cada elemento de cada conjunto tenga uno, y solo uno, asociado en el otro conjunto, decimos que hay una correspondencia uno a uno entre ambos conjuntos.

Dos conjuntos que se pueden poner en correspondencia uno a uno entre sí, se dice que son equivalentes. Si  $A$  es equivalente a  $B$ , se escribe  $A \sim B$ .

### **Cardinalidad de un Conjunto.**

Contar es el proceso por el cual ponemos en correspondencia los elementos de un conjunto con algún subconjunto propio de  $N$ , comenzando con 1 y usando los elementos de  $N$  en orden y sin saltar ninguno. Un subconjunto así se llama subconjunto estándar de  $N$ . Por ejemplo, el subconjunto estándar de  $N = \{1,2,3,4\}$  es equivalente a  $a = \{a,b,c,d\}$ . Decimos entonces que  $S$  tiene cuatro elementos. Esto lleva a la siguiente definición: "Cuando un conjunto  $S$  se equipara con un subconjunto estándar de  $N$ , el último elemento de  $N$  usado se llama cardinalidad del conjunto  $S$  y se denota por  $n(S)$ ".

Dos números enteros no negativos  $m$  y  $n$ , son iguales si ambos son la cardinalidad del mismo conjunto o de conjuntos equivalentes. En tal caso, escribimos  $m = n$ .

### **Conjuntos Finitos e Infinitos.**

Si es posible encontrar un subconjunto estándar de  $N$  que se puede hacer corresponder uno a uno con un conjunto dado  $S$ , o si  $S$  es el conjunto vacío, se dice que  $S$  es finito. En

caso contrario, se dice que es infinito. Nótese que, sin embargo, si hay un equiparamiento de  $N$  con uno de sus subconjuntos, que no es un subconjunto estándar como el conjunto de los números pares, vemos que  $N$  se puede poner en correspondencia con un subconjunto propio de sí mismo. Esto solo se puede hacer en un conjunto infinito.

### **CONCEPTOS BÁSICOS SOBRE LOS CONJUNTOS CLÁSICOS.**

Un Conjunto es cualquier colección de objetos el cual puede ser tratado como una entidad y un objeto de la colección se dice que es un elemento o miembro del conjunto. Dado un objeto  $x$ , y un conjunto  $S$ , si  $x$  es un elemento del conjunto  $S$ , lo podemos escribir como " $x \in S$ "; si  $x$  no es un elemento del conjunto  $S$ , podemos escribirlo como " $\neg(x \in S)$ " o también " $x \notin S$ ". Los términos conjunto, colección y clase son usados como sinónimos, así como también los términos elemento o miembro.

Casi cualquier cosa puede ser tratada como un conjunto, viéndolo desde un punto de vista muy matemático, lo que se trata de ilustrar con los siguientes ejemplos.

- ✓ El conjunto de enteros no negativos menores que 4. Éste es un conjunto finito con cuatro miembros: 0, 1, 2 y 3.
- ✓ El conjunto de enteros mayores que 3. Como es de suponerse, éste se trata de un conjunto infinito y no hay ninguna dificultad para definir cualquiera de los miembros de éste conjunto.

Dado que un conjunto es caracterizado por sus miembros, un conjunto puede ser especificado por declaración cuando un objeto está en el conjunto. Un conjunto finito puede ser especificado explícitamente por una lista de sus elementos. Los elementos de la lista deben ser separados por comas y la lista encerrada en llaves  $\{ \}$ , como lo muestran los siguientes ejemplos:

- ✓ El conjunto que contiene los elementos  $A$ ,  $B$  y  $C$  está denotado por  $\{A, B, C\}$ .

Los elementos de un conjunto infinito no pueden ser listados explícitamente; en consecuencia, necesitamos una forma para describirlos implícitamente. La especificación implícita frecuentemente es hecha por el significado de predicados con una variable libre. El conjunto es definido de manera que los elementos del universo establecido por el conjunto hagan el predicado verdadero. De aquí, si  $P(x)$  es un predicado con una variable libre, el conjunto  $\{x \mid P(x)\}$  denota el conjunto  $S$  tal que  $c \in S$ , si y sólo si,  $P(c)$  es verdadero.

Si un conjunto es finito pero muy largo como para listarse fácilmente o si es un conjunto infinito, las elipses suelen ser usadas para especificar implícitamente un conjunto. Las siguientes especificaciones usan elipses para caracterizar una lista de los elementos de un conjunto.

- ✓ El conjunto de enteros del 1 al 50 es especificado por  $\{1, 2, 3, \dots, 50\}$
- ✓ El conjunto de enteros pares no negativos es especificado por  $\{0, 2, 4, 6, \dots\}$

En un desarrollo más formal de la teoría de conjuntos, el siguiente axioma es usado para establecer que los conjuntos son completamente especificados por sus elementos. El axioma nos sirve como una definición de igualdad de conjuntos.

Axioma de Extensión: Dos conjuntos  $A$  y  $B$  son iguales si y sólo si tienen los mismos elementos.

El axioma de extensión puede ser expresado en notación lógica de dos maneras:

$$A = B \Leftrightarrow \forall x [x \in A \Leftrightarrow x \in B]$$

$$A = B \Leftrightarrow \{ \forall x [x \in A \Rightarrow x \in B] \wedge \forall x [x \in B \Rightarrow x \in A] \}$$

El axioma de extensión declara que si dos conjuntos tienen los mismos elementos, aún sin considerar como están especificados, son iguales. Es decir, si un conjunto es

especificado explícitamente con una lista, el orden en el que esté listado es irrelevante. Por ejemplo: el conjunto denotado por  $\{A, B, C\}$  es el mismo que (igual a) el conjunto denotado por  $\{C, B, A\}$  y  $\{B, C, A\}$ . Además, no importa el número de veces que aparezca un elemento en el conjunto:  $\{A, B, A\}$ ,  $\{A, B\}$  y  $\{A, A, A, B, B\}$

Son diferentes especificaciones de un mismo conjunto. Un conjunto finito puede ser caracterizado implícita o explícitamente, como lo demuestran los conjuntos:

$$\{1, 2, 3, 4, 5\} \text{ y } \{x \mid x \in \mathbb{I} \wedge 1 \leq x \leq 5\}$$

Que son el mismo conjunto.

### 2.2.1.1.3 OPERACIONES CON CONJUNTOS.

Existen operaciones que nos permiten crear nuevos conjuntos a partir de otros conocidos. Definimos la unión  $A \cup B$  y la intersección  $A \cap B$  de dos conjuntos  $A$  y  $B$  como sigue:

$$A \cup B = \{x: x \in A \text{ o } x \in B \text{ o ambas}\}$$

$$A \cap B = \{x: x \in A \text{ y } x \in B\}$$

Añadimos “o ambas” para dar énfasis y claridad a la definición de  $A \cup B$ . En español, la palabra o tiene dos significados. A veces es el “o” inclusivo que significa lo uno, lo otro o ambos. Dos conjuntos  $A$  y  $B$  son disjuntos si no tienen elementos comunes, es decir,

$$\text{Si } A \cap B = \emptyset.$$

Para dos conjuntos  $A$  y  $B$ , el complemento relativo  $A \setminus B$  es el conjunto de elementos que están en  $A$  y no están en  $B$ .

$$A \setminus B = \{x: x \in A \text{ y } x \notin B\} = \{x \in A: x \notin B\}$$

Es el conjunto que se obtiene al quitar de A los elementos que están en B.

Existen operaciones que nos permiten crear nuevos conjuntos a partir de otros conocidos.

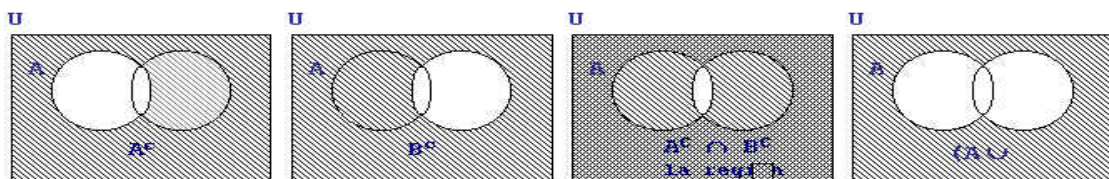
La diferencia simétrica  $A \oplus B$  de los conjuntos A y B es el conjunto

$$A \oplus B = (A \cup B) \setminus (A \cap B) = (A \setminus B) \cup (B \setminus A)$$

A veces, es conveniente ilustrar las relaciones entre conjuntos con dibujos llamados diagramas de Venn, en donde los conjuntos corresponden a subconjuntos del plano, por ejemplo:

Sea  $A = \{n \in \mathbb{N} : n = 7\}$ ,  $B = \{n \in \mathbb{N} : n \text{ es par y } n = 16\}$  y  $E = \{n \in \mathbb{N} : n \text{ es par}\}$ . Entonces tenemos

Figura 6 Diagramas de Venn.



$$A \cup B = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 16\},$$

$$A \cap B = \{0, 2, 4, 6\},$$

$$A \setminus B = \{1, 3, 5, 7\},$$

$$B \setminus A = \{8, 10, 12, 14, 16\},$$

$$A \oplus B = \{1, 3, 5, 7, 8, 10, 12, 14, 16\}.$$

### **2.2.1.2 LOGICA BORROSA.**

Es una lógica basada en la teoría de conjuntos que posibilita imitar el comportamiento de la lógica humana. El término "borroso" procede de la palabra inglesa "fuzzy" que significa "confuso, difuso, indefinido o desenfocado".

La lógica borrosa es una rama de la inteligencia artificial que se funda en el concepto "Todo es cuestión de cuanto se cumple", lo cual permite manejar información vaga o de difícil especificación. Es entonces posible con la lógica borrosa gobernar un sistema por medio de reglas de 'sentido común', las cuales se refieren a cantidades indefinidas.

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

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

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

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

La Lógica Borrosa ha sido probada para ser particularmente útil en sistemas expertos y otras aplicaciones de inteligencia artificial. Es también utilizada en algunos correctores de voz para sugerir una lista de probables palabras a reemplazar un término erróneo. La Lógica Borrosa, que hoy en día se encuentra en constante evolución, nació en los años 60 como la lógica del razonamiento aproximado y en ese sentido podía considerarse una extensión de la Lógica Multivaluada.

Psicológicamente la lógica borrosa puede definirse como la resolución de problemas que implica cierto grado de inferencia e intuición para lograr la conclusión propia; vista como una distinción crucial entre la inteligencia humana y la mecánica. Desde el punto de vista de la inteligencia artificial, es un método de razonamiento de maquina similar al pensamiento humano que puede procesar información incompleta o incierta, característico de muchos sistemas expertos. Un modelo de lógica borrosa consiste básicamente en que cada variable se asocia en cierto grado a alguna categoría particular y en cierto grado a otras. Por ejemplo, si el número 1 es considerado "pequeño" y el número 10 "grande", ¿Entonces como se considera el número 6? Podría ser considerado como "mediano", o en realidad también como "grande" si lo miramos con relación al número 1. Si nos fijamos bien, el número 6 tiene algo de grande, pero también algo de mediano. Es decir, su definición es "borrosa".

Los sistemas de control borrosos tienen una gran variedad de aplicaciones que van desde la estimación de parámetros, toma de decisiones, sistemas mecánicos de control tales como el aire acondicionado o lavadoras automáticas, hasta el control de automóviles o casas "inteligentes". Las nociones como "más bien caliente" o "poco frío" pueden formularse matemáticamente y ser procesados por computadoras. De esta manera, se ha realizado un intento de aplicar una forma más humana de pensar en la programación de computadoras [2].

### **2.2.1.2.1 CONJUNTOS BORROSOS.**

#### **Historia**

Los conjuntos borrosos fueron introducidos por primera vez en 1965. En cierto nivel, la disciplina de la lógica borrosa puede ser vista como un lenguaje que permite trasladar sentencias sofisticadas en lenguaje natural, a un lenguaje matemático formal. Mientras la motivación original fue ayudar a manejar aspectos imprecisos del mundo real, la práctica temprana de la lógica borrosa permitió el desarrollo de aplicaciones prácticas. Aparecieron numerosas publicaciones que presentaban los fundamentos básicos con aplicaciones potenciales. Esta fase marcó una fuerte necesidad de distinguir la lógica borrosa de la teoría de probabilidad. Tal como la entendemos ahora, la teoría de conjuntos borrosos y la teoría de probabilidad tienen diferentes tipos de incertidumbre.

En 1994, la teoría de la lógica borrosa se encontraba en la cumbre, pero esta idea no es nueva. Para muchos, estuvo bajo el nombre de lógica borrosa durante 25 años, pero sus orígenes se remontan hasta 2,500 años. Aún Aristóteles consideraba que existían ciertos grados de veracidad y falsedad. Platón había considerado ya grados de pertenencia.

En el siglo XVIII, el filósofo y obispo anglicano Irlandés George Berkeley y David Hume describieron que el núcleo de un concepto atrae conceptos similares. Hume en particular, creía en la lógica del sentido común, el razonamiento basado en el conocimiento que la gente adquiere en forma ordinaria mediante vivencias en el mundo. En Alemania, Emmanuel Kant, consideraba que sólo los matemáticos podían proveer definiciones claras y muchos principios contradictorios no tenían solución. Particularmente la escuela americana de la filosofía llamada pragmatismo, fundada a principios de siglo por Charles Sanders Peirce, cuyas ideas se fundamentaron en estos conceptos, fue la primera en considerar "vaguedades", más que falso o verdadero, como forma de acercamiento al mundo y a la forma en que la gente funciona.

La idea de que la lógica produce contradicciones fue popularizada por el filósofo y matemático británico Bertrand Russell a principios del siglo XX. Estudió las vaguedades

del lenguaje, concluyendo con precisión que la vaguedad es un grado. El filósofo austriaco Ludwig Wittgenstein estudió las formas en las que una palabra puede ser empleada para muchas cosas que tienen algo en común. La primera lógica de vaguedades fue desarrollada en 1920 por el filósofo Jan Lukasiewicz. Visualizó los conjuntos con un posible grado de pertenencia con valores de 0 y 1, después los extendió a un número infinito de valores entre 0 y 1. En los años sesentas, Lotfi Zadeh inventó la lógica borrosa, que combina los conceptos de la lógica y de los conjuntos de Lukasiewicz mediante la definición de grados de pertenencia [3].

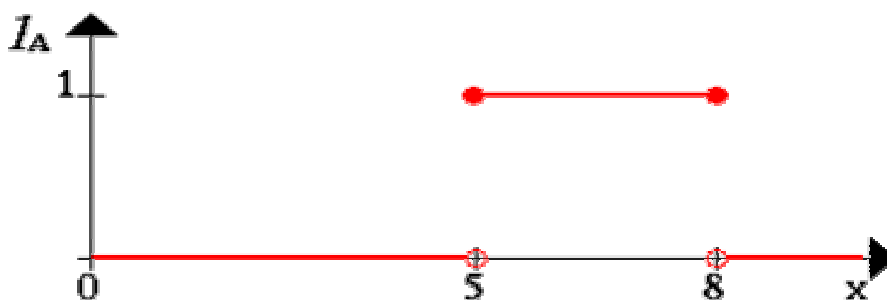
### Notación.

En primer lugar, se considera un conjunto  $X$  con todos los números reales entre 0 y 10 llamado el universo de discurso. Ahora, se define un subconjunto  $A$  de  $X$  con todos los números reales en el rango entre 5 y 8.

$$A = [5,8]$$

Ahora se muestra el conjunto  $A$  por su función característica, es decir esta función asigna un número 1 o 0 al elemento en  $X$ , dependiendo de si el elemento está en el subconjunto  $A$  o no. Esto conlleva a la figura siguiente:

Figura 7 Función Característica de un conjunto clásico.



Se podrán interpretar los elementos que han asignado el número 1 como los elementos que están en el conjunto A y los elementos que han asignado el número 0 como los elementos que no están en el conjunto A.

Este concepto es suficiente para muchas áreas de aplicación. Pero se podrán encontrar fácilmente situaciones donde carece de flexibilidad, como por ejemplo:

Para describir el conjunto de gente joven, formalmente se denota como:

$$B = \{\text{conjunto de gente joven}\}$$

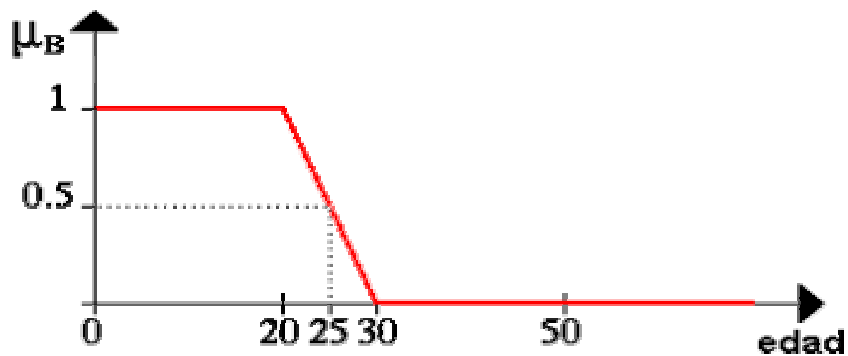
Como en general la edad comienza en 0, el rango más inferior de este conjunto está claro. El rango superior, por otra parte, es más bien complicado de definir. Como un primer intento se establece el rango superior en 20 años. Por lo tanto, B se define como un intervalo denominado:

$$B = [0,20]$$

Ahora la pregunta es: ¿por qué alguien es en su 20 cumpleaños joven y al día siguiente no? Obviamente, este es un problema estructural, porque si se mueve el límite superior del rango desde 20 a un punto arbitrario se podrá plantear la misma pregunta.

Una manera más natural de construir el conjunto B estaría en suavizar la separación estricta entre el joven y el no joven. Esto se hará para permitir no solamente la "crispada" decisión "el / ella SI está en el conjunto de gente joven" o "el / ella NO está en el conjunto de gente joven", sino también frases más flexibles como "el / ella SI pertenece un poquito más al conjunto de gente joven" o "el / ella NO pertenece aproximadamente al conjunto de gente joven".

Figura 8 Conjunto Gente joven.



De lo anterior se llega a un concepto a priori: “Se pueden usar conjuntos borrosos para hacer computadoras más sabias”. Resta codificar la idea más formalmente. En el primer ejemplo se codifican todos los elementos del Universo de Discurso con 0 o 1. Una manera de generalizar este concepto está en permitir más valores entre 0 y 1. De hecho, permitiendo infinitas alternativas entre 0 y 1, denominando el intervalo de unidad  $Y_0 = [0,1]$ .

La interpretación de los números ahora asignados a todos los elementos del Universo de Discurso es algo más difícil. Por supuesto, el número 1 asignado a un elemento significa que el elemento está en el conjunto B y 0 significa que el elemento no está definitivamente en el conjunto B. El resto de valores significan una pertenencia gradual al conjunto B.

De esta forma, 25 años de edad todavía sería joven al grado de 50 por ciento

#### 2.2.1.2.2 OPERACIONES CON CONJUNTOS BORROSOS.

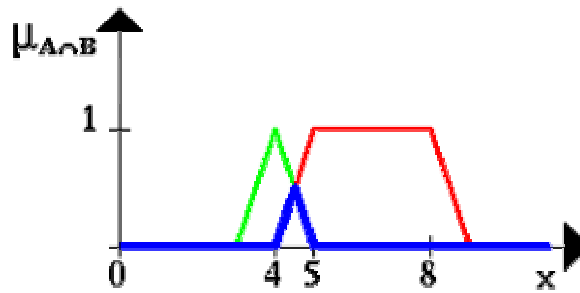
Para operar con conjuntos borrosos se extienden las operaciones con conjuntos clásicos:

Igualdad  $A = B \Leftrightarrow \forall x \in X : \mu_A(x) = \mu_B(x)$

Inclusión  $A \subseteq B \Leftrightarrow \forall x \in X : \mu_A(x) \leq \mu_B(x)$

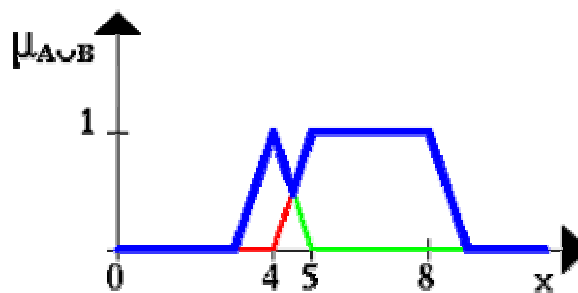
Intersección  $\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$ .

Figura 9 Intersección de conjuntos Borrosos.



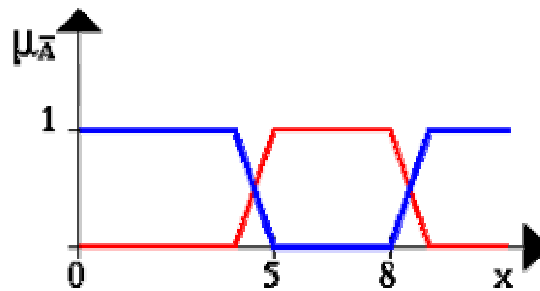
Unión  $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$

Figura 10 Unión de Conjuntos Borrosos.



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

Figura 11 Complemento de un Conjunto Borroso.



Existen generalizaciones de estas operaciones, ya que tanto las funciones de pertenencia de los conjuntos borrosos como sus operaciones dependen del contexto: T-normas (intersección) y T-conormas (unión)

### **T-Normas**

Generaliza el concepto de intersección.

$$T : [0,1] \times [0,1] \rightarrow [0,1]$$

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

Conmutativa:	$T(a, b) = T(b, a)$
Asociativa:	$T(a, T(b, c)) = T(T(a, b), c)$
Monotonía:	$T(a, b) \geq T(c, d)$ , si $a \geq c$ y $b \geq d$
Condiciones de frontera:	$T(a, 1) = a$

### Ejemplos de T-normas

Intersección estándar:  $T(a, b) = \min. (a, b)$

Producto algebraico:  $T(a, b) = a \cdot b$

Diferencia acotada:  $T(a, b) = \max. (0, a + b - 1)$

Intersección drástica:  $T(a, b) = \begin{cases} a, & \text{si } b = 1 \\ b, & \text{si } a = 1 \\ 0, & \text{e .o .c.} \end{cases}$

### T-Conormas.

Generaliza el concepto de unión

$$S : [0,1] \times [0,1] \rightarrow [0,1]$$

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

Conmutativa:  $S(a, b) = S(b, a)$

Asociativa:  $S(a, S(b, c)) = S(S(a, b), c)$

Monotonía:  $S(a, b) \geq S(c, d)$ , si  $a \geq c$  y  $b \geq d$

Condiciones de frontera:  $S(a, 0) = a$

### Ejemplos de t-conormas.

Unión estándar:  $S(a, b) = \max. (a, b)$

Suma algebraica:  $S(a, b) = a + b - a \cdot b$

Suma acotada:  $S(a, b) = \min. (1, a + b)$

Unión drástica:  $S(a, b) = \begin{cases} a, & \text{si } b = 0 \\ b, & \text{si } a = 0 \\ 1, & \text{e .o .c.} \end{cases}$

### 2.2.1.2.3 PROPIEDADES BÁSICAS DE LOS CONJUNTOS BORROSOS.

- Conmutativa:  $A \cup B = B \cup A$   $A \cap B = B \cap A$ .
- Asociativa:  $A \cup (B \cap C) = (A \cup B) \cap C = A \cup B \cap C$
- Idempotencia:  $A \cup A = A$ ;  $A \cap A = A$ ;
- Distributiva:  $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ ;
- Involución:  $\neg(\neg A) = A$ .

### 2.2.1.2.4 CARACTERÍSTICAS DE UN CONJUNTO BORROSO.

Soporte: Conjunto de elementos cuyo grado de pertenencia es distinto de cero.

$$Sop(A) = \{x \in X \mid \mu_A(x) > 0\}$$

Núcleo: Conjunto de elementos cuyo grado de pertenencia es igual a 1.

$$Nucleo(A) = \{x \in X \mid \mu_A(x) = 1\}$$

Altura: Grado de pertenencia más grande de los elementos del conjunto.

$$Altura(A) = \sup_{x \in X} \mu_A(x)$$

Conjunto Borroso Normal: Conjunto borroso cuya altura es igual a 1

$$Altura(A) = 1$$

### 2.2.1.2.5 RELACIONES ENTRE CONJUNTOS BORROSOS.

- Igualdad (*equality*): Dos conjuntos borrosos, definidos en el mismo Universo, son iguales si tienen la misma función de pertenencia:

$$A = B \Leftrightarrow A(x) = B(x), \forall x \in X$$

- Inclusión (*inclusion*): Un conjunto borroso está incluido en otro si su función de pertenencia toma valores más pequeños:

$$A \subseteq B \Leftrightarrow A(x) \leq B(x), \forall x \in X$$

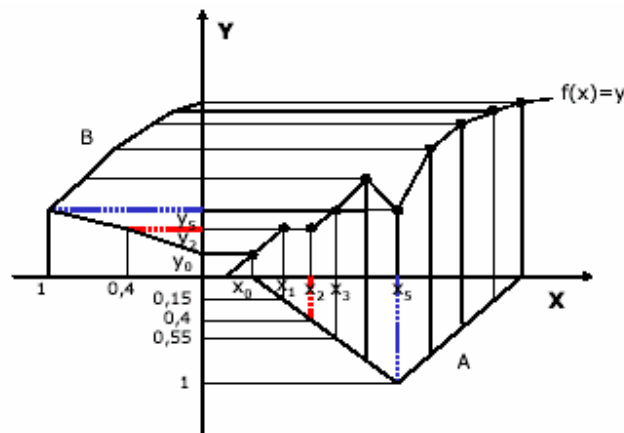
- Inclusión Borrosa: Si el Universo es finito, podemos relajar la condición anterior para medir el grado en el que un conjunto borroso está incluido en otro.

$$S(A, B) = \frac{1}{\text{Card}(A)} \left\{ \text{card}(A) - \sum_{x \in X} \text{Max} \{0, A(x) - B(x)\} \right\}$$

### 2.2.1.2.6 PRINCIPIO DE EXTENSIÓN.

El principio de extensión permite extender funciones crisp para operar con conjuntos borrosos, es decir, dada una relación de variables y unas entradas representadas con conjuntos borrosos, devuelve una salida expresada mediante un conjunto borroso.

Figura 12 Principio de Extensión.



### 2.2.1.2.7 RAZONAMIENTO APROXIMADO.

Como principales características del razonamiento aproximado tenemos:

- Comprender el concepto de variable lingüística y su uso para manejar conceptos expresados lingüísticamente.
- Conocer el concepto de regla borrosa, distintas interpretaciones de la misma, junto con sus propiedades y fórmulas de cálculo.
- Repasar las reglas de inferencia básicas y comprender su generalización a proposiciones borrosas.
- Entender la regla Composicional de inferencia y su aplicación.

### 2.2.1.2.8 VARIABLES LINGÜÍSTICAS.

Son variables cuyos valores se representan mediante términos lingüísticos. El significado de estos términos lingüísticos se determina mediante conjuntos borrosos.

- Proporcionan una transición gradual de estados.
- Tienen capacidad para expresar y trabajar con observaciones y medidas de incertidumbre.
- Al capturar medidas de incertidumbre, son más ajustadas a la realidad que las variables clásicas.

#### **Definición:[4]**

Una variable borrosa se caracteriza mediante:

(V, T, X, G, M)

[V] es el nombre de la variable.

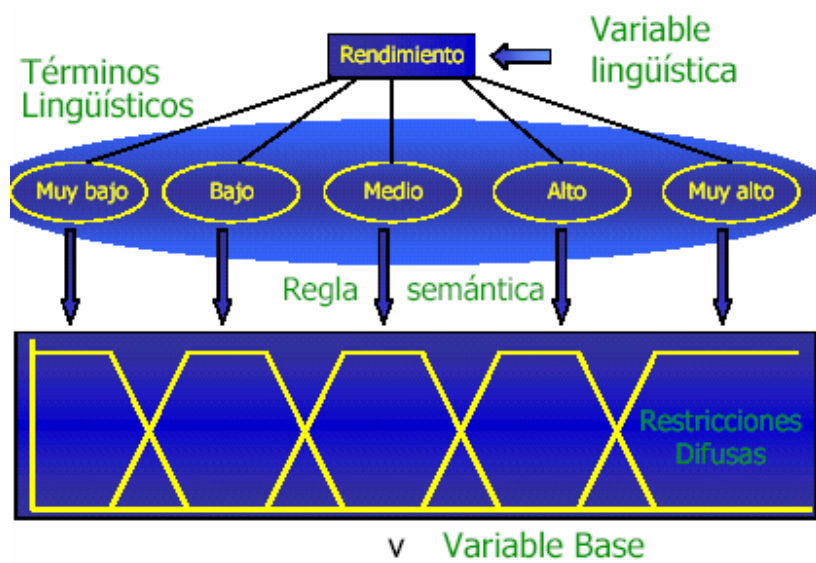
[T] es el conjunto de términos lingüísticos de v.

[X] es el universo de discurso para v.

[G] es una regla sintáctica para generar términos lingüísticos, y

[M] es una regla semántica que asigna a cada término lingüístico t su significado m (t) que es un conjunto borroso en X.

Figura 13 Variable Lingüística.



### 2.2.1.2.9 MODIFICADORES LINGÜÍSTICOS.

Un modificador lingüístico es un operador unario que se aplica a un conjunto borroso

$$h: [0,1] \rightarrow [0,1]$$

- "muy":  

$$h(a) = a^2, \quad a \in [0,1]$$
- "Mas o menos":  

$$h(a) = \sqrt{a}, \quad a \in [0,1]$$
- otros: "extremadamente", "bastante", etc.

### **Consideraciones Generales.**

- Con el uso de modificadores lingüísticos se debe evitar la ambigüedad.
- Los modificadores lingüísticos y los conectivos permiten obtener un amplio conjunto de términos compuestos que amplían la potencia descriptiva de la variable lingüística.
- Si el número de términos de una variable aumenta indefinidamente se llegará a la indistinguibilidad semántica de algunos de ellos.
- Granularidad (Lotfi Zadeh): Nivel de distinción entre los distintos niveles de incertidumbre contenida en las variables lingüísticas de forma que se pueda representar Correctamente la distinción que desea el usuario.

#### **2.2.1.2.10 PROPOSICIONES BORROSAS.**

El conocimiento humano se expresa en términos de reglas borrosas

SI \_ ENTONCES:

SI <proposición borrosa> ENTONCES <proposición borrosa>

Tipos de proposiciones borrosas:

- Atómicas:  $x$  es  $A$ , donde  $x$  es una variable lingüística y  $A$  es un valor lingüístico de  $x$ .
- Compuestas: Composición de proposiciones borrosas atómicas con las conectivas “y”, “o” y “no”, representando intersección, unión y complemento borroso, respectivamente.

Ejemplo de proposición borrosa atómica

Error es Negativo-Grande

- La interpretación o significado de una proposición borrosa atómica se define mediante la función de pertenencia del conjunto borroso Negativo-Grande.

- El grado de pertenencia de un error concreto al conjunto borroso Negativo-Grande determinará el grado con que se verifica la proposición borrosa.

Ejemplos de proposiciones borrosas compuestas:

1. X es A o X no es B.
2. X es A y X es B.
3. X no es A y X no es B.
4. (X es A y X no es B) o X es C.
5. X es A y Y es D.

- En una proposición borrosa compuesta pueden estar implicadas variables distintas.
- Las proposiciones borrosas compuestas se pueden considerar relaciones borrosas.

¿Cómo se determina la interpretación de estas relaciones borrosas y su respectiva función de pertenencia?

- Para las conectivas “y” se deben utilizar intersecciones borrosas.
- 

X es A y Y es B.

$$\mu_{A \cap B}(x, y) = T[\mu_A(x), \mu_B(y)]$$

- Para las conectivas “o” se deben utilizar uniones borrosas.

X es A o Y es B.

$$\mu_{A \cup B}(x, y) = S[\mu_A(x), \mu_B(y)] .$$

- Para las conectivas “no” se deben utilizar complementos borrosos.

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

### 2.2.1.2.11 REGLAS BORROSAS.

Una regla borrosa es una proposición borrosa condicional

Ejemplo:

- SI error es Negativo-Grande ENTONCES Y es Negativo-pequeño
- Su significado se representa mediante una relación borrosa entre el error y la variable de salida Y
- La función de pertenencia de esta relación borrosa se determina mediante un operador de implicación borroso.

#### Operadores De Implicación.

Algunos operadores de implicación borrosos:

Larsen:  $\mu_R(x, y) = \mu_A(x) \bullet \mu_B(y)$

Mamdani:  $\mu_R(x, y) = \min\{\mu_A(x), \mu_B(y)\}$

Dienes-Rescher:  $\mu_R(x, y) = \max\{1 - \mu_A(x), \mu_B(y)\}$

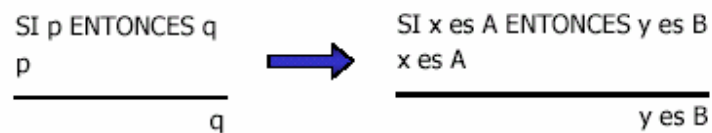
Lukasiewicz:  $\mu_R(x, y) = \min\{1, 1 - \mu_A(x) + \mu_B(y)\}$

Zadeh:  $\mu_R(x, y) = \max\{\min\{\mu_A(x), \mu_B(y)\}, 1 - \mu_A(x)\}$

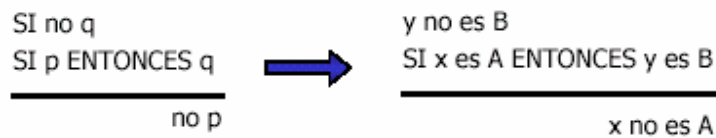
### 2.2.1.2.12 RAZONAMIENTO EN LÓGICA BORROSA.

Las reglas de inferencia permiten obtener valores de verdad a partir de valores de verdad probados

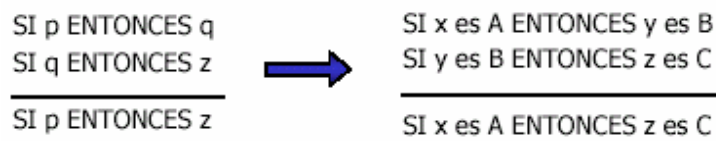
**Modus ponens:**



**Modus tollens:**



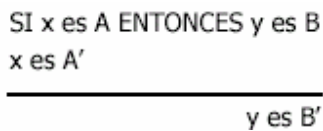
**Silogismo hipotético:**



**Modus Ponens Generalizado.**

Razonamiento aproximado: Obtención de conclusiones borrosas a partir de proposiciones borrosas utilizando la teoría de conjuntos borrosos como principal herramienta.[11 ]

**Modus ponens generalizado:**



Criterio	x es A'	y es B'
<b>1</b>	x es A	y es B
<b>2</b>	x es muy A	y es muy B
<b>3</b>	x es muy A	y es B
<b>4</b>	x es más o menos A	y es más o menos B
<b>5</b>	x es más o menos A	y es B
<b>6</b>	x no es A	y es desconocido
<b>7</b>	x no es A	y no es B

### Regla Composicional De Inferencia.

¿Cómo obtener el conjunto borroso B'?

Regla Composicional de Inferencia: Permite traducir el modus ponens de la lógica clásica a la lógica borrosa.

Si x es pequeño ENTONCES y es grande  
x es muy pequeño

El significado de la primera proposición se podría definir como una relación borrosa R.

$$B' = A' \circ R \Rightarrow \mu_{B'}(y) = \text{Max}[T(\mu_{A'}(x), \mu_R(x, y))]$$

### 3. SISTEMAS INTELIGENTES DE CONTROL.

Desarrollo de métodos de control para emular características importantes de la inteligencia humana como:

- Adaptación,
- Aprendizaje,
- Tratamiento de grandes cantidades de datos, y
- Tratamiento de incertidumbre.
- 

Lo que es control inteligente hoy será simplemente control mañana; esto quiere decir que son áreas de límites cambiantes.

El informe Task Force on Intelligent Control define el control inteligente a través de varias propiedades propias de los sistemas inteligentes:

- Adaptación y aprendizaje: Capacidad para adaptarse a condiciones cambiantes.
- Autonomía e inteligencia: Habilidad para actuar adecuadamente en un entorno con incertidumbre.
- Estructuras y jerarquías: Arquitectura funcional apropiada para afrontar problemas complejos.

El objetivo es el de diseñar sistemas de control automático (controladores) que sean Robustos, Adaptables, con capacidad de aprendizaje de la experiencia y de la intervención humana y autónomos.

Concretamente se estudiarán los controladores borrosos y los modelos de aprendizaje neuronal y evolutivo.

### **Definiciones más utilizadas:**

- Sistema: Combinación de componentes que actúan juntos y realizan un objetivo determinado.
- Variable controlada: Condición que se mide y controla
- Variable manipulada: Condición que el controlador modifica para afectar el valor de la variable controlada.
- Controlar: Medir la variable controlada y alterar la variable manipulada para corregir o limitar la variable controlada.
- Planta: Parte del sistema que se controla.
- Proceso: “Operación continua, marcada por cambios graduales que se suceden uno a otro de una forma relativamente fija y que conduce a un resultado determinado”.
- Perturbación: Señal que tiende a afectar negativamente el valor de la salida de un sistema.

### **3.1 CONTROL INTELIGENTE.**

Entre los sistemas de control inteligente que existen hoy en día encontramos tres clases de control:

- Sistemas Neuro-Borrosos.
- Sistemas Genético-Borrosos.
- Sistemas basados en reglas borrosas.

#### **3.1.1 SISTEMAS DE CONTROL NEURO-BORROSOS.**

Las redes neuronales artificiales y la lógica borrosa son dos de los avances científicos recientes más importantes del conocimiento. Mediante estas herramientas se ha intentado

simular dos de las características más importantes con las cuales cuenta el cerebro humano: la capacidad de aprendizaje y el poder procesar información incompleta o que no es precisa. Estas técnicas se han podido utilizar para la solución de problemas tanto científicos como de la vida diaria. Con una buena combinación de estas dos técnicas es posible un desarrollo tecnológico más amplio en diversos campos, desde la medicina hasta el mejoramiento de electrodomésticos.

### **3.1.1.1 REDES NEURONALES.**

#### **Introducción.**

Mediante esta técnica se intenta imitar el proceso de aprendizaje del cerebro humano. El cerebro está formado por miles de millones de neuronas conectadas entre sí. Utiliza una información que es percibida, transmitida hasta las neuronas y allí es procesada por ellas para dar una respuesta a cada uno de los diferentes estímulos. Cada neurona tiene tres partes: un cuerpo celular, una estructura de entrada (Dendrita), y una de salida (Axón). La mayoría de los terminales de los axones se conectan con las dendritas de otras neuronas (Sinapsis). El comportamiento de una neurona es el siguiente: recibe una señal de entrada con una fuerza determinada, dependiendo de ellas la neurona emite una señal de respuesta, las sinapsis pueden variar en fuerza, algunas pueden enviar una señal fuerte y otras débiles. A una neurona pueden llegar miles de señales de entrada, cada una con una fuerza o peso diferente. Matemáticamente el comportamiento de una neurona puede representarse por una lista de sus señales de entrada que son multiplicadas por sus pesos respectivos y posteriormente sumados, el resultado es llamado nivel de activación de la neurona del cual depende la señal de salida que es enviada a cada una de las neuronas a las que está conectada a ella.

Una red neuronal artificial (RNA) es un sistema compuesto de muchos sistemas procesadores simples conectados en paralelo, cuya función es determinada por la

estructura de la red, la fuerza en las conexiones y el procesamiento realizado por los elementos en los nodos (Jang 1997). Las RNA, igual que las personas, aprenden de la experiencia.

### **Referencia Histórica.**

McCulloch y Pitts (1943)

- Propusieron un modelo de una neurona.
- Modelo de McCulloch-Pitts, Hebb.
- Se presenta el funcionamiento de la regla de aprendizaje fisiológico mediante modificación sináptica.

Rosenblatt (1958)

- Se introduce un nuevo enfoque al problema de reconocimiento de patrones; perceptron.

Widrow y Hoff (1960)

- Se introduce el algoritmo de mínimos cuadrados (LMS).
- Se formula el Adaline (elemento lineal adaptativo).

Hopfield (1982)

- Se usa la idea de la función de energía para formular el cálculo realizado por redes recurrentes.
- Redes estables dinámicas.

Barto, Sutton y Anderson (1983)

- Se propone el Aprendizaje por Refuerzo.

Rumelhart, Hinton, Williams (1986)

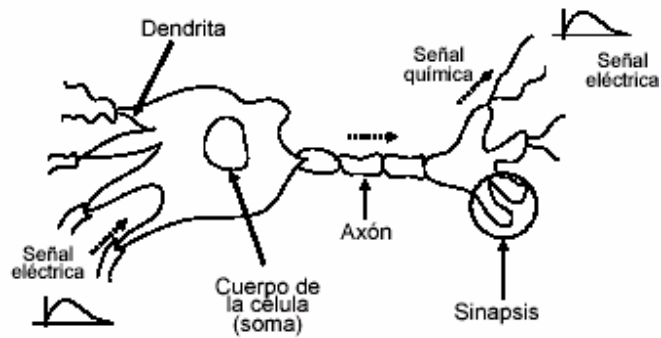
- Se propone el algoritmo de retro-propagación ( backpropagation)

Broomhead y Lowe (1988).

- Se propone la Red de Funciones de Base Radial (radial basis function, RBF).

### Neurona Biológica.

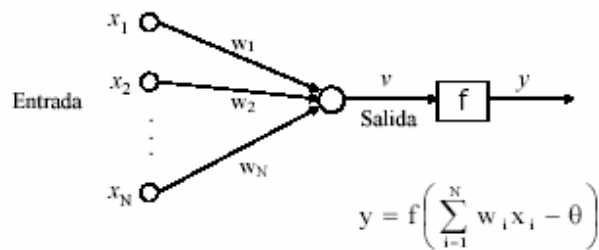
Figura 14 Neurona Biológica.



El cerebro humano contiene aproximadamente  $1.5 \times 10^{10}$  neuronas y cada neurona recibe señales de 104 sinapsis. Esta es la metáfora que se quiere implementar en los modelos de redes neuronales artificiales.

### Modelo de una Neurona Artificial.

Figura 15 Modelo de una Neurona artificial.



Los siguientes son los componentes de una neurona artificial:

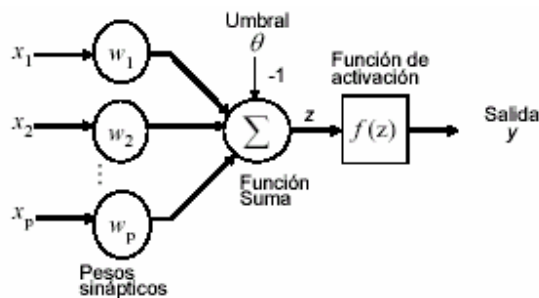
- Un conjunto de enlaces de entrada desde otros nodos
- Un conjunto de pesos que ponderan cada entrada
- Una salida.
- Una función de activación, generalmente no lineal.

### Perceptron.

Consiste en una única neurona con pesos sinápticos ajustables.

Se usa la función de activación Limitador Estricto.

Figura 16 Perceptron.



El perceptrón permite ser entrenado, es decir, a partir de un conjunto de pares entrada-salida conocidos, podemos modificar los pesos  $w_i$  de cada entrada para que la salida obtenida se aproxime más a la salida deseada.

### Algoritmo de Entrenamiento del Perceptrón.

El objetivo del perceptrón es clasificar un conjunto de estímulos externos  $X_1, X_2, \dots, X_p$

en dos clases: C1 y C2.

Variables y Parámetros

$\mathbf{x}(n)$ : vector de entrada

$\mathbf{w}(n)$ : vector de pesos

$\theta(n)$ : umbral

$y(n)$ : respuesta real (cuantificada)

$d(n)$ : salida esperada

$a$ : ratio de aprendizaje, constante positiva menor que 1

Actualización de Pesos

$$\mathbf{w}(n+1) = \mathbf{w}(n) + a [ d(n) - y(n) ] \mathbf{x}(n)$$

Donde  $d(n) =$  +1 si  $\mathbf{x}(n)$  pertenece a la clase C1

-1 si  $\mathbf{x}(n)$  pertenece a la clase C2

Repetir para cada  $\mathbf{x}(n)$  del conjunto de entrenamiento.

### Separabilidad lineal del perceptrón

A partir del modelo se encuentra que la salida combinada lineal (es decir, la entrada al limitador estricto) es:

$$Z = \sum_{i=1}^p w_i x_i - \theta$$

Se obtienen dos regiones de decisión separadas por un hiperplano definido como:

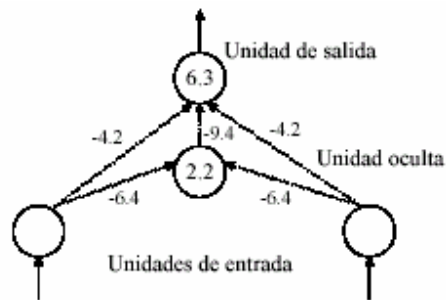
$$Z = \sum_{i=1}^p w_i x_i - \theta = 0$$

Esto implica que un perceptrón sólo puede resolver problemas linealmente separables.

Función XOR

- Ejemplo de la red XOR generada.

Figura 17 Función XOR generada.



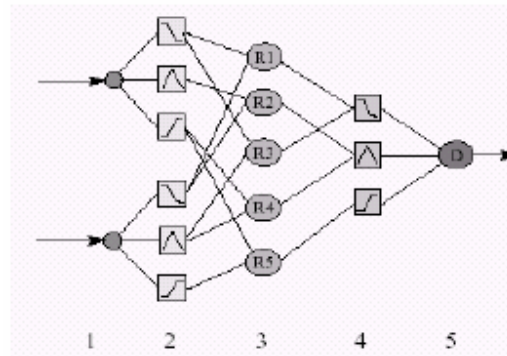
### 3.1.1.2 SISTEMAS NEURO - BORROSOS.

Son sistemas híbridos que combinan técnicas de redes neuronales y sistemas de inferencia borrosa. Asociando de este modo la capacidad de aprendizaje de las RNA con la tolerancia a fallos, interpretabilidad y robustez de los sistemas borrosos. Además permiten la integración de conocimiento (métodos previos, expertos, etc.) siendo posible extraer el conocimiento incluido en la RNA en formato de reglas borrosas (por eso son considerados modelos de “caja gris”) [6].

La arquitectura más común consta de 5 capas:

1. Entradas
2. Emborronado.
3. Reglas.
4. Consecuentes.
5. Desemborronado.

Figura 18 Sistema Neuro – Borroso.



### 3.1.1.2.1 LIMITACIONES DE LOS SISTEMAS NEURO – BORROSOS.

Entre las principales limitaciones de los sistemas Neuro -Borrosos tenemos:

- Número pequeño de entradas (curso de la dimensionalidad: crecimiento geométrico de la complejidad según el número de entradas).
- Dificultad para aprender la estructura de las reglas. Generalmente, sólo aprenden la forma de las funciones de pertenencia y los coeficientes del consecuente (en TSK).
- Dificultad para tratar funciones no diferenciables (por ejemplo, la t-norma del mínimo).
- Problemas de convergencia: caída en óptimos locales.
- Problemas de sobre aprendizaje: Error de aproximación (conjunto de entrenamiento) mucho menor que el de generalización (conjunto de validación).

Algunos ejemplos de Sistemas Neuro -Borrosos:

NEFCLASS (D. Nauck, 1994)

- Particionamiento de rejilla.
- Reglas de clasificación. FSOM (P. Vuorimaa, 1996)

- Particionamiento de grafos borrosos (SBRD aproximativo). NFH (F.J. de Soutza, 1997)
- Particionamiento jerárquico.

### Anfis.

Adaptive Network based Fuzzy Inference System (Jyh-Shing Roger Jang, 1993).

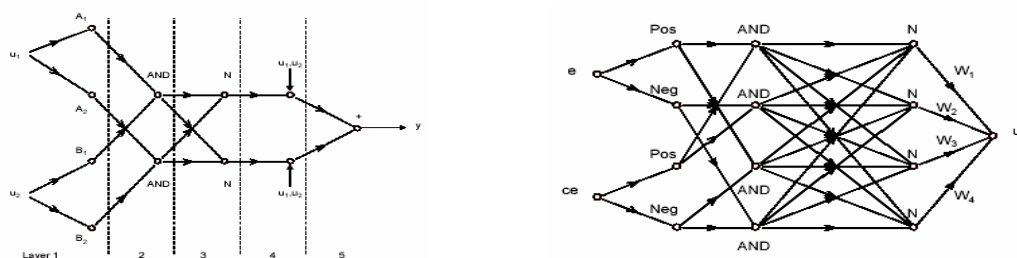
- Utiliza variables lingüísticas (particionamiento de rejilla).
- Únicamente ajusta las funciones de pertenencia.
- Apto sólo para reglas TSK de orden 0 (el consecuente es un número real) ó 1 (el consecuente es un polinomio de primer grado).

Entrenamiento en dos pasos:

- Fijar el consecuente y ajustar los parámetros del antecedente (funciones de pertenencia) mediante Gradiente Descendente.
- Fijar el antecedente y ajustar los parámetros del consecuente (coeficientes de los polinomios) mediante Optimización por Mínimos Cuadrados.

### Estructura de Anfis.

Figura 19 Estructura de Anfis.



### **3.1.2 SISTEMAS DE CONTROL GENÉTICO- BORROSOS.**

Hoy en día las aplicaciones de sistemas de control borroso apoyados con algoritmos genéticos están alcanzando niveles muy altos de aceptación debido a su gran potencialidad y bajo costo computacional. De ahí que se este presentando un gran fenómeno de personas interesadas en investigar mucho más acerca de las grandes ventajas y virtudes que estos sistemas pueden brindar en un futuro.

#### **3.1.2.1 INTRODUCCIÓN SOBRE ALGORITMOS GENÉTICOS.**

Los Algoritmos Genéticos (GA) fueron introducidos por John Holland en 1970, inspirándose en el proceso observado en la evolución natural de los seres vivos. Los biólogos han estudiado en profundidad los mecanismos de la evolución y aunque quedan parcelas por entender, muchos aspectos están bastante explicados. De manera muy general podemos decir que en la evolución de los seres vivos el problema al que cada individuo se enfrenta cada día es la supervivencia. Para ello cuenta con las habilidades innatas provistas en su material genético. A nivel de los genes, el problema es el de buscar aquellas adaptaciones beneficiosas en un medio hostil y cambiante. Debido en parte a la selección natural, cada especie gana una cierta cantidad de "conocimiento", el cual es incorporado a la información de sus cromosomas. Así pues, la evolución tiene lugar en los cromosomas, en donde está codificada la información del ser vivo. La información almacenada en el cromosoma varía de unas generaciones a otras. En el proceso de formación de un nuevo individuo, se combina la información cromosómica de los progenitores aunque la forma exacta en que se realiza es aún desconocida.

Aunque muchos aspectos están todavía por discernir, existen unos principios generales ampliamente aceptados por la comunidad científica. Algunos de estos son:

- La evolución opera en los cromosomas en lugar de en los individuos a los que representan.

- La selección natural es el proceso por el que los cromosomas con "buenas estructuras" se reproducen más a menudo que los demás.
- En el proceso de reproducción tiene lugar la evolución mediante la combinación de los cromosomas de los progenitores. Se denomina Recombinación a este proceso en el que se forma el cromosoma del descendiente. También son de tener en cuenta las mutaciones que pueden alterar dichos códigos.
- La evolución biológica no tiene memoria en el sentido de que en la formación de los cromosomas únicamente se considera la información del período anterior

Los algoritmos genéticos establecen una analogía entre el conjunto de soluciones de un problema y el conjunto de individuos de una población natural, codificando la información de cada solución en un string (vector binario) a modo de cromosoma. En palabras del propio Holland:

*"Se pueden encontrar soluciones aproximadas a problemas de gran complejidad computacional mediante un proceso de "evolución simulada",*

A tal efecto se introduce una función de evaluación de los cromosomas, que se llama calidad ("fitness") y que está basada en la función objetivo del problema. Igualmente se introduce un mecanismo de selección de manera que los cromosomas con mejor evaluación sean escogidos para "reproducirse" más a menudo que los que la tienen peor.

Los algoritmos desarrollados por Holland inicialmente eran sencillos pero dieron buenos resultados en problemas considerados difíciles [10].

## **Computación Evolutiva**

Está compuesta por modelos de evolución basados en poblaciones cuyos elementos representan soluciones a problemas. La simulación de este proceso en un ordenador

resulta ser una técnica de optimización probabilística, que con frecuencia mejora a otros métodos clásicos en problemas difíciles.

Existen cuatro paradigmas básicos:

**Algoritmos Genéticos:** que utilizan operadores genéticos sobre cromosomas.

**Estrategias de Evolución:** que enfatizan los cambios de comportamiento al nivel de los individuos.

**Programación Evolutiva:** que enfatizan los cambios de comportamiento al nivel de las especies.

**Programación Genética:** que evoluciona expresiones representadas como árboles.

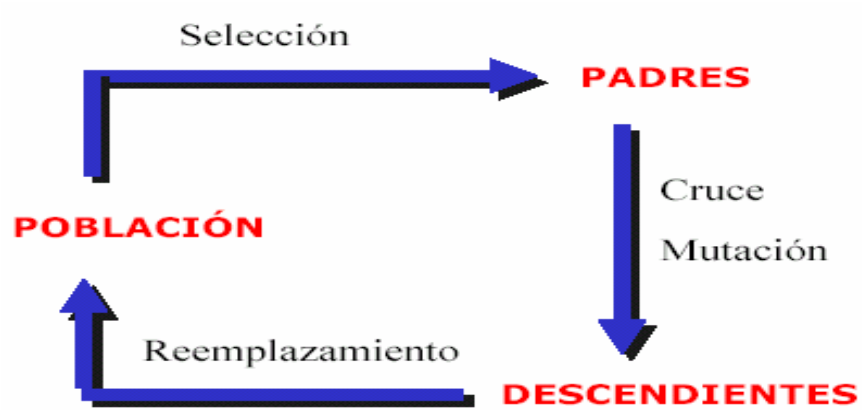
Dominios De Aplicación.

- Optimización combinatoria y en dominios reales.
- Modelado e identificación de sistemas.
- Planificación y control.
- Ingeniería.
- Vida artificial.
- Aprendizaje y minería de datos.
- Internet y Sistemas de Recuperación de Información.

### **El Ciclo De La Evolución.**

De la forma como se da el ciclo de la evolución en la naturaleza, así trabaja la programación evolutiva.

Figura 20 ciclo de la Evolución.



### 3.1.2.2 SISTEMAS GENÉTICO-BORROSOS.

#### Diseño.

En la actualidad se están desarrollando aplicaciones de sistemas que integran los algoritmos genéticos con sistemas basados en reglas borrosas, dando excelentes resultados en las aplicaciones ejecutadas a través de ellos.

Figura 21 Diseño de un Sistema Genético Borroso.



### **Objetivo del proceso de aprendizaje de un SBRB.**

-Encontrar una Base de Conocimiento tal que el SBRB que la incluya resuelva un problema dado.

¿Qué partes del SBRB se van a optimizar?

- Procesos de aprendizaje: Diseño de algunos componentes de la Base de Conocimiento o de la Base de Conocimiento al completo.
- Procesos de ajuste: Optimización de un SBRB existente.

Según las componentes que se optimicen:

Espacio de búsqueda más pequeño

- Proceso de aprendizaje más sencillo y rápido.
- Las soluciones pueden ser suboptimales.

Espacio de búsqueda más completo

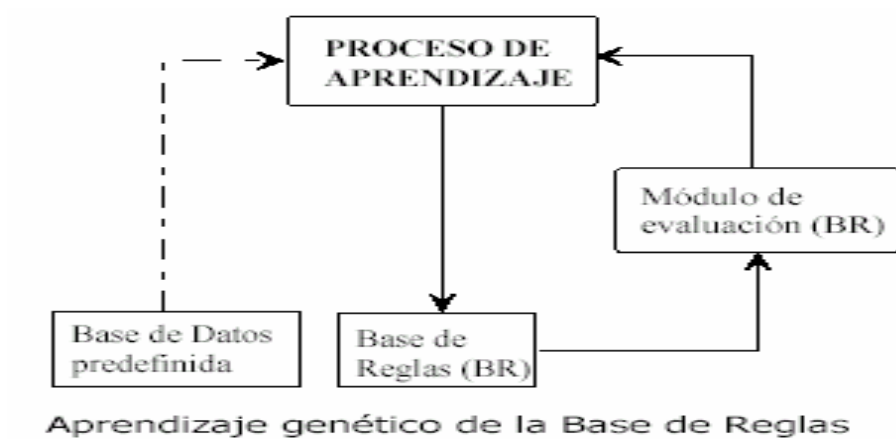
- Proceso de aprendizaje más complejo e ineficiente.
- Mayor granularidad en el aprendizaje, mejor consideración de la interdependencia, mayor probabilidad de encontrar soluciones óptimas.

### **Tipos de Sistemas basados en reglas Genéticos-borrosas:**

- Sistemas con ajuste genético de la Base de Datos
- Sistemas con aprendizaje genético de la Base de Reglas
- Sistemas con aprendizaje genético de la Base de Conocimiento
- Sistemas con aprendizaje genético del Mecanismo de Inferencia (Poco usuales)

### 3.1.2.2.1 SISTEMAS BASADOS EN REGLAS BORROSAS CON APRENDIZAJE GENÉTICO DE LA BASE DE REGLAS.

Figura 22 SBRBs Con Aprendizaje Genético de la Base de Reglas.



El aprendizaje genético de la Base de Reglas asume la existencia de un conjunto predefinido de funciones de pertenencia.

- Objetivo de la búsqueda: Un conjunto adecuado de reglas borrosas.
- Esquema de representación: Alternativas:
  - ✓ Un cromosoma representa una base de reglas al completo (Enfoque Pittsburgh) Apto para diseño off-line.
  - ✓ Un cromosoma representa una regla y la población al completo, la base de reglas (Enfoque Michigan). Apto para diseño on-line.

#### Ajuste de las Funciones de Escala.

- Trasladan el universo de discurso en los que se definen las variables de entrada y salida al dominio en el que se definen los conjuntos borrosos.

- Se adaptan para que el universo de discurso escalado se corresponda mejor con el rango de la variable.

#### Parámetros

- ✓ Factor de escala.
- ✓ Cota superior e inferior (función de escala lineal).
- ✓ Parámetros de contracción / expansión (función de escala no lineal).
- ✓ Forma de codificación: esquema real de longitud fija.

#### **Ajuste de las funciones de pertenencia.**

- Componente a optimizar: Funciones de pertenencia de los términos lingüísticos utilizados en las reglas borrosas.

Cada individuo = Base de Datos al completo.

Forma de codificación. Depende de:

- Tipo de función de pertenencia utilizada
- Tipo de SBRB:  
 Descriptivo, todas las funciones se adaptan de forma global a la Base de Reglas, o  
 Aproximativo, cada variable se asocia a un conjunto borroso distinto en cada regla.

### **3.1.3 SISTEMAS DE CONTROL BORROSO.**

Los diseñadores de sistemas de control han buscado incesantemente nuevos caminos para la solución del problema del control de procesos complejos.

Esto se da en los casos en que no existe un modelo preciso del proceso y la información apriorística acerca de este sea esencialmente cualitativa.

Por otro lado, el problema de control constituye uno típico de toma de decisiones y para este tipo de problemas la creación de la Lógica borrosa significó la aparición de una nueva y poderosa herramienta para enfrentarlo en situaciones de imprecisión esencial en modelos, información, objetivos, restricciones, y acciones de control.

En el campo específico de los sistemas de control, es frecuente el caso de procesos o plantas en las que controladores diseñados según otros patrones (clásicos, adaptables, etc.) no dan los resultados deseados o simplemente fallan del todo. Se trata de procesos de elevada complejidad y a pesar de ello, en la inmensa mayoría de los casos operadores humanos diestros y experimentados alcanzan resultados satisfactorios en su dirección (quizá “óptimos” en esas circunstancias).

Cuando se investiga el proceder de esos expertos, se concluye de inmediato que, tanto la información acerca del estado del proceso como sus acciones de control, las procesan y manejan de un modo básicamente cualitativo. Esto es típico para el proceder humano aún en casos de la dirección de procesos relativamente simples. Observando la actuación de un conductor de automóvil se aprecia que nunca accionará sobre el timón en términos de radianes o de grados, ni sobre el pedal del freno en términos de milímetros o newton. El, ante una esquina cerrada gira el timón mucho, al estar cerca de la señal de pare frena un poco, etc. Sin embargo, nadie pone en duda las excelentes virtudes del ser humano para conducir autos, las que frecuentemente llegan al virtuosismo (¿lo óptimo?).

Con sus principales antecedentes en la lógica multinivel de Lukasiewicz de los años 30, el apelativo “Fuzzy”, que se traduce como borroso es introducido por Zadeh en 1962 en un trabajo que vincula la teoría de los circuitos eléctricos con la de sistemas. Tres años más tarde (Zadeh 1965) da a luz la llamada teoría de conjuntos borrosos que echa los cimientos de la llamada síntesis lingüística, mostrando como pueden usarse planteos lógicos vagos para derivar inferencias (también vagas) a partir de datos vagos. Aunque este último trabajo sugiere aplicaciones a sistemas humanísticos en realidad la extensión al control de procesos industriales se hizo evidente. Al menos así lo fue para Ebrahim Mandami y su grupo del Queen Mary College de Londres, así como para su contemporáneo Richard M. Tong de la cercana universidad de Cambridge. Fue un numeroso colectivo de varias nacionalidades europeas como pioneros de la aplicación de la lógica borrosa al control, a partir de la segunda mitad de los 70.

La temática de la lógica borrosa ha sido abordada en la literatura por un número geoméricamente creciente de trabajos, desde 2 en 1965, 69 en 1970, alrededor de 600 en 1975, mas de 1500 en 1979 y, aunque no hay referencias exactas, puede estimarse en varios miles la cifra acumulada hasta ahora. Inclusive, desde 1978 existe la publicación *Fuzzy set and Systems*. Artículos como el de Maiers y Sherif (1985) y Lee (1990), por sólo citar algunos, incluyen referencias a centenares de trabajos, a pesar de ser seleccionados y con preferencia para las aplicaciones en control. Llegado este punto es bueno subrayar que las aplicaciones de Lógica Borrosa abarcan, además del control automático, infinidad de esferas de la ciencia y la técnica, desde la economía y la biología hasta las ciencias sociales.

Vale la pena comentar brevemente el hecho de que, aunque hoy la teoría de los conjuntos borrosos goza de la aceptación generalizada, el reconocimiento a su validez, rigor y aplicabilidad no ha estado exento a controversias. Algunos científicos de renombre han llegado a lanzarle fuertes invectivas imputándole ingenuidad filosófica, carencia de aplicabilidad e interés, excepto el puramente matemático. Sin embargo, las aplicaciones introducidas y el desarrollo teórico paralelo, de los últimos 20 años se han encargado de darle a esta teoría su justo valor, probablemente menos “milagrosa” que como nos la describiera Zadeh al principio, pero indudablemente muy útil y poderosa.

Los cuestionamientos se han relacionado frecuentemente, con su similitud con la teoría de probabilidades. No obstante, como un fruto de la lógica borrosa surgió la llamada teoría de las posibilidades, que guarda diferencias esenciales con aquella. Zadeh (1978) da un simple ejemplo que ilustra muy bien la diferencia entre la incertidumbre probabilística y la imprecisión posibilística. Para ello analiza el planteamiento “Hans comió X huevos al desayuno”. Allí el acota:

“Podemos asociar una distribución de posibilidades con X, interpretando a  $F(u)$  como el grado de facilidad con la que Hans puede comer u huevos. También podemos asociar una distribución de probabilidades con X, interpretando a  $P(u)$  como la probabilidad de

que Hans se comiera  $u$  huevos al desayuno, los valores de  $F(u)$  y  $P(u)$  pueden ser como se muestra:

U	1	2	3	4	5	6	7	8
F(u)	1	1	1	1	0.8	0.6	0.4	0.2
P(u)	0.1	0.8	0.1	0	0	0	0	0

Observamos que, mientras que la posibilidad de que Hans se comiera 3 huevos es 1, la probabilidad de que lo hiciera puede ser muy pequeña, 0.1. De ahí que un alto grado de posibilidad no implica un alto grado de probabilidad ni un grado bajo de probabilidad implica un grado bajo de posibilidad, por supuesto, si un evento es imposible queda circunscrito a ser improbable”.

Esta referencia sirve para subrayar la aplicación de la teoría de conjuntos borrosos al control, objetivo primordial de este estudio. Viene de interpretar planteos imprecisos de los operadores como la imposición de restricciones posibilísticas a la clase de eventos que los satisfacen. Tales restricciones se hacen corresponder con pertenencias graduadas, de modo que cada evento tiene su grado de pertenencia dentro del conjunto que delimita la medida en que es consistente con la restricción. Para un planteo como “Si la temperatura es alta...” existirá una gradación posibilística, entre 0 y 1. Entonces, una temperatura de 300 grados puede considerarse como perteneciente al conjunto de las “altas” (claro esta, en dependencia del proceso en cuestión), con un cierto grado de pertenencia, por ejemplo de 0.9, claro que los planteos no sólo implicarán la simple correspondencia de eventos con conjuntos (unión, intersección, etc.) que tuvieron sus primeras definiciones formales con los trabajos iniciales de Zadeh [4].

### **3.1.3.1 SISTEMAS BASADOS EN REGLAS BORROSAS.**

Sistemas que emplean reglas borrosas del tipo SI - ENTONCES para representar el conocimiento.

Aplicaciones:

Modelado de sistema:

- Obtención de modelos que representan realidades complejas.

Control:

- Plantas industriales complejas

Control en línea

- Sistemas de navegación con perturbaciones

Clasificación:

- Detección de patrones, diagnóstico médico.

Sistemas expertos:

- Ayuda a la decisión, recuperación de información, planificadores financieros.

Minería de datos y descubrimiento de información:

- Extracción del conocimiento intrínseco contenido en grandes bases de datos con reglas de asociación borrosas.

#### **Alternativas para Controles Digitales.**

PID (proporcional-integral-derivativo).

- Problemas en entornos de control cambiantes o sistemas no lineales.

MRAC (control adaptativo de modelo de referencia).

- Resuelve el problema anterior ajustando los parámetros del controlador comparando la salida con un modelo de referencia.
- Necesita un modelo matemático.

## Control Borroso

- Las entradas, salidas y respuesta de control se especifican con términos similares a los utilizados por un experto en control.
- No se requiere un modelo del sistema.
- Aprendizaje y ajuste automático fácil de realizar.

### 3.1.3.1.1 TIPOS DE SISTEMAS BASADOS EN REGLAS BORROSAS.

En función del tipo de regla borrosa que utilicen se pueden distinguir principalmente dos tipos.

Sistemas basados en reglas borrosas tipo Mamdani.

- SI X1 es Alto y X2 es Bajo ENTONCES Y es Alto.

Sistemas basados en reglas borrosas tipo TSK (Takagi, Sugeno y Kang).

- SI X1 es Alto y X2 es Bajo ENTONCES  $Y = f(X1, X2)$ .

A continuación se darán a conocer las principales características que identifican a cada uno de estos sistemas borrosos:

#### **Motor de inferencia.**

Tipo Mandami.

- Utiliza reglas borrosas para obtener la respuesta del sistema borroso ante una determinada entrada.
- Hay dos formas de realizar este proceso:
  - ✓ Inferencia basada en reglas individuales: Aplicar la entrada a la primera regla, a la segunda y así sucesivamente. Posteriormente las salidas de las reglas se unen para obtener una única salida.
  - ✓ Inferencia basada en la composición: Calcular la relación borrosa que representa el significado de toda la base de reglas para aplicar la entrada a esa relación borrosa global.

Esquema simplificado de un motor de inferencia basado en reglas individuales:

Disparo de reglas:

Una regla se dispara si el grado de emparejamiento (o aplicabilidad) del antecedente de la regla con la entrada es mayor que cero.

Cálculo del grado de emparejamiento.

- Antecedente con una variable.
- Antecedente con más de una variable.

Escalado o corte de la salida borrosa.

Agregación de las salidas (si es necesario).

Parámetros de diseño para el motor de inferencia:

Elección del tipo de motor de inferencia.

- Basado en reglas individuales.
- Basado en la composición de reglas.

Elección de la representación del significado de las reglas borrosas.

- Operadores de conjunción, disyunción, complemento, modificadores lingüísticos, según el caso.
- Operador de implicación.
- Operador de agregación de reglas.

Tipo TSK (Takagi, Sugeno, Kang).

Normalmente  $f(x_1, \dots, x_n) = a_0 + a_1 x_1 + \dots + a_n x_n$

- El antecedente se procesa igual que el de las reglas tipo Mamdani.
- Para una entrada específica, el resultado de disparar una regla es un valor nítido.
- Finalmente los valores nítidos obtenidos al dispararse distintas reglas se combinan para obtener una única salida (máximo, media aritmética ponderada, etc.).

**Ventajas y desventajas de los sistema basados en Reglas Borrosas.**

**Ventajas.**

Tipo Mandami.

- Facilidad para la derivación de reglas.
- Interpretabilidad de las reglas borrosas.
- Fueron propuestos antes y se han utilizado con más frecuencia.

Tipo TSK.

- Incrementan la precisión.
- Mayor eficiencia computacional.
- Facilidad para el análisis del sistema.
- No necesitan interfaz de Desemborronado.
- Garantizan la continuidad de la superficie de salida.

#### **Desventajas.**

Tipo Mandami.

- No garantizan la continuidad de la superficie de salida.
- Menor eficiencia computacional.

Tipo TSK.

- El consecuente es una fórmula matemática y no proporciona un marco natural para representar conocimiento humano.
- Limitan la representación de los principios de la lógica borrosa [1].

### **3.1.3.2 CONTROLADORES BORROSOS.**

Principales definiciones:

- ✓ Son Sistemas expertos en tiempo real que implementan una parte de la experiencia del operador humano o ingeniero de procesos que no se presta a ser fácilmente expresado con parámetros PID o ecuaciones diferenciales pero sí mediante reglas de situación o acción.
- ✓ Matriz multidimensional donde los subíndices son variables físicas de entrada, y el valor del elemento matricial representa la acción para obtener el estado deseado del sistema a controlar [8].

- ✓ Forma heurística y modular de definir sistemas de control basados en tablas no lineales.

### Objetivos de un controlador borroso.

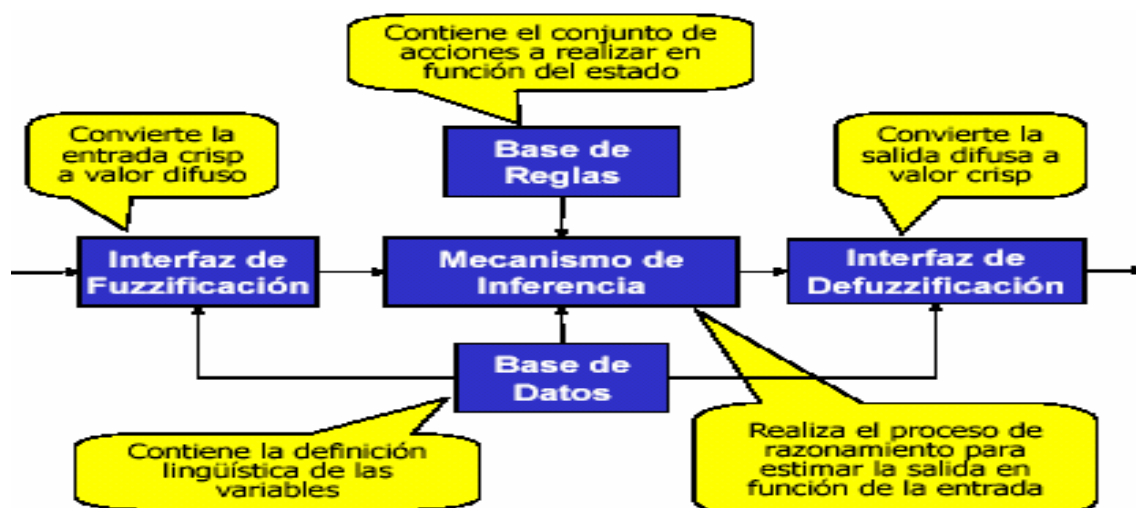
Entre los principales objetivos de los controladores borrosos tenemos:

- Clasificar los sistemas basados en reglas borrosas en base a su estructura y a la estructura de regla borrosa utilizada.
- Conocer ventajas e inconvenientes de cada uno de los tipos de sistemas basados en reglas borrosas.
- Comprender la función de las distintas componentes de un controlador borroso y las posibilidades de diseño disponibles para cada caso.
- Comprender el funcionamiento global de un sistema borroso para control.

#### 3.1.3.2.1 ESTRUCTURA DE UN CONTROLADOR BORROSO.

Esta es la estructura básica que posee un controlador borroso tradicional:

Figura 23 Estructura de un Controlador Borroso.



Según los enfoques tradicionales, el diseño de un regulador se basa en un análisis cabal del proceso y una vez que se posee un modelo cuantitativo de este, todas las decisiones se calculan usando algoritmos estrictamente numéricos. Es obvio que solo a procesos "Dóciles" al análisis cuantitativo le son aplicables técnicas de control de entidad numérica. Por otra parte, existe un número de procesos difíciles de controlar automáticamente en virtud de los obstáculos que se precisan para su modelado, sin embargo los operadores humanos exhiben magníficos resultados en su dirección. Esto, junto a la aparición de la teoría de los conjuntos borrosos y la asequibilidad de medios de cómputo poderosos, estimuló la investigación acerca de las estrategias de control de esos operadores, expresadas por medio de reglas heurísticas, para convertirlas en estrategias de control automático. El controlador lógico borroso resultante se basaría entonces en el modelo lingüístico de la estrategia del operador humano, es decir, en un modelo decisional del experto. La esencia de tal modelo es un programa basado en reglas, por lo que clasifica entre los llamados sistemas expertos [5].

### **3.1.3.3 FASES DEL CONTROL BORROSO**

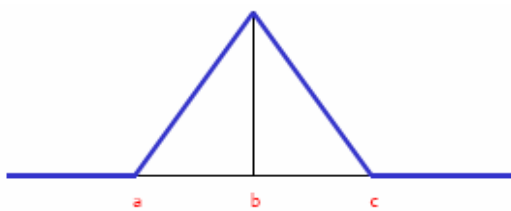
#### **3.1.3.3.1 EMBORRONADO.**

Es la transformación de la información determinista, enviada por el proceso al controlador lógico borroso, en información cualitativa que toma como referencia a conjuntos borrosos. También se evidencia de la explicación que esta transformación depende de parámetros fijados a priori al controlador lógico borroso asociados a la caracterización de las variables medidas del proceso, como variables lingüísticas, número de conjuntos y forma de las funciones de pertenencia.

### Funciones de pertenencia.

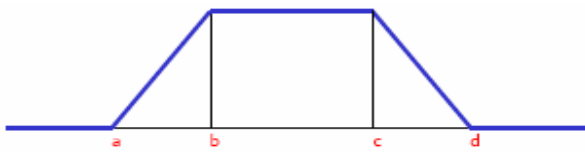
Entre las funciones de pertenencia más utilizadas tenemos las siguientes (ver figuras 24,25,26).

Figura 24 Función triangular



$$f(x; a, b, c) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & x > c \end{cases}$$

Figura 25 Función trapezoidal



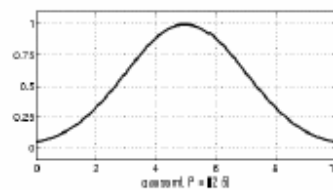
$$f(x; a, b, c, d) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & x > d \end{cases}$$

### Funciones gaussianas y de campana.

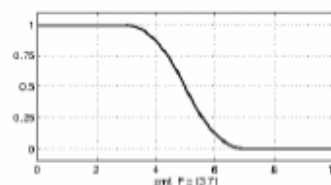
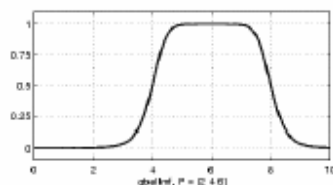
Figura 26 Funciones Gaussianas y de Campana.

- Funciones gaussianas

$$f(x; \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}}$$



- Otras: campana, S, Z, etc.



### 3.1.3.3.1.1 INTERFAZ DE EMBORRONADO.

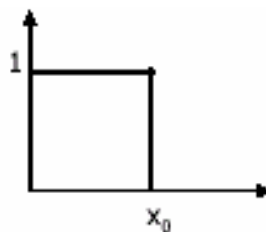
Para cada una de las entradas del sistema:

1. Adquirir los valores nítidos de las variables de entrada.
2. Trasladar los valores de las variables a los universos de discurso correspondientes
3. En función del tipo de sistema borroso, se puede optar por:
  - Convertir cada valor crisp en un conjunto borroso con grado de pertenencia igual a 1 para ese valor y 0 para el resto (fuzzy singleton)
  - Hacer corresponder a cada valor crisp el término lingüístico más adecuado
  - Calcular el grado de pertenencia a cada uno de los conjuntos borrosos utilizados para dicha variable lingüística.

Algunas posibilidades:

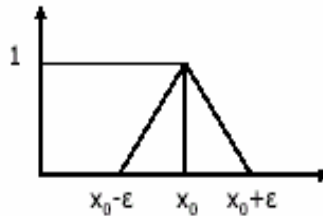
- El valor crisp se convierte en un conjunto borroso tipo singleton Ver figura 27
- Es la opción más sencilla y la más utilizada.
- Adecuado cuando la medición de las variables de estado es fiable.

Figura 27 Conjunto singleton.



- Se genera un conjunto borroso con centro el valor crisp y un soporte acorde con la incertidumbre de la medición Ver figura28.

Figura 28 Conjunto Triangular.



### 3.1.3.3.2 DESEMBORRONADO.

Es la transformación de un valor borroso dado después del proceso de inferencia, en un valor cuantitativo o crisp.

Interfaz de Desemborronado.

El Desemborronado transforma el conjunto borroso de salida en un valor crisp

Mediante métodos desarrollados por investigadores de la lógica borrosa.

Entre los principales métodos de desemborronado tenemos:

1. **Centro de área o centro de gravedad.**

Figura 29 Método del Centro de Área.



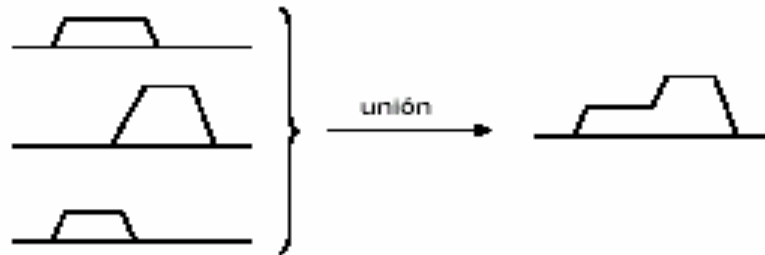
$$B^* = \frac{\sum_{i=1}^l y_i \cdot \mu_{B'}(y_i)}{\sum_{i=1}^l \mu_{B'}(y_i)}$$

Inconvenientes:

- El cálculo del conjunto borroso agregado es costoso.

- No tiene en cuenta el hecho de que dos áreas se solapen.

Figura 30 Agregación de Particiones activas.



## 2. Centro de sumas.

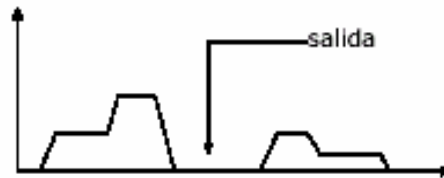
$$B = \frac{\sum_{i=1}^l y_i \cdot \sum_{k=1}^m \mu_{B^{(k)}}(y_i)}{\sum_{i=1}^l \sum_{k=1}^m \mu_{b^{(k)}}(y_i)}$$

- Considera la contribución de cada área de forma independiente. El método del centro de área toma la unión de los  $B^{(k)}$ , mientras que este método toma la suma de los conjuntos. De esta forma, si un área se repite, se considera de nuevo, evitando el problema de solapamiento visto anteriormente.
- No requiere el cálculo del conjunto borroso de salida.

## 3. Centro de mayor área.

Problema: si  $B'$  no es convexo, el centro de área y de sumas da una salida en la zona intermedia, donde el conjunto borroso tiene baja importancia.

Figura 31 Método del centro de Mayor Área.



- Solución: se determina el conjunto borroso con mayor área y se calcula su centro de gravedad.
- Es un método muy costoso computacionalmente hablando.

#### 4. Método de la altura.

- No requiere el cálculo del conjunto borroso de salida.
- Rápido.
- Requiere la definición del punto umbral (primer punto de un conjunto borroso con grado de pertenencia máximo).

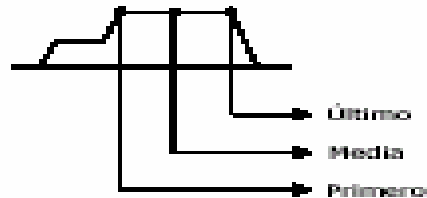
$$B = \frac{\sum_{k=1}^m c^{(k)} \cdot \mu_{B^{(k)}}(c^{(k)})}{\sum_{k=1}^m \mu_{B^{(k)}}(c^{(k)})}$$

Siendo  $c^{(k)}$  el valor umbral del conjunto borroso  $B^{(k)}$ .

#### 5. primero del máximo, ultimo del máximo, y media de los máximos.

Toma el valor más pequeño, más grande o medio del núcleo del conjunto borroso resultante.

Figura 32 Primer máximo, Último máximo y Media de Máximos.



Ventaja:

- coste computacional muy bajo.

Inconvenientes:

- Valor de salida menos representativo.
- Puede producir discontinuidades, es decir, genera una salida no continua para pequeños cambios en la entrada [7].

### 3.1.3.3.3 BASE DE CONOCIMIENTO.

Está formada por la Base de Reglas y la Base de Datos.

Parámetros de diseño implicados:

- Elección de las variables de estado del proceso y de control del mismo.
- Elección del conjunto de términos lingüísticos para las variables de estado y de control.
- Elección de la estructura del antecedente y consecuente de las reglas.
- Conjunto de reglas borrosas.

Formas de obtención de la base de conocimiento:

A través de experiencia experta, conocimiento de ingeniería de control o acciones de un operador de control experimentado.

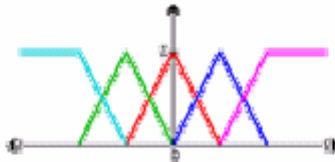
- Experto capaz de describir de forma lingüística sus reglas de decisión (factores de escala, semántica de los conjuntos borrosos, operadores implicados, etc.).
- Información extraída a partir de un cuestionario realizado al experto.
- Información extraída a partir de la observación de las acciones de control de un operador.
- Obtención basada en un modelo borroso.
- Obtención basada en aprendizaje automático (métodos ad hoc, computación evolutiva, redes neuronales, clustering, etc.).

#### 3.1.3.3.4 BASE DE DATOS.

Proporciona la información necesaria para el funcionamiento del módulo de Emborronado, de desemborronado y del motor de inferencia.

- Se define la semántica de cada variable lingüística.

Figura 33 Base de datos.



- También se pueden definir factores de escalado para extender o reducir el universo de discurso, así como cambiar la sensibilidad.

#### 3.1.3.3.5 BASE DE REGLAS.

Representa de forma estructurada la política de control experto

Se deben determinar los siguientes aspectos:

- ✓ Qué variables de estado y de control se considerarán.
- ✓ Qué estructura tendrá la regla borrosa.
- ✓ Qué conjunto de reglas (en su representación simbólica) se utilizará.

Ejemplo: Base de reglas para el controlador borroso de una aspiradora

Objetivo: Regular la fuerza de aspiración

¿Variables de entrada?

Cantidad de suciedad:

{muy sucio, sucio, algo sucio, casi limpio, limpio}.

- ¿Variable de control?

Fuerza:

{muy fuerte, fuerte, normal, débil, muy débil}.

Propuesta 1 para la base de reglas:

R1: SI la superficie está sucia ENTONCES la fuerza es fuerte.

R2: SI la superficie está algo sucia ENTONCES la fuerza es normal.

R3: SI la superficie está casi limpia ENTONCES la fuerza es débil.

R4: SI la superficie está limpia ENTONCES la fuerza es muy débil.

Se puede mejorar el rendimiento incluyendo más información.

¿Variables de entrada?

Cantidad de suciedad:

{muy sucio, sucio, algo sucio, casi limpio, limpio}.

Tipo de superficie:

{madera, caucho, alfombra}.

¿Variable de control?

Fuerza:

{muy fuerte, fuerte, normal, débil, muy débil}.

Propuesta 2 de base de reglas:

	LIMPIO	CASI LIMPIO	ALGO SUCIO	SUCIO	MUY SUCIO
Madera	Muy débil	Muy débil	Débil	Normal	Fuerte
Caucho	Muy débil	Débil	Normal	Fuerte	Muy fuerte
Alfombra	Débil	Normal	Normal	Fuerte	Muy fuerte

### 3.1.3.4 ANÁLISIS DE CONTROLADORES BORROSOS.

Objetivos principales:

- Entender la importancia de la etapa de análisis de un controlador borroso.
- Conocer las propiedades estáticas y dinámicas en un controlador borroso y su influencia en el comportamiento del sistema.
- Conocer algunos métodos de análisis de estabilidad para control borroso.
- Conocer formas de valorar la interpretabilidad de un controlador borroso.

Introducción.

- El análisis de la fiabilidad de un controlador borroso: se estudia la coherencia del conocimiento usado y su comportamiento dinámico.
  - ✓ Propiedades estáticas: relacionadas con la base de conocimiento usada.
  - ✓ Propiedades dinámicas: relacionadas con el comportamiento del controlador.
- Análisis de la interpretabilidad de un controlador borroso: se estudia en qué grado es fácil comprender el conocimiento empleado en un controlador borroso.

### Análisis de fiabilidad.

- Un controlador borroso no requiere un modelo matemático de la dependencia entre las salidas y entradas de control.
- La base de conocimiento se obtiene generalmente de conocimiento experto y es tan buena como el conocimiento humano que representa.
- Es fácil obtener un sistema borroso para control, pero es necesario un análisis y simulación para determinar la confianza en su fiabilidad.

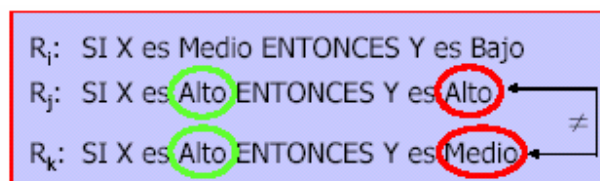
### Consistencia

Un conjunto de reglas es consistente si no contiene contradicciones.

Existen varias posibilidades para definir la contradicción:

- Un conjunto de reglas es inconsistente si existen dos reglas con igual antecedente pero distinto consecuente.

Figura 34 Consistencia de Reglas.

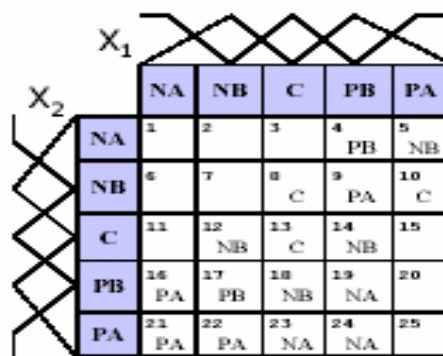


- Algo más flexible sería decir que son inconsistentes cuando, teniendo el mismo antecedente, los consecuentes son de intersección vacía.

### Compleitud.

- Un sistema borroso de control debe ser capaz de inferir una acción de control para cualquier estado del proceso.
- Un conjunto de reglas es completo si cualquier combinación de entrada produce como resultado un conjunto borroso de salida.
- Esto no significa que deban existir todas las reglas posibles, ya que debido al solapamiento de los conjuntos borrosos, algunas entradas pueden cubrirse indirectamente por reglas adyacentes.

Figura 35 Compleitud de Reglas.



- El subespacio de entrada correspondiente a R15 está cubierto indirectamente por R9, R10, R14 y R19.
- Sin embargo, los subespacios de entrada correspondientes a R1, R2 y R6 no los cubre ninguna otra regla y, por tanto, el conjunto de reglas del controlador no es completo.

## Continuidad.

Un conjunto de reglas borrosas es continuo si los consecuentes asociados a cada par de reglas vecinas tienen una intersección no vacía.

-Dos reglas borrosas se consideran vecinas si sólo difieren en el término lingüístico de una única variable de entrada, y dichos términos son consecutivos.

Figura 36 Continuidad de Reglas.

		NA	NB	C	PB	PA
NA	1	2	3	4	PB	NB
NB	6	7	8	C	PA	C
C	11	12	NB	C	NB	15
PB	16	17	18	NB	NA	20
PA	21	22	23	NA	NA	25

- Vecinas de R18: R13, R17, R19 y R23.
- Dada la siguiente semántica para el consecuente: desemborronado.



- R18 y R17 provocan un conjunto de reglas no continuo porque NB y PB tienen intersección vacía.

## Interacción.

$R_k$  : SI X es  $A_k$  ENTONCES Y es  $B_k$

- Si la entrada X toma el valor  $A_k$  se espera que la salida sea exactamente  $B_k$ .
- Existe interacción cuando no ocurre esto (por el efecto de otras reglas).
- Factores que contribuyen a la interacción:
  - Uso de un conjunto de reglas inconsistente.
  - Definición inapropiada de los conjuntos borrosos y sus funciones de pertenencia en el universo de discurso.
  - Elección inapropiada del método de inferencia y el operador de Desemborronado.

## Robustez.

La robustez refleja la habilidad del sistema borroso de control para resistir a cambios abruptos en las entradas.

A esta propiedad afecta:

- La resolución de control.
- La partición del espacio de entrada.
- El número de reglas de control.

Las funciones de pertenencia de los conjuntos borrosos

- Se ha demostrado experimentalmente que el uso de funciones de pertenencia trapezoidales colabora a que el sistema sea más robusto ante cambios drásticos.

## Estabilidad en Control Borroso.

Habitualmente, en un controlador borroso, se desconoce el modelo matemático de la planta, esto dificulta (o hace imposible, según el caso) la aplicación de las técnicas de análisis de estabilidad existentes en control clásico.

- El problema del análisis de estabilidad en sistemas borrosos para control no está resuelto completamente.

- En general, se puede decir que un controlador es estable si produce una respuesta acotada ante una entrada o distorsión acotada.

Técnicas de análisis:

1. Si se conoce el modelo de la planta:

- ✓ Planta lineal: técnica de descripción de la función, criterio de estabilidad L2.
- ✓ Planta no lineal: principio de invarianza / alfa-estabilidad, función de Lyapunov.

2. Sin modelo de la planta:

Análisis de indicadores de estabilidad.

Análisis de la trayectoria de disparo de reglas.

- Estas técnicas de análisis sirven para comprobar la estabilidad del controlador borroso. Si se detecta que es inestable, habrá que ajustar las funciones de pertenencia, las reglas borrosas y los operadores de inferencia para reducir la inestabilidad.

#### **Indicadores de estabilidad.**

- Un controlador es estable cuando se observan las siguientes características en la curva de respuesta del sistema.
  - ✓ Radio de exceso (overshoot) bajo.
  - ✓ Tiempo de reposo (settle time) corto.

#### **Análisis de interpretabilidad.**

- Generalmente, en los sistemas borrosos para control no es tan importante la interpretabilidad.
- No obstante, si se desea comprender el funcionamiento de la planta a través del controlador diseñado, la interpretabilidad sí es importante.
- La interpretabilidad también cobra importancia en control borroso cuando se desea integrar el conocimiento extraído por varios expertos.

- En muchas ocasiones, forzar un controlador borroso a ser interpretable lo hace más rígido y, generalmente, menos preciso.
- Esta restricción de interpretabilidad puede favorecer o perjudicar, según el caso, la fiabilidad del controlador.

### Trayectoria de disparo de reglas.

Figura 37

Orden de disparo de reglas típico en un sistema inestable

		$\Delta\epsilon$							
		NA	NM	NB	C	PB	PM	PA	
$\epsilon$	NA	NA	NA	NA	NM	NM	PM	NB	C
	NM	NA	NA	NA	NM	NB	C	PB	
	NB	NA	NA	NM	NB	C	PB	PM	
	C	NA	NM	NB	C	PB	PM	PA	
	PB	NM	NB	C	PB	PM	PA	PA	
	PM	NB	C	PB	PM	PA	PA	PA	
	PA	C	PB	PM	PA	PA	PA	PA	

Figura 38

Orden de disparo de reglas típico en un sistema estable

		$\Delta\epsilon$						
		NA	NM	NB	C	PB	PM	PA
$\epsilon$	NA	NA	NA	NA	NA	NM	PB	C
	NM	NA	NA	NA	NM	NB	C	PB
	NB	NA	NA	NM	NB	C	PB	PM
	C	NA	NM	NB	C	PB	PM	PA
	PB	NM	NB	C	PB	PM	PA	PA
	PM	NB	C	PB	PM	PA	PA	PA
	PA	C	PB	PM	PA	PA	PA	PA

### Sencillez de la estructura.

- El Sistema basado en reglas borrosas de tipo Mamdani es más fácilmente interpretable que el TSK debido a la estructura del consecuente.
- El uso de reglas con variables lingüísticas (donde existe un conjunto de términos común) en lugar de reglas con variables borrosas (donde cada variable en cada regla usa un conjunto borroso distinto) hace que el controlador sea más interpretable.

- Si se añaden modificadores lingüísticos, factores de escala, pesos para indicar el factor de importancia de cada regla, etc., la estructura del controlador se hace más flexible pero se pierde interpretabilidad.

### **Simplicidad de la base de conocimiento.**

- Un controlador borroso será más fácilmente interpretable cuanto más simple sea.
- La simplicidad viene determinada por el tamaño del conocimiento empleado.

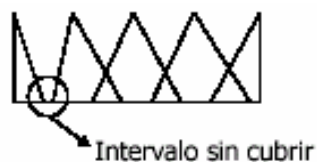
Número de términos lingüísticos reducido.

Número de reglas borrosas reducido.

### **3.1.3.5 ERRORES EN LA SEMÁNTICA.**

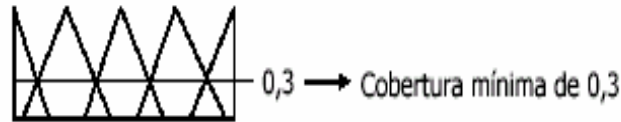
- Cada valor del universo de discurso debería pertenecer al menos a un término lingüístico.

Figura 39 errores en la Semántica. (Intervalos)



- Alternativamente, se puede definir un criterio más estricto estableciendo un nivel mínimo de grado de pertenencia a algún término lingüístico para cada valor del universo de discurso.

Figura 40 Errores en la Semántica (cobertura).



### Normalidad de los Conjuntos Borrosos.

- Cada conjunto borroso debería mostrar un grado de pertenencia máximo (generalmente 1) con al menos un valor del universo de discurso.
- Para ello, basta con que los conjuntos borrosos sean normales.

Figura 41 Normalidad de los Conjuntos Borrosos.



- Además, algunas veces es deseable que los valores de los extremos del universo de discurso tengan un grado de pertenencia máximo a algún conjunto borroso.

Figura 42 Grados de Pertenencia De un Conjunto borroso.

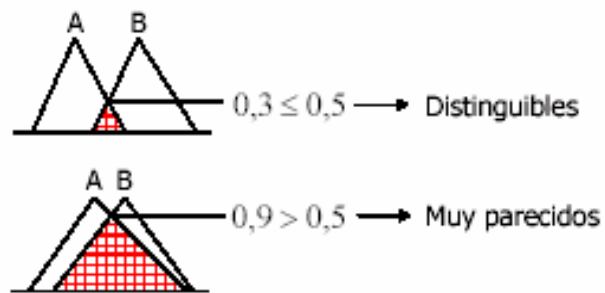


### Distinguibilidad entre los conjuntos borrosos.

- Cada término lingüístico debe tener un significado claro y el conjunto borroso asociado debe definir claramente un rango del universo de discurso.

- Las funciones de pertenencia deben ser suficientemente distintas unas de otras.
- Se pueden usar distintas medidas de similitud entre conjuntos borrosos. Por ejemplo, que la altura de la intersección sea inferior a un valor.

Figura 43 Distinguibilidad de un Conjunto Borroso.



## 4 DISEÑO Y SISTEMA COBOR 1.0

### 4.1 DISEÑO

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.

A continuación se presentan los diferentes elementos de diseño del proyecto Cobor 1.0.

#### 4.1.1 DIAGRAMA DE CLASES

Un diagrama es una presentación gráfica de un conjunto de elementos, que la mayoría de las veces se dibuja como un grafo conexo de nodos (elementos) y arcos (relaciones). Las Clases son la base para la construcción de un sistema orientado a objetos. Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica. Un Diagrama de Clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Una relación es una conexión entre elementos, en el modelado orientado a objetos las tres relaciones más importantes son las *dependencias* (son relaciones de uso, un cambio en un elemento puede afectar a otro elemento), *generalizaciones* (conectan clases generales con otras más especializadas, por ejemplo subclase/superclase o hijo/padre) y las *asociaciones* (son relaciones estructurales entre instancias, cuando existen elementos compuestos por otros elementos).

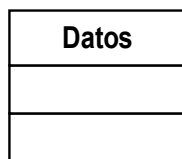
#### **4.1.1.1 DIAGRAMA DE CLASES DEL PROYECTO COBOR 1.0.**

Para la realización del proyecto COBOR 1.0 se elaboraron tres prototipos en los cuales se fueron agregando progresivamente las clases correspondientes para añadir las funcionalidades o componentes requeridos.

##### **4.1.1.1.1 PROTOTIPO NO 1.**

El diagrama de clases prototipo No 1 se muestra a continuación:

Figura 44 Diagrama de Clases del Prototipo No 1



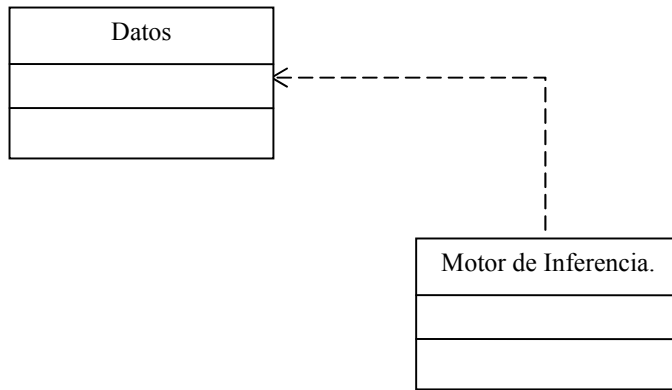
En este prototipo se implementaron los métodos de:

- Entrada de datos.
- Manipulación de datos permitiendo Agregar, Eliminar y Modificar conjuntos y reglas.
- Almacenamiento de datos en un archivo plano.
- Manipular subconjuntos de datos, con las funcionalidades dadas por el sistema.

##### **4.1.1.1.2 PROTOTIPO NO 2.**

El diagrama de clases de este prototipo se aprecia en la siguiente figura:

Figura 45 Diagrama de Clases del Prototipo No 2

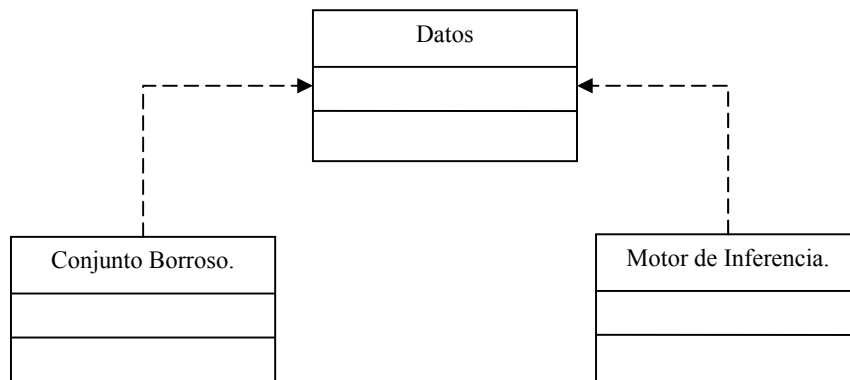


En este prototipo se agregó la clase Motor de inferencia la cual permite la fácil creación de la base de reglas borrosas con todas las características que se tienen en cuenta para el normal funcionamiento del controlador lógico borroso. Teniendo la posibilidad de crear las reglas de dos formas según la necesidad del experto que maneje el proceso.

#### 4.1.1.1.3 PROTOTIPO NO 3.

El diagrama de clases de este prototipo se aprecia en la siguiente figura:

Figura 46 Diagrama de Clases del Prototipo No 3



-En este prototipo se ha agregado la clase Conjunto Borroso la cual se maneja toda la información de los controladores borrosos, como lo son conjuntos borrosos, funciones de membresía.

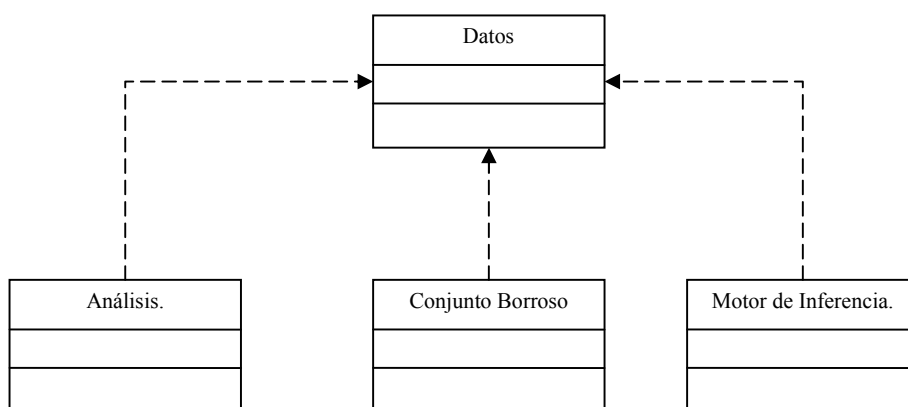
En este prototipo también se implementaron los siguientes métodos:

- Entrada de información específica.
- Emborronado, que nos permite transformar la información cuantitativa en información cualitativa, por medio de variables y modificadores lingüísticos.
- Buscar reglas activas.
- Operar las reglas activas.
- Operadores de implicación.
- Agregación de reglas.

#### 4.1.1.1.4 PROTOTIPO NO 4.

El diagrama de clases de este prototipo se aprecia en la siguiente figura:

Figura 47 Diagrama de Clases del Prototipo No 4

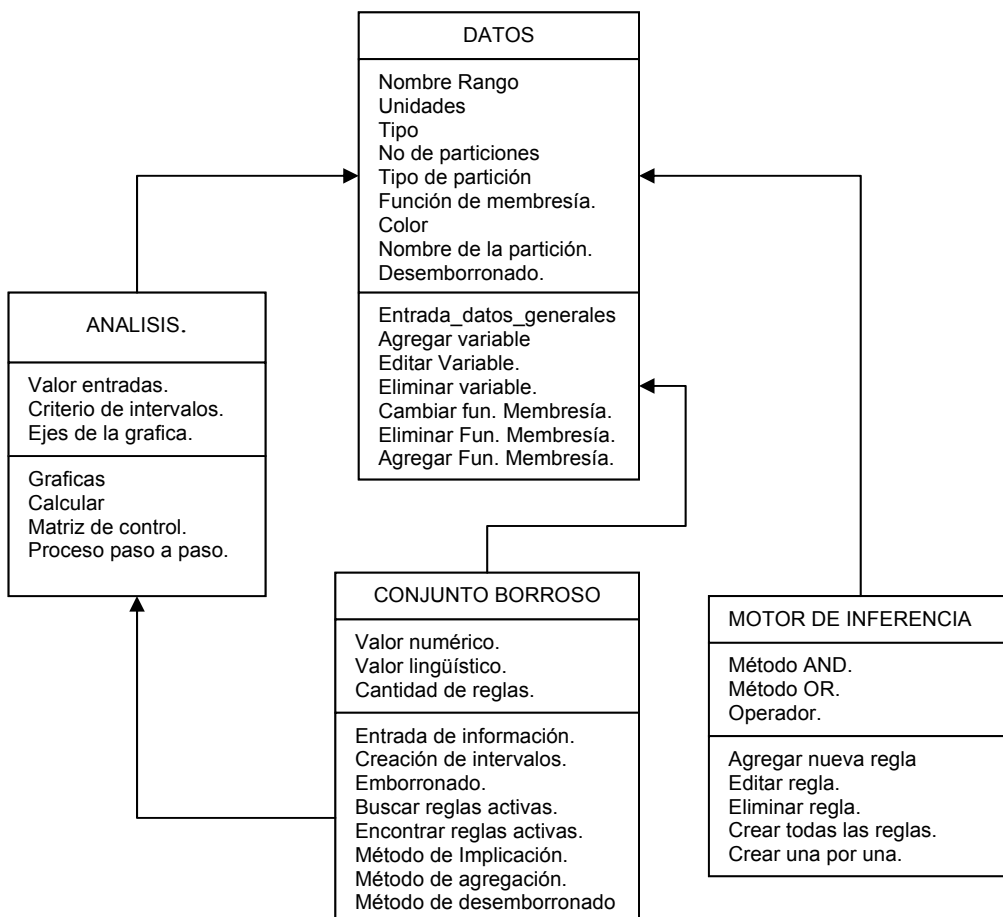


En este prototipo se agrego la clase Análisis la cual permite la toma de decisiones del usuario usando como referencia los resultados obtenidos por la herramienta, ya sea en forma gráfica o analítica, también se dará a conocer la solución del proceso paso a paso que es un objetivo del proyecto para dar mas flexibilidad y entendimiento a la herramienta.

#### 4.1.1.2 DIAGRAMA DE CLASES FINAL.

El diagrama de clases final se aprecia en la siguiente figura:

Figura 48 Diagrama de Clases Final.

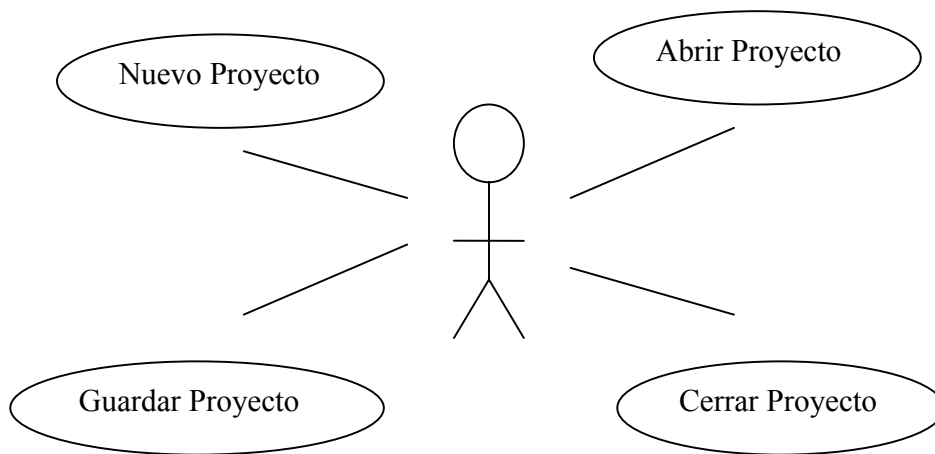


#### 4.1.2 DIAGRAMA DE CASOS DE USO.

##### 4.1.2.1 DIAGRAMA DE CASOS DE USO PROYECTO COBOR 1.0.

El diagrama de casos de uso del sistema para la construcción de controladores borrosos COBOR 1.0 se presenta en la figura.

Figura 49 Diagrama casos de uso de la aplicación COBOR 1.0



**Caso de uso: Nuevo Proyecto**

**Actor: Usuario**

**Propósito:** Permite al usuario crear un nuevo proyecto, para luego trabajar con el.

**Resumen:** El objetivo es ingresar los valores iniciales para el emborronado, desemborronado, implicación, agregación y crear las diferentes variables de entrada y salida.

**Curso Normal de Eventos.**

Acción de los actores	Respuesta del sistema
1. El usuario elige la opción "Nuevo Proyecto" en el menú Archivo.	2. Despliega una ventana donde el usuario debe introducir la información necesaria para la creación de un nuevo proyecto.
	3. Despliega un cuadro de diálogo para guardar archivos
4. El usuario proporciona la ubicación donde se va a guardar el archivo plano que contiene la información del proyecto que va a ser creado.	5. Guarda los datos, del proyecto en la ubicación indicada por el usuario.
	6. Activa los botones inhabilitados en el menú principal de la aplicación.

**Caso de Uso: Abrir Proyecto**

**Actor:** Usuario

**Propósito:** Abrir un proyecto ya existente.

**Resumen:** El usuario podrá utilizar proyectos creados con anterioridad.

**Curso Normal de Eventos.**

Acción de los usuarios	Respuesta del sistema
1. El usuario elige la opción "Abrir"	2. El sistema despliega un cuadro de

en el menú Archivo.	diálogo de apertura de archivos.
3. El usuario selecciona el archivo que desea abrir.	4. El sistema verifica el formato del sistema y carga el archivo.
	5. El sistema crea las variables de entrada y salida además de los valores iniciales para el emborronado, desemborronado, implicación, agregación

### Caso de uso: Guardar Proyecto

**Actor: Usuario**

**Propósito:** Permite al usuario guardar los cambios hechos al proyecto.

**Resumen:** El usuario puede guardar los cambios hechos al proyecto, para luego poder trabajar con el.

### Curso Normal de Eventos.

Acción de los actores	Respuesta del sistema
1. El usuario elige la opción "Guardar" en el menú Archivo.	2. El sistema despliega el cuadro de dialogo de guardar archivo.
3. El usuario digita el nombre del archivo.	
4. El usuario proporciona la ubicación donde se va a guardar el archivo plano que contiene la información del proyecto que va a ser creado.	5. Guarda los datos, del proyecto en la ubicación indicada por el usuario.

**Caso de uso: Cerrar Proyecto**

**Actor: Usuario**

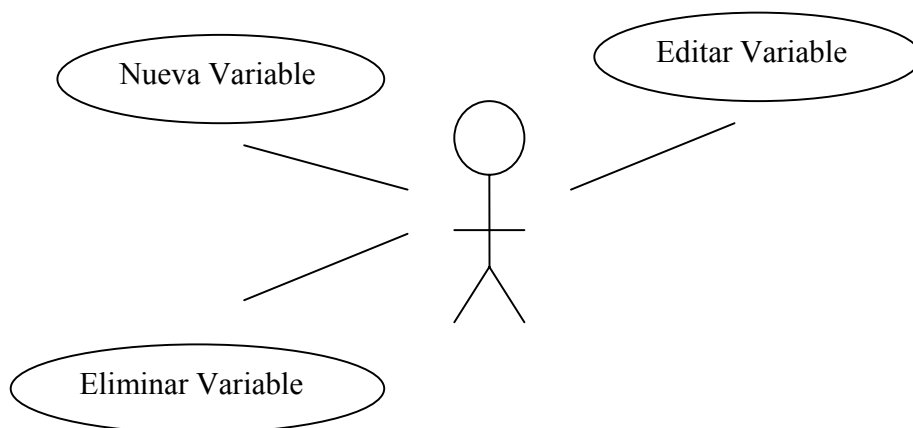
**Propósito:** Permite al usuario cerrar la aplicación.

**Curso Normal de Eventos.**

Acción de los actores	Respuesta del sistema
1. El usuario elige la opción "Salir" en el menú Archivo.	2. El sistema muestra un mensaje donde le indica al usuario si desea guardar o no el proyecto actual.
3. El usuario decide la operación a realizar: <b>Guardar Proyecto</b> <b>No guardar los cambios</b>	4. El sistema responde adecuadamente ante la operación seleccionada.

**Casos de Uso: Variables Borrosas.**

Figura 50 Diagrama de Casos De Uso: Variables Borrosas



### Caso de uso: Nueva Variable

**Actor: Usuario**

**Propósito:** Permite al usuario agregar una nueva variable, al proyecto en curso.

**Resumen:** El objetivo es ingresar los valores y características de la nueva variable, ya sea de entrada o salida.

#### Curso Normal de Eventos.

Acción de los actores	Respuesta del sistema
1. El usuario elige la opción "Agregar Variable" en el menú Variables	2. Despliega una ventana donde el usuario debe seleccionar el tipo de variable que desea crear
3. El usuario selecciona el tipo de variable que desea agregar. Entrada o salida.	4. El sistema genera una ventana donde el usuario debe introducir la información necesaria sobre la nueva variable.
5. El usuario ingresa la información necesaria para la creación de la nueva variable.	6. Agrega la nueva variable al proyecto abierto.

### Caso de uso: Editar Variable

**Actor: Usuario**

**Propósito:** Permite al usuario editar las variable existentes en el proyecto en curso.

**Resumen:** El objetivo es permitir al usuario modificar las características de las variables ya creadas.

**Curso Normal de Eventos.**

Acción de los actores	Respuesta del sistema
1. El usuario elige la opción "Editar" en el menú Variables	2. Despliega una ventana donde el usuario debe seleccionar el tipo de variable que desea editar.
3. Selecciona el tipo de variable que desea editar: entrada o salida.	4. Según el tipo de variable elegido, muestra todas las variables que existen en el proyecto, y queda en espera de que se seleccione una.
5. Selecciona la variable que quiere editar.	6. Muestra las características propias de la variable seleccionada en cuadros que permiten modificar las mismas
	7. Queda en espera de que el usuario modifique las características de la variable seleccionada.
8. Modifica las características de la variable en cuestión.	9. Hace los cambios indicados por el usuario a la variable en cuestión.
	10. Queda en espera de la siguiente acción del usuario.

**Caso de uso: Eliminar Variable**

**Actor: Usuario**

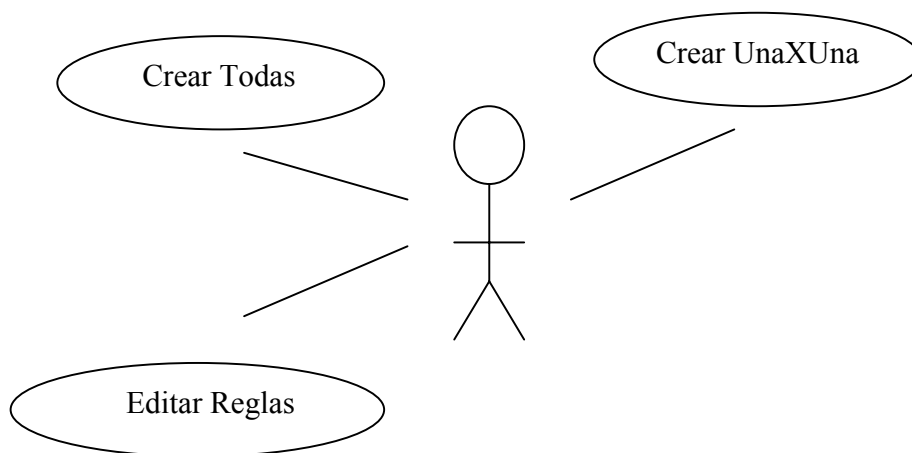
**Propósito:** Permite al usuario eliminar una variable del proyecto en curso.

**Curso Normal de Eventos.**

Acción de los actores	Respuesta del sistema
1. El usuario elige la opción "Eliminar" en el menú Variables	2. Despliega una ventana donde el usuario debe seleccionar el tipo de variable que desea eliminar.
3. El usuario elige el tipo de variable que desea eliminar.	4. Según el tipo de variable escogida, muestra una ventana con las variables existentes.
5. Selecciona la variable que desea eliminar.	6. Muestra un cuadro de dialogo donde confirma si en realidad se desea o no eliminar o no la variable seleccionada.
7. Selecciona la opción que se ajuste a su necesidad.	8. Según la opción tomada por el usuario : elimina NO elimina La variable en cuestión.
	9. Queda en espera de la siguiente acción del usuario.

## Casos de Uso: Motor de Inferencia.

Figura 51 Diagrama Casos De Uso: Motor de Inferencia



### Caso de uso: Crear Todas

**Actor:** Usuario

**Propósito:** Permite al usuario crear la base de reglas al proyecto en curso.

**Resumen:** El objetivo es ingresar todas las posibles reglas que puede manejar la base de reglas según las variables del proyecto.

### Curso Normal de Eventos.

Acción de los actores	Respuesta del sistema
1. El usuario elige la opción "Crear Todas" en el menú Reglas.	2. Busca las características de las variables de entrada y las

	variables de salida, para crear las posibles combinaciones.
	3. Muestra todas las combinaciones en una ventana.
4. Actualiza los valores de las variables de salida según sea el caso.	5. Guarda los cambios hechos a los valores de las variables de salida.
	6. Queda en espera a la siguiente acción del usuario.

**Caso de uso: Crear Una X Una**

**Actor: Usuario**

**Propósito:** Permite al usuario crear la base de reglas del proyecto en curso.

**Resumen:** El objetivo es ingresar reglas que puede manejar la base de reglas según las variables del proyecto, y según las necesidades del usuario.

**Curso Normal de Eventos.**

Acción de los actores	Respuesta del sistema
1. El usuario elige la opción "Una x Una" en el menú Reglas.	2. Busca las características de las variables de entrada y las variables de salida.
	3. Muestra los valores de las variables en una ventana.
4. Ingresar las reglas una por una según su criterio.	5. Guarda las reglas creadas en la base de reglas.
	6. Queda en espera a la siguiente acción del usuario.

**Caso de uso: Editar Reglas**

**Actor: Usuario**

**Propósito:** Permite al usuario editar la base de reglas al proyecto en curso.

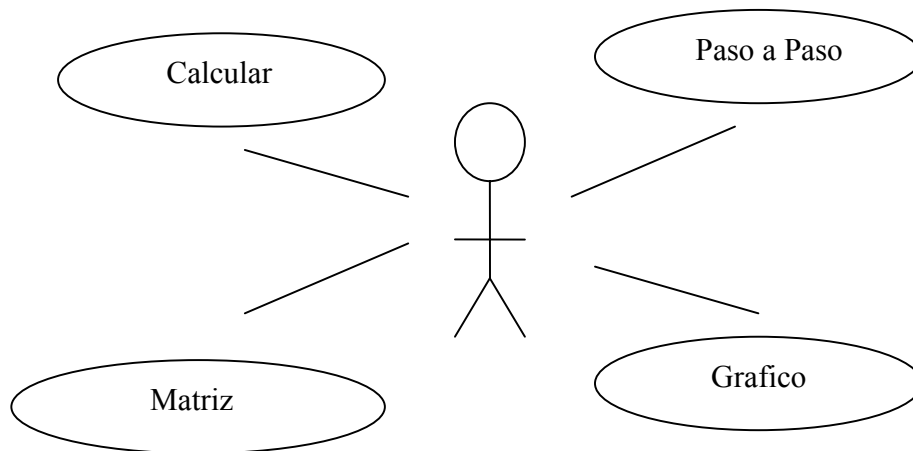
**Resumen:** El objetivo es editar las reglas que puede manejar la base de reglas.

**Curso Normal de Eventos.**

Acción de los actores	Respuesta del sistema
1. El usuario elige la opción "Editar" en el menú Reglas.	2. Busca las reglas de la base de reglas y las muestra en una ventana.
3. Hace los cambios que considera necesarios.	4. Guarda los cambios hechos por el usuario.
	5. Queda en espera de la siguiente acción del usuario.

**Casos de Uso: Analizador.**

Figura 52 Diagrama de Casos De Uso: Analizador.



**Caso de uso: Calcular**

**Actor: Usuario**

**Propósito:** Permite al usuario dar la respuesta a un caso específico.

**Resumen:** El objetivo es que el usuario ingrese valores para un caso particular y el sistema calcule el valor de las variables de salida.

**Curso Normal de Eventos.**

Acción de los actores	Respuesta del sistema
1. El usuario elige la opción "Calcular" en el menú Analizador.	2. Despliega una ventana donde el usuario debe introducir los valores de las variables de entrada para el caso en particular que desea resolver.
3. Ingresar los valores de las variables de entrada.	4. Calcula los valores de las variables de salida.
	5. Muestra los valores de la variable de salida
	6. Queda en espera de la siguiente acción del usuario.

**Caso de uso: Paso a Paso**

**Actor: Usuario**

**Propósito:** Permite al usuario dar la respuesta a un caso específico. Mostrando el proceso completo

**Resumen:** El objetivo es que el usuario ingrese valores para un caso particular y el sistema calcule el valor de las variables de salida, mostrando el proceso que se sigue para tal fin.

**Curso Normal de Eventos.**

Acción de los actores	Respuesta del sistema
1. El usuario elige la opción "Paso a Paso" en el menú Analizador.	2. Despliega una ventana donde el usuario debe introducir los valores de las variables de entrada para el caso en particular que desea resolver
3. Ingresa los valores de las variables de entrada.	4. Calcula los valores de las variables de salida.
	5. Muestra los valores de la variable de salida.
	6. Queda en espera de la siguiente acción del usuario.

**Caso de uso: Matriz**

**Actor: Usuario**

**Propósito:** Permite al usuario generar la matriz de acciones.

**Resumen:** El objetivo es generar la matriz de acciones para las posibles combinaciones de valores en la variable de entrada.

**Curso Normal de Eventos.**

Acción de los actores	Respuesta del sistema
1. El usuario elige la opción "Matriz" en el menú Analizador.	2. . Despliega una ventana donde el usuario debe introducir los intervalos de valores para los cuales desea calcular la matriz.
3. Ingresar los intervalos de valores de las variables de entrada.	4. Calcula los valores de las variables de salida.
	5. Muestra la matriz en una tabla.
	6. Queda en espera de la siguiente acción del usuario.

### Caso de uso: Paso a Paso

#### Actor: Usuario

**Propósito:** Permite al usuario dar la respuesta a un caso específico. Mostrando el proceso completo

**Resumen:** El objetivo es que el usuario ingrese valores para un caso particular y el sistema calcule el valor de las variables de salida, mostrando el proceso que se sigue para tal fin.

#### Curso Normal de Eventos.

Acción de los actores	Respuesta del sistema
7. El usuario elige la opción "Paso a Paso" en el menú Analizador.	8. Despliega una ventana donde el usuario debe introducir los valores de las variables de entrada para el caso en particular que desea resolver
9. Ingresar los valores de las	10. Calcula los valores de las

variables de entrada.	variables de salida.
	11. Muestra los valores de la variable de salida.
	12. Queda en espera de la siguiente acción del usuario.

### Caso de uso: Grafico

#### Actor: Usuario

**Propósito:** Generar un grafico del comportamiento del sistema. **Resumen:** Generar un grafico del comportamiento del sistema tomando dos variables de entrada y una de salida.

#### Curso Normal de Eventos

Acción de los actores	Respuesta del sistema
1. El usuario elige la opción "Grafico" en el menú Analizador.	2. Despliega una ventana donde el usuario debe seleccionar dos variables de entrada y una de salida.
3. Selecciona dos variables de entrada y una de salida.	4. Calcula los valores de las variables de salida.
	5. Grafica los datos obtenidos.
	6. Muestra el grafico generado.
	7. Queda en espera de la siguiente acción del usuario.

### **4.1.3 PRUEBAS DE INGENIERÍA DE SOFTWARE.**

Estas pruebas fueron realizadas con el fin de mejorar la calidad del software tanto en facilidad de uso como en eficiencia de su desempeño.

#### **4.1.3.1 PRUEBAS ALFA.**

Estas pruebas se llevan a cabo en un entorno controlado, en presencia del desarrollador. Para la realización de las pruebas alfa en el sistema COBOR 1.0 se elaboró una guía para que el usuario final la desarrollara con el fin de registrar errores y los problemas de uso de la herramienta.

##### **Guía para pruebas alfa.**

1. Se realiza una pequeña exposición sobre el funcionamiento del paquete y el objetivo para el cual ha sido diseñado.
2. Entregar varios proyectos con diferentes variables de entradas y salidas al usuario.
3. Solicitar al usuario probar los proyectos variando los conjuntos borrosos y utilizando diferentes funciones de membresía para observar los resultados.
4. Construir con el usuario la base de reglas borrosas par entender la facilidad de cada uno de los métodos dados por la herramienta, y verificar cual es el mas optimo según sea el caso.
5. Mostrar al usuario los métodos de desemborronado y que el interactúe con cada uno de ellos para que observe cual es el más eficiente.
6. Construir un controlador borroso basado en un proyecto ya guardado.
7. Introducir parámetros inválidos para comprobar las diferentes validaciones utilizadas.
8. Realizar Observaciones y recomendaciones para todo el proceso de ejecución del sistema.

Se realizaron pruebas indicando a los usuarios en términos generales el funcionamiento del sistema, sus áreas de utilización y las características de construcción de controladores borrosos, dentro de los resultados se obtuvieron las siguientes sugerencias:

- Dar una mejor explicación sobre los conceptos de la lógica borrosa para la creación de los controladores.
- Que el sistema sea un poco más flexible, en cuanto a la construcción de los conjuntos borrosos.
- Sería bueno proveer al sistema de un análisis de sensibilidad con los resultados obtenidos.
- Que tenga graficas para observar la simulación del proceso de control borroso y compararlo con otros controladores.

En general se observó que el sistema trabaja bien aunque se tienen algunas falencias, la mayoría de ellas de carácter conceptual puesto que los usuarios desconocían las operaciones básicas para trabajar con Lógica borrosa, se tratarán de corregir estas anomalías dependiendo del tiempo que se tenga para la entrega del proyecto.

## **4.2 SISTEMA COBOR 1.0.**

### **4.2.1 ARQUITECTURA DE COBOR 1.0.**

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.

#### **4.2.2 CARACTERÍSTICAS DEL SISTEMA**

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 del 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.

#### **4.2.3 CARACTERÍSTICAS DE LOS COMPONENTES.**

Cada uno de los componentes tiene funcionalidades que permiten ayudar a todo el proceso global para la creación de los controladores borrosos.

### **Componente de carga de datos**

Este componente esta formado por la clase Datos permitiendo la carga a partir de archivos planos ya sean propios del sistema o en el formato de COBOR 1.0, tomando esa información recibida para su manipulación por medio de las leyes de la lógica borrosa, esto con el fin de generar las soluciones requeridas por los usuarios, después de los procesos propios de un controlador borrosos como son:

- ✓ Emborronado.
- ✓ Motor de inferencia.
- ✓ Desemborronado.

### **Componente de creación de los conjuntos borrosos.**

Este componente brinda facilidades para la construcción de los conjuntos borrosos de entrada y salida que son la base para la creación del controlador borroso, permitiendo la manipulación de las funciones de membresía, los implicadores, la construcción de la base de reglas y otras características propias del sistema.

### **Componente de visualización de resultados.**

Para este componente se utilizó un componente grafico que permite la visualización, para una fácil comprensión y asociación de los resultados, realizando todo esto por métodos implementados durante el desarrollo del proyecto.

### **Motor e Interfaz a Usuarios**

En el motor del sistema se centralizan o agrupan instancias de los diferentes componentes base del sistema, permitiendo la realización del proceso en forma global. La

interfaz a usuarios comprende todos los componentes y formularios gráficos con los cuales se interactúa, construida de forma amigable y de fácil manejo.

#### **4.2.4 PARÁMETROS ESPECÍFICOS.**

Para la construcción de los controladores borrosos se requieren una serie de parámetros que se explican a continuación:

- Mínimo numero de variables: la herramienta exige como mínimo la inclusión de dos variables de entrada y una variable de salida al proceso ya que es la única restricción para operar la herramienta.
- Máximo numero de variables: La herramienta da un rango máximo de variables de entrada al controlador que es de nueve.
- Base de reglas: la base de reglas tendrá dos opciones para su construcción, generar todas las reglas automáticamente, o generar las reglas una por una, esto se deja a elección del experto.
- Funciones de membresía: la funciones de membresía que se implementaron fueron tres, Triangular, trapezoidal y gaussiana. Que son las más utilizadas por los expertos en lógica borrosa, debido a su sencillez y buena transformación de los datos numéricos. aunque se deja abierta la posibilidad de implementar otro tipo de funciones no se implementaron ya que las más utilizadas son las anteriormente enunciadas.
- Método de desemborronado: Los métodos de desemborronado implementados son los siguientes, centro de área, Media de máximos, Primer máximo, y ultimo máximo. La razón mas importante es que son los mas eficientes computacionalmente hablando, y además los que tienen mas exactitud en los resultados obtenidos, se da la posibilidad de implementar otros métodos pero los implementados en esta herramienta son lo considerados como los mas eficientes.

#### **4.2.5 REQUERIMIENTOS DE HARDWARE Y SOFTWARE.**

Para el desarrollo del sistema COBOR 1.0 se utilizó un equipo con las siguientes características:

- Procesador AMD de 750 Mhz.
- 128 MB en RAM.

Para la ejecución del sistema COBOR 1.0 se requieren como mínimo 64 MB de memoria RAM, se recomienda tener 128 MB tener procesador Pentium o superior. Esta herramienta funciona en sistemas operativos Windows series 9X, NT, 2000, XP.

## **5. APLICACIONES.**

### **5.1 SISTEMAS DE CONTROL BASADOS EN REGLAS BORROSAS.**

Las primeras aplicaciones de la teoría Borrosa fueron principalmente industriales, tales como el control de procesos en cementeras. Mas tarde en 1987, se puso en servicio en Sendai, al Norte del Japón, el primer metro controlado mediante Lógica Borrosa. Los controladores basados en esta lógica, hicieron mucho más confortables los viajes en Metro, gracias a las suaves frenadas y aceleraciones. En efecto, todo lo que debe hacer el conductor es apretar el pulsador de marcha.

La Lógica Borrosa se incluyo en ascensores para reducir el tiempo de espera. A partir de 1990 se la comienza a implementar en los controles de inyección electrónica de carburante y en los sistemas de control de guiado automático de coches, haciendo los controles complejos más eficientes y fáciles de utilizar como las lavadoras “Borrosas” o “Fuzzy” que trabajan con más de 400 ciclos preprogramados; a pesar de su complejidad tecnológica resultan más fáciles de operar que las lavadoras tradicionales. El usuario solo pone en marcha la lavadora, el resto queda en manos del control Borroso.

El evalúa automáticamente el material, el volumen, la suciedad de la ropa, elige el ciclo óptimo de lavado, así como el caudal de agua que ha de emplear. En acondicionadores de aire, la lógica Borrosa ahorra energía dado que comienza a enfriar con mayor potencia solo cuando un sensor detecta la presencia de personas en la habitación, permaneciendo hasta entonces en una situación prácticamente de “stand-by”. En USA la NASA esta trabajando para implementar un control borroso en las condiciones del espacio, una tarea extremadamente difícil para lógicas tradicionales. Se ha empleado en los controles de cámaras de foto y video, e incluso en sistema de

gestión financiera, en tratamientos de depuración de agua, en control de robots, en detección y control de averías, y en un interminable número de aplicaciones. En las actividades domésticas cotidianas se inventó un sistema de ventilación. Utiliza control borroso para conmutar un ventilador según los conocimientos de cantidad de polvo, olores, temperatura y humedad ambiente. El baño borroso, por ejemplo tiene un controlador que mantiene el agua a la temperatura ideal para el usuario, “ni muy caliente, ni muy fría”. Si en un principio el agua está templada, añade el agua caliente en menor proporción que si estuviese fría, evitando elevar demasiado la temperatura del agua.

La teoría de los conjuntos borrosos puede aplicarse en cualquier campo en donde surgen problemas de incertidumbre o de ambigüedad en la información, lo que hace muy amplio el rango de áreas en donde podría aplicarse satisfactoriamente.

Actualmente la lógica borrosa se usa con mayor frecuencia en aplicaciones relacionadas con el control, y se ha mostrado mayor efectividad en procesos complejos, no lineales y dependiente de un experto humano.

## **5.2. APLICACIONES EN LA INGENIERÍA**

En el campo de la ingeniería, la teoría de conjuntos borrosos ha tenido varias aplicaciones y la principal de estas es el Control Borroso. El propósito del control automático es alcanzar o mantener algún estado deseado de un proceso o sistema, monitoreando ciertas variables de estado y tomando las acciones de control apropiadas. Muchos procesos industriales hacen uso de este control en reemplazo o en complemento de un operador humano. El diseño de un controlador automático usualmente requiere un modelo matemático preciso del proceso o sistema en cuestión, y para muchos procesos complejos, la construcción de un modelo se hace difícil o imposible, debido a la variación no-lineal del tiempo, o a una deficiencia de los datos disponibles. En estos casos, un operador humano ofrece un control superior al de un controlador convencional. Como una alternativa al modelo preciso del sistema, una descripción imprecisa del proceso o de la

forma de control puede ser elaborada por el operador humano haciendo uso de conceptos vagos o borrosos.

Es así como las reglas de control heurístico, pueden tomar la forma:

**IF** Temperatura es muy alta **AND** presión es ligeramente baja

**THEN** cambio de calor debe ser ligeramente pequeño.

Donde la temperatura y la presión son las variables observadas del estado del proceso y el cambio de calor es la acción a ser tomada por el controlador. Los términos vagos “muy alta”, “ligeramente baja”, y “ligeramente negativo” se pueden representar mediante conjuntos borrosos definidos sobre el universo de discurso de los valores de temperatura y presión, y los valores del cambio de calor respectivamente.

Este tipo de regla lingüística ha formado las bases para el diseño de prototipos de controladores borrosos muy diferentes, muchos para procesos industriales incluyendo hornos de cemento, procesos de intercambio de calor y máquinas de vapor entre otros. Además, se han creado controladores borrosos para el control de velocidad de automóviles, intersecciones de tráfico, robots y como piloto automático de un avión.

### **5.2.1 SISTEMA DE CONTROL DEL CALOR DE UNA CASA.**

Para maximizar la economía y el confort en el sistema de calefacción de una casa, una compañía alemana está usando control borroso en una nueva generación de controladores de hornos.

El controlador asegura adaptación óptima a diferentes y cambiantes necesidades de los clientes usando un sensor menos que la generación anterior.

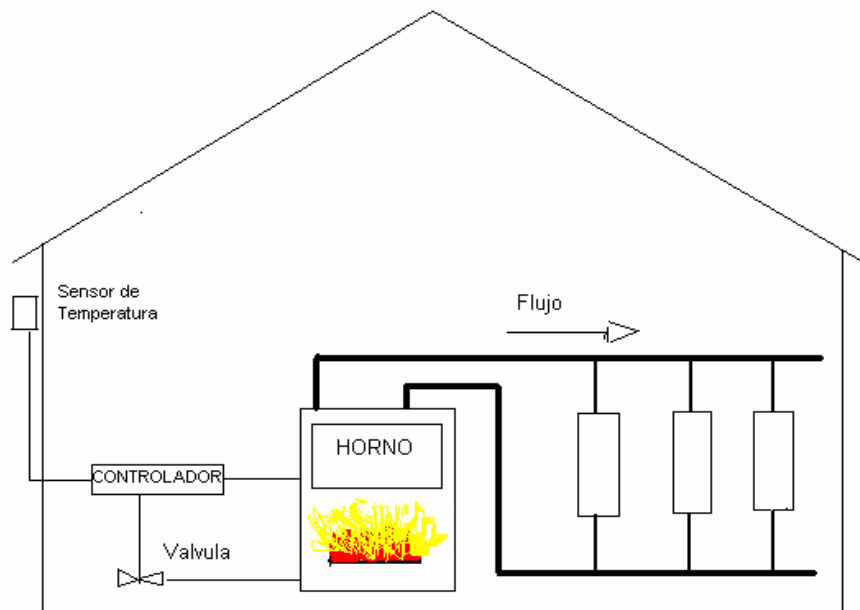
La mayoría de las casas europeas tienen sistemas centralizados de calefacción que usan hornos de combustible diesel para calentar el suministro de agua (caldera).

De la caldera, el agua caliente es distribuida por un sistema de tuberías a los radiadores individuales en las habitaciones de la casa. Para satisfacer las diferentes necesidades de calefacción de los clientes, la temperatura del agua calentada por el horno debe ajustarse

constantemente con relación a la temperatura externa. Para medir la temperatura exterior, se instala un sensor en el exterior de la casa.

El controlador por si mismo realiza un acción de encendido y apagado. Si la temperatura del agua en el horno baja a 2 grados kelvin por debajo de la temperatura fija, la válvula del combustible se abre y el sistema de encendido inicia el proceso de quemado. Cuando la temperatura del agua en el horno alcanza los “1-2” grados kelvin sobre la temperatura fija, la válvula de combustible se cierra.

Figura 53 Control de Calor de una Casa.



Aunque la estructura de este bucle de control es muy simple, la tarea de determinar la temperatura fija apropiada para el horno, no lo es. Debido a que la máxima disipación de calor de los radiadores de las habitaciones depende de la temperatura del agua que llega (aprox., la temperatura del horno), el punto fijo para la temperatura del agua en el horno, nunca debe ser tan bajo que no pueda calentar la casa cuando sea necesario. Por otro

lado, fijar la temperatura del horno demasiado alta, puede resultar en pérdidas de energía en el horno y en el sistema de tuberías. Por estas razones, la temperatura del horno debe ser fijada cuidadosamente para asegurar confort al usuario y eficiencia en la energía.

En aquellos días cuando la mayoría de las casas tenían poco aislamiento térmico, se asumía que la energía a ser liberada por un sistema de calefacción fuera dependiente de la temperatura exterior era apropiada. Hoy esto es obsoleto, debido a los crecientes costos de la energía y las cuestiones ambientales, las casas modernas están construidas con aislamiento incorporado. Por consiguiente, para lograr alta eficiencia, la temperatura exterior no es el único parámetro que refleja la cantidad de energía requerida. Otros factores tales como ventilación, apertura de puertas y ventanas y el estilo de vida personal, también debe considerarse.

### **5.2.2 ROBOTS MÓVILES Y NAVEGACIÓN AUTÓNOMA.**

Los robots móviles pueden caracterizarse en términos de su movilidad, autonomía, inteligencia y conocimiento previo del entorno.

El sistema de navegación del robot incluye tareas de planificación, percepción y control de movimientos. La primera comprende la planificación de la misión, planificación de otra ruta, planificación de la trayectoria (zhao 94), y las funciones específicas para evitar obstáculos imprevistos, no considerados en las planificaciones anteriores.

La percepción se ocupa de tareas relacionadas con la detección de obstáculos, construcción de mapas, modelado del entorno, y reconocimiento de objetos.

Por último, el control de movimientos convierte las órdenes de control en acciones, utilizando para ello la diferencia entre la posición deseada y la posición actual del robot (esta última a menudo estimada). Se incluyen también tareas de monitorización de la ejecución.

En algunos casos, en vez de utilizarse directamente la variable de control en los consecuentes de las reglas, se emplea una combinación lineal de los valores de las variables de situación, empleando coeficientes, que se calculan mediante la identificación

de las actuaciones de un operador humano trabajando en simulación. En este punto, conviene mencionar el interés de la aplicación de técnicas de aprendizaje, en cualquier caso, una vez obtenidos los valores inferidos por cada regla, la acción final de control resulta de una suma ponderada normalizada de los resultados de cada regla.

Por otra parte, cabe mencionar que el conjunto de reglas puede descomponerse en subconjuntos relativos a operaciones diferentes (conducir hacia delante, girar a la izquierda en intersección...), o bien maniobras diferentes (aparcar en paralelo, aparcar en batería).

Las aplicaciones mencionadas anteriormente cabe considerarlas como de control borroso directo. Luego la aplicación de estos modelos pueden realizarse directamente (sustituyendo los métodos geométricos y dinámicos), o en combinación de ellos. De esta forma, es posible compensar errores de seguimiento mediante reglas que tienen como antecedentes la desviación de ángulo y la variación de la desviación del ángulo y como consecuentes la modificación de la dirección para compensar el error.

Así mismo, otra estrategia muy interesante consiste en seleccionar mediante técnicas de razonamiento aproximado los parámetros de control, que típicamente se eligen mediante consideraciones heurísticas. Por ejemplo, los métodos geométricos o dinámicos de seguimiento de trayectorias tienen como parámetro importante el número de punto, o la distancia sobre el camino que pretende seguir (look ahead). Este parámetro puede elegirse mediante técnicas de razonamiento aproximado, teniendo en cuenta las características del camino, la velocidad, y la posición del vehículo con respecto al camino. Los comentarios anteriores se refieren fundamentalmente al control de la dirección. Existe también el problema del control de la velocidad del vehículo. Aunque hay un evidente acoplamiento cinemático y dinámico con el anterior, se ha demostrado que en diversas aplicaciones prácticas pueden realizarse desacoplamientos. En cualquier caso, el problema de planificación y control de velocidad es notablemente complejo, y suele realizarse con criterios heurísticos. Por consiguiente, la aplicación de técnicas de razonamiento aproximado resulta particularmente interesante.

Si bien la mayor parte de los robots autónomos se basan en topologías clásicas, haciendo uso de ruedas, también se han estudiado otras posibilidades. Así, para el acceso a

terrenos agrestes resulta de interés el uso de robots con patas. Estos son capaces de moverse también en entornos de oficinas e industrias, por ser capaces de subir y bajar escaleras. En los robots con patas se ha de prestar atención al control del movimiento de las patas para conseguir un desplazamiento uniforme. Estos estudios se han basado habitualmente en el movimiento de los animales, como se puede comprobar en (Raibert 86).

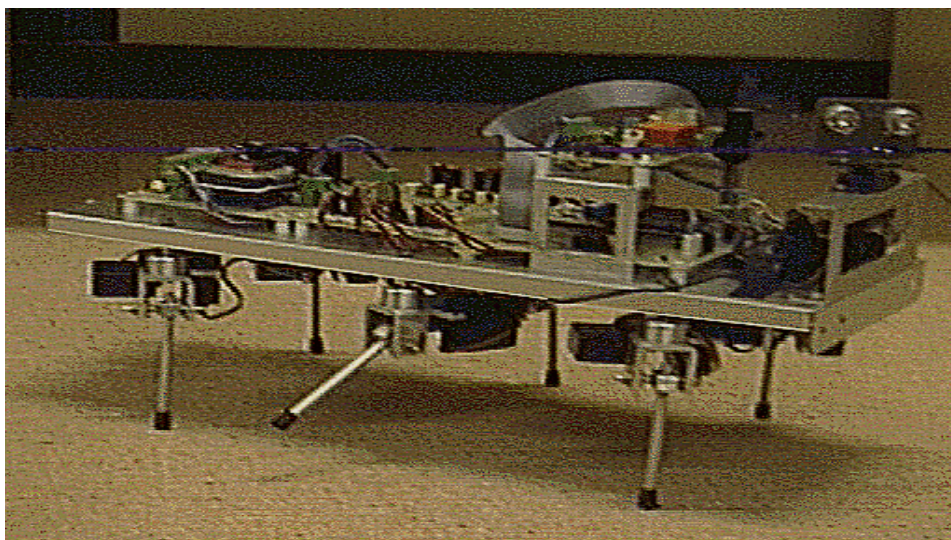
### **El robot autónomo PILAR.**

El diseño del control de este robot autónomo se ha realizado de manera totalmente jerárquica el control de mayor nivel, el de navegación, implementa dos comportamientos diferentes posibles:

- a) Seguir a un objeto y detenerse a una distancia predefinida.
- b) Movimiento libre en un entorno genérico, evitando obstáculos, y tendiendo al área libre más amplia.

En el contexto de este control jerárquico, se propone además un método para la eliminación de bloqueos en el planteamiento de la trayectoria.

Figura 54 Robot autónomo PILAR.



El prototipo de robot autónomo de seis patas ha sido bautizado PILAR (programmable intelligent legged autonomous robot) estando ideado para que pueda moverse en cualquier ambiente, terreno natural, volcanes, minas, plantas nucleares, fabricas, otros.

Cada una de sus patas es controlada mediante dos servomotores eléctricos. El corazón de PILAR es un microcontrolador 68HC11; todo el sistema es alimentado mediante una batería de níquel-cadmio de 9.6V. En la cabeza móvil incluye un sensor de ultrasonidos, mediante el cual mide la distancia a los obstáculos; aunque su capacidad de giro es de  $\pm 80$  grados, solamente se implementan  $\pm 45$  grados. PILAR actúa en función de los obstáculos detectados, y de las reglas borrosas (programadas en el, microcontrolador), que codifican su comportamiento.

El sistema se ha diseñado modularmente, de modo que pueda ser fácilmente expandible y reconfigurable. Absolutamente autónomo, puede caminar en terrenos accidentados. De la misma manera que el modelo ARS esta inspirado en el sistema nerviosos de los insectos, el patrón de coordinación del movimiento de las seis patas de PILAR esta basado en el de los insectos, presentando su misma secuencia temporal.

La Lógica Borrosa tiene una historia corta, pero un rápido crecimiento debido a su capacidad de resolver problemas relacionados con la incertidumbre de la información o del conocimiento de los expertos. Además, proporciona un método formal para la expresión del conocimiento en forma entendible por los humanos.

Estas cualidades le aseguran un amplio campo de aplicabilidad, y un alto interés para las aplicaciones industriales, presentes y futuras. Como conclusión se enuncia un principio dado por Terano "cuanto mas humano deba ser un sistema, mas lógica borrosa tendrá".

La lógica borrosa no debe ser tomada como una panacea no nos debemos llevar por los magníficos resultados que esta presentando en situaciones de control complejas. Con frecuencia la solución óptima consiste en una inteligente combinación de diversas

técnicas, cada una de las cuales debe ser aplicada a aquel aspecto parcial del problema que mejor se adecue.

Si se dispone de un conjunto de reglas proporcionadas por los expertos en un determinado tema, el empleo de sistemas borrosos puede ser tremendamente útil. No obstante, de la combinación de varias técnicas son esperables mejores resultados en cualquier proceso que se pueda controlar.

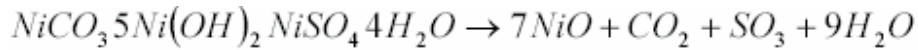
### **5.2.3 PLANTA DE CALCINACIÓN DE LA INDUSTRIA DE PRODUCCIÓN DE NÍQUEL**

#### **Resumen**

Los hornos rotatorios de calcinación constituyen procesos multivariables caracterizados por una gran incertidumbre y fuerte no linealidad en su comportamiento. Los principales objetivos del control en los mismos son garantizar la calidad del producto final y lograr un ahorro de energía, lo que se logra manteniendo un perfil de temperatura adecuada y una buena combustión. Basados en algunos resultados obtenidos con la simulación del control borroso de un horno de calcinación de níquel empleando un software propio que se corresponde con una estructura clásica de controlador borroso. Se realiza una comparación de estos resultados con los que se obtienen mediante la utilización de un algoritmo PI discreto ya implementado y otros paradigmas de control tales como robusto y óptimo, observándose que el control borroso ofrece respuestas más suaves.

#### **Breve descripción del proceso de calcinación en el horno (kiln).**

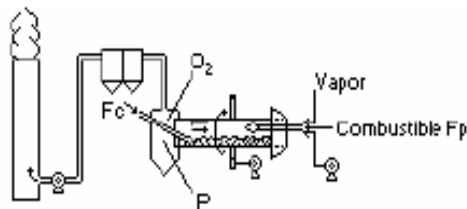
El proceso de calcinación de níquel constituye una importante etapa dentro de esta planta niquelífera, la cual usa como materia prima mineral laterítico y lixiviación amoniacal. En este proceso se obtiene óxido de níquel a partir de carbonato básico de níquel en un horno rotatorio (kiln). En la primera etapa dentro del horno el material se seca, eliminándose totalmente el agua libre y parcialmente la de constitución de las moléculas. En la segunda etapa ocurre la calcinación consistente en la descomposición del carbonato, la cual se puede representar por la siguiente ecuación:



En la etapa final tiene lugar la combustión, donde la temperatura alcanza valores entre 1000 y 1400 °C, aquí ya se obtiene el producto final que pasa a la zona de enfriamiento antes de su descarga.

Existe invariablemente incertidumbre en las características del material procesado que pueden atribuirse al contenido de humedad variable y a la naturaleza también variable de su composición química. El proceso de calcinación es por tanto no lineal y no estacionario por naturaleza. El horno analizado (ver figura 55), consiste en un cilindro rotatorio de acero de 56 m de longitud y 3.5 m de diámetro, éste posee una ligera inclinación con respecto a la horizontal y está revestido interiormente con aislante refractario. Dicho cilindro rota a una velocidad constante de 48 rph y el material es alimentado por el extremo superior. La inclinación del cilindro y su rotación permiten que el material se desplace a través de las distintas zonas

Figura 55 Horno de Calcinación.



El calor en el horno se provee mediante la combustión de petróleo atomizado con vapor y mezclado con aire en un quemador situado en la parte inferior. Los gases que se producen durante el proceso son extraídos continuamente mediante un ventilador de tiro inducido, por la parte superior; mientras que el polvo arrastrado por éstos se recupera en un ciclón y un precipitador electrostático dispuestos en serie y se devuelve al horno a través de transportadores sinfín. La operación del horno debe cumplir los siguientes objetivos:

- ✓ Obtener un producto con calidad, lo que se expresa a través de las especificaciones siguientes:

Níquel. 76 %

Azufre. 0.03 %

Hierro. 0.70 %

Cobalto. 1.30 %

Densidad: 275 - 285 g/cm<sup>3</sup>.

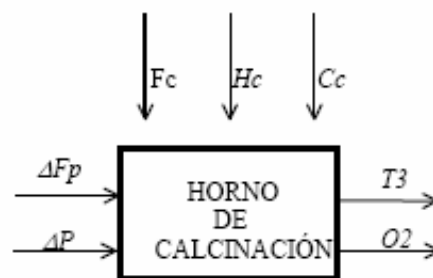
- ✓ Obtener la mayor cantidad posible de producto en la unidad de tiempo.
- ✓ Consumir la menor cantidad posible de petróleo.

Según la experiencia, acumulada durante años por el personal encargado del trabajo del horno para obtener un producto con calidad, es decir, dentro de las especificaciones dadas se deben tener en cuenta los tres aspectos principales siguientes:

- Mantener cierto perfil dado de temperaturas dentro del horno.
- Lograr una buena combustión.
- Mantener una alimentación estable.
- 

En la (Figura 56) se muestra un esquema en bloques que representa el proceso del horno como "caja negra".

Figura 56 Esquema de Bloques.



## **Conclusiones.**

Como se aprecia en las simulaciones realizadas la respuesta del sistema cuando se emplea el controlador borroso es más lenta y presenta error a estado estacionario, en tanto que la respuesta para el PI, es mas rápida y precisa pero presenta un mayor sobrepaso y en el caso del oxígeno mayores oscilaciones. En general debe señalarse que la respuesta del sistema cuando se emplea el regulador borroso es satisfactoria teniendo en cuenta que nunca las desviaciones sobrepasan la norma establecida en el manual de operación del horno, para estas variables: T3 (480 - 610) y O<sub>2</sub> (0.3 - 2.5).

Los resultados obtenidos, nos permiten evaluar de satisfactoria la prueba realizada para la implementación práctica del control borroso en el horno de calcinación estudiado, que según el análisis comparativo realizado con respecto a otros paradigmas verificó su efectividad. La utilización del algoritmo PI para el control de este proceso demostró la necesidad del uso de desacopladores para eliminar las fuertes interacciones entre las variables del proceso, lo que no resulta necesario cuando se usa el controlador borroso, el cual lo tiene en cuenta intrínsecamente.

Los ensayos realizados demuestran la robustez del controlador borroso ante diferentes perturbaciones. En trabajos futuros se presentará el perfeccionamiento del algoritmo planteado introduciendo otro tipo de consecuente en las reglas que permita usar un método de desemborronado menos consumidor de tiempo así como incluir como variable de entrada el cambio del error lo que debe mejorar la respuesta del sistema.

### **5.2.4 CONTROL BORROSO DE UN APARATO DE AIRE ACONDICIONADO.**

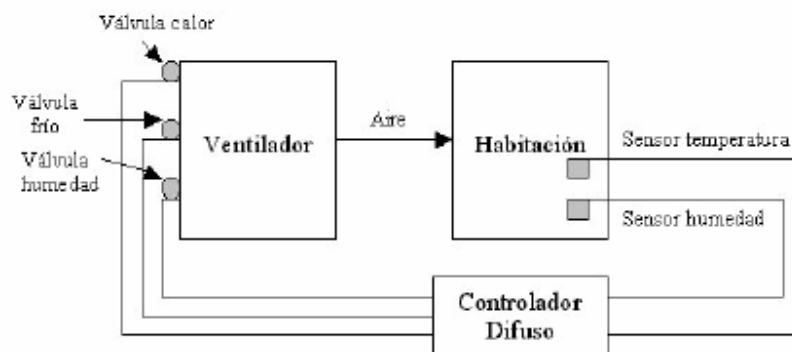
El control de la temperatura es usado en varios procesos. Estos procesos indistintamente de si son de una gran planta industrial, de una casa particular o de un automóvil presentan características poco favorables: no linealidad, interferencias, tiempo muerto, otras. Las aproximaciones convencionales no son satisfactorias habitualmente para resolver el problema de control de temperatura de una manera automática.

Se puede plantear un controlador borroso cuya entrada venga de un sensor de temperatura situado en la habitación que se desea acondicionar y cuya salida sean ordenes a sendas válvulas de frío y calor que permitan acondicionar la temperatura.

Sin embargo, un acondicionador de este tipo no tiene en cuenta la humedad relativa, que es importante para tener un ambiente confortable. Es necesario tener en cuenta la relación entre temperatura y humedad; así, cuando en verano se conecta el aire acondicionado para bajar la temperatura, el aire suele hacerse mas seco con lo que sería importante, además de abrir la válvula de frío, abrir también ligeramente la válvula de humedad. Otra posibilidad es abrir la válvula de humedad cuando el aire es seco, pero es preferible la otra opción de control predictivo para conseguir una mayor sensación de bienestar.

Por tanto se puede plantear un controlador modificado cuyas variables de estado (entradas al sistema de control) sean la temperatura y la humedad y cuyas variables de control sean órdenes a tres válvulas: señales de control a una válvula de calor, a otra de frío y a otra de humedad. El sistema de aire acondicionado se aprecia en la figura:

Figura 57 Controlador de Aire Acondicionado.



En la universidad industrial de Santander un grupo de estudiantes de mecánica elaboro un aparato de aire acondicionado Guiado por un controlador borroso.

## **Control de una Cabina de Aire Acondicionado con Carga Térmica Parcial.**

Es un proyecto realizado por estudiantes de ingeniería mecánica para proyecto de grado el objetivo del sistema es:

- ✓ Implementar un sistema de control para el mantenimiento de condiciones ambientales adecuadas par el confort, en una cabina existente en el laboratorio de aire acondicionado de la escuela de ingeniería mecánica.
- ✓ Se requiere de adaptaciones físicas a la instalación y la incorporación de un sistema de control basado en lógica borrosa para ello se contó con las herramientas COBOR 1.0 y Zadeh 1.0 de estudiantes de ingeniería de sistemas.

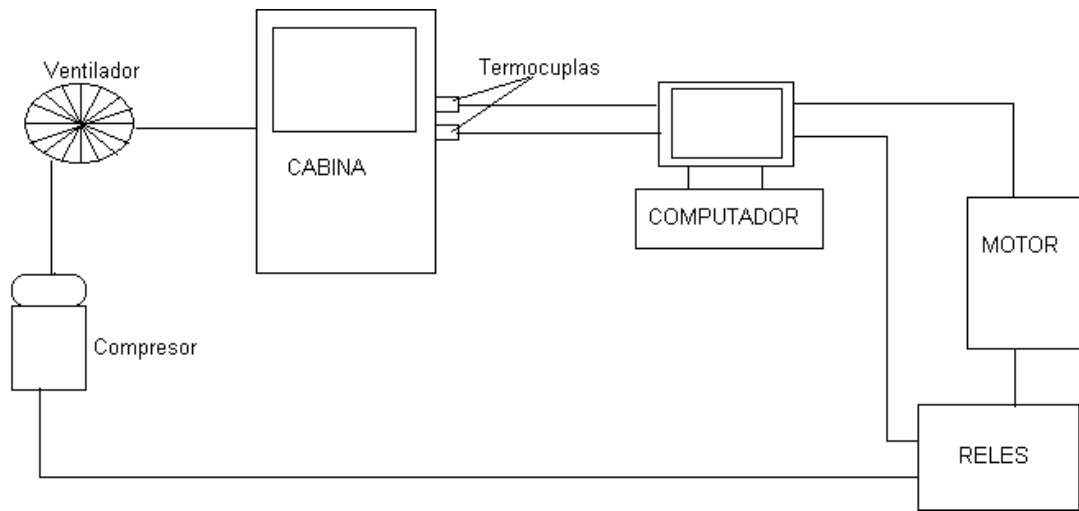
Para lograr mantener las condiciones ambientales de la cabina se utilizaron dos procesos:

- ✓ Control de la capacidad de refrigeración variando la velocidad del motor del compresor Control del flujo de aire, por medio de la variación de la velocidad del ventilador.
- ✓ Con esta aplicación se pretende mostrar los efectos que producen la definición de los conjuntos borrosos, así como las reglas en las variable a controlar ( humedad y temperatura).

Los usuarios ingenieros mecánicos obtuvieron como resultado de COBOR 1.0 una matriz de las acciones de control de velocidad del ventilador y encendido del compresor para la obtención de los valores de referencia de temperatura y humedad.

En la grafica siguiente se puede apreciar los elementos que se necesitaron en la adecuación de la cabina de aire acondicionado y la forma de cómo funciona.

Figura 58 Elementos de la cabina de aire acondicionado.



## CONCLUSIONES.

- La ingeniería de Sistemas se preocupa por encontrar soluciones prácticas a los problemas que se presentan en todas las áreas de la ciencia, por esta razón se ve avocada a buscar nuevas herramientas para la consecución de este objetivo, prueba de ello es la utilización de una teoría como la lógica borrosa, que nos sirve de apoyo para el control de procesos de ingeniería agilizando y facilitando la implementación de los controladores.
- La Lógica Borrosa demostró ser una herramienta muy potente y eficaz para la construcción de controladores, y manejo de procesos de elevada complejidad ingenieril, ya sea por la cantidad de variables que se obtienen como por la no linealidad de sus parámetros.
- Frente a la posibilidad de “modelar” un determinado comportamiento, diferentes personas pueden dar diferentes soluciones expresadas con lenguaje natural dependiendo de la abstracción que cada uno haga de la situación. La lógica borrosa es una rama de la inteligencia artificial que se funda en el concepto "Todo es cuestión de cuanto se cumple", lo cual permite manejar información vaga o de difícil especificación. Es entonces posible con la lógica borrosa gobernar un sistema por medio de reglas de 'sentido común', las cuales se refieren a cantidades indefinidas. Esto hace que la lógica borrosa de un soporte formal al razonamiento basado en lenguaje natural.

- Cuando se trabaja con una herramienta como Cobor 1.0 se busca obtener como resultado “La matriz de Control”, que no hace mas que indicar que acción tomar ante las posibles combinaciones de estados en las entradas del sistema.
- Las aplicaciones de los controladores borrosos han ido creciendo rápidamente gracias a que dan una solución eficiente a la mayoría de procesos donde los controladores tradicionales no dan resultados óptimos.
- Se obtuvo una herramienta con una interfaz amigable y didáctica que lleva de la mano al usuario final a cumplir con su objetivo, la elaboración de un controlador borroso, esta herramienta esta caracterizada por:
  - ✓ Flexibilidad al cargar datos almacenados en archivos planos.
  - ✓ Tener una completa sección para la elaboración de conjuntos borrosos, con las funciones características más utilizadas y los métodos de desemborronado más eficientes.
  - ✓ Tener un motor de inferencia que permita la posibilidad de creación de la base de reglas ya sea combinando las variables de entrada y salida o creando las reglas una por una.
  - ✓ Contar con un analizador de la información que da varias posibilidades según las necesidades del experto del proceso, ya sea de forma grafica, numérica, o generando la matriz de control.
- La elección como herramienta de programación de delphi 6.0 permitió la creación de un motor eficiente y versátil, que absorbió la alta complejidad del proyecto brindando al usuario una transparente y muy fácil manera de crear controladores borrosos.

## RECOMENDACIONES FINALES.

A continuación se señalan una serie de posibilidades para ampliar este trabajo de Investigación:

- La combinación de la lógica borrosa con otros tipos de técnicas de control adaptativo como lo son las redes neuronales o los algoritmos genéticos dan como resultado herramientas poderosas, que se están implementando hoy en día de una manera eficaz y práctica en la solución de problemas complejos. En este sentido se recomienda implementar los sistemas borrosos TSK que son una combinación de la lógica borrosa con redes neuronales.
- Es conveniente entrar a analizar la complejidad de la situación que se desea controlar, además de que es muy recomendable verificar la existencia de un modelo matemático de la situación en cuestión. Ya que si este existe, la solución que este da suele ser más exacta que la de un controlador borroso. Por lo general la lógica borrosa se emplea para resolver problemas de alta complejidad y cuyo modelo matemático no es lineal. La lógica borrosa no es una panacea, pero sabiéndola utilizar puede resultarnos de mucha utilidad en la solución de sistemas con muchas variables.
- Queda abierta la posibilidad de implementar nuevas funciones de membresía o métodos de desemborronado, debido a la flexibilidad de la herramienta, cabe anotar que las implementadas en Cobor 1.0 son las más confiables y eficientes.

## BIBLIOGRAFIA

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

DUBOIS D., and H. PRADE Fuzzy sets and Systems: Theory and applications, academis press, New Cork, 1980.

Eco, Humberto. *Cómo se Hace una Tesis*, Barcelona, España, Editorial Gedisa S.A., 1994.

GELVES O., RANGEL G., y VARGAS H., Control de una cabina de aire acondicionado con carga térmica parcial, pp. 67-102, 1998.

HABER R., Introducción al control borroso, GIALBACOL, Grupo de interés de aplicaciones de la lógica borrosa. TERANO, T., ASAI, K., SUGENO, M., Theory and its applications, 1991.

HELLENDORM H., Y C. Thomas "Defuzzyfiication in fuzzy controllers", intelligent and fuzzy Systems, Vol 1, pp 109-123, 1993.

H. R. BERENJI and P. KHEDKAR, Learning and tuning fuzzy logic controllers through reinforcements. IEEE Trans. Neural Networks 3 (1992)

J. J. BUCKLEY and Y. HAYASHI, Fuzzy neural networks: A survey, Fuzzy Sets and Systems 66 (1994).

KOSKO B., Neural networks and Fuzzy systems, Prentice Hall Englewood cliffs, New Jersey, 1992.

LARMAN, C. UML y Patrones.: Introducción al Análisis y Diseño Orientado a objetos. Pearson.

PRESSMAN, Roger. (1995). Ingeniería del Software, McGraw-Hill. 3ª Ed.

RUIZ D. Fernando. SIMULACIÓN DIGITAL. División de Publicaciones Universidad Industrial de Santander. Bucaramanga. UIS, 2001.

RUIZ D. Fernando., Apuntes de clase “tecnologías adaptativas” UIS 2000.

RUIZ F., PEREZ M., y COBOS M., Desarrollo de un sistema de ayudas para el diseño de controladores borrosos 1997.

SUGENO M., “An introductory survey on fuzzy control” Vol 36, pp 59-83, 1985.

TEIXEIRA Steve & PACHECO Xavier. Guía de desarrollo Delphi 5. ED Prentice Hall.2000.

VELARDE, J. (1991), Gnoseología de los sistemas difusos.Serv. Publicaciones Univ. De Oviedo, Oviedo.

ZADEH L., “Fuzzy and sets “inf, control, Vol 8, pp 338-353, 1965.

ZADEH, L. A. y KACPRZYK, J. (eds.)(1992), Fuzzy Logic for the Management of Uncertainty. John Wiley, Nueva York.

## REFERENCIA BIBLIOGRAFICA.

### TEORIA

- [1] TERANO, T., ASAI, K., SUGENO, M., Theory and its applications, 1991.
- [2] ZADEH L., "Fuzzy and sets "inf, control, Vol 8, pp 338-353, 1965.
- [3] DUBOIS D., and H. PRADE Fuzzy sets and Systems: Theory and applications, academis press, New Cork, 1980.
- [4] HABER R., Introducción al control borroso, GIALBACOL, Grupo de interés de aplicaciones de la lógica borrosa.
- [5] SUGENO M., "An introductory survey on fuzzy control" Vol 36, pp 59-83, 1985.
- [6] KOSKO B., Neural networks and Fuzzy systems, Prentice Hall Englewood cliffs, New Jersey, 1992.
- [7] HELLENDORM H., Y C. Thomas "Defuzzyfiication in fuzzy controllers", intelligent and fuzzy Systems, Vol 1, pp 109-123, 1993.
- [8] RUIZ F., Apuntes de clase "tecnologías adaptativas" UIS 2000.

- [9] GELVES O., RANGEL G., y VARGAS H., Control de una cabina de aire acondicionado con carga térmica parcial, pp. 67-102, 1998.
- [10] DARWIN, C. (1859). On the Origin of Species by Means of Natural Selection or the Preservations of Favored Races in the Struggle for Life. London: John Murray.
- [11] CORREDOR M., Martha Vitalia. Principios de Inteligencia Artificial y Sistemas Expertos. Ediciones UIS. 1982.

## **TECNICA**

- [11] CANTU M, (1999) La Biblia del Delphi 5. 2ª edición, Prentice Hall.
- [12] AYRES J, BOWDEN D, DIEHI L, DORCAS P, (1998) Nucleo API Win 32 McGraw-Hill.
- [13] REISDOROPH, Kent. (1999). Aprendiendo Delphi 5.0 en 21 días. Sams Publishing.

## **METODOLOGICA**

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

- [12] LARMAN, C. UML y Patrones.: Introducción al Análisis y Diseño Orientado a objetos. Pearson.
- [13] PRESSMAN, Roger. (1995). Ingeniería del Software. McGraw-Hill. 3ª Ed.

## ANEXO

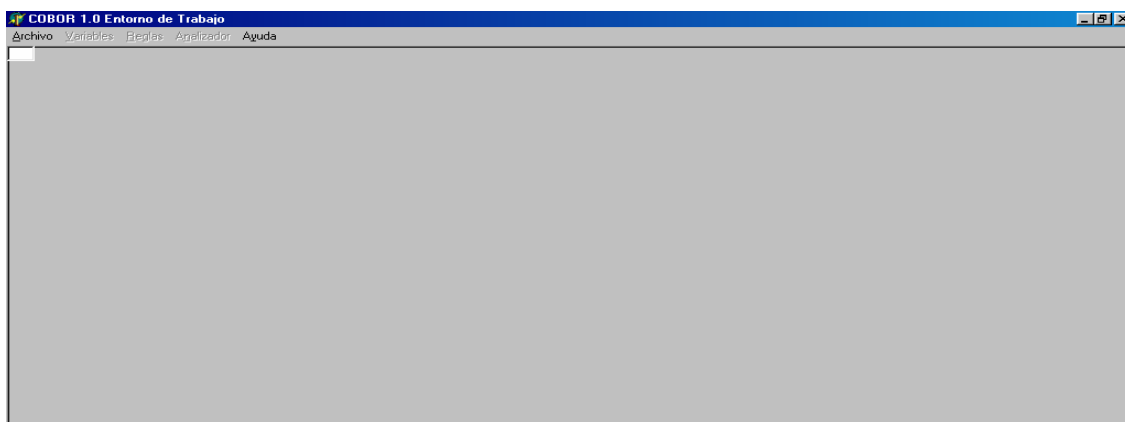
### 1. Que es COBOR 1.0

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 difusos.

### 2. Uso de la Herramienta

Al abrir el programa aparece el entorno de trabajo. Que nos muestra el menú principal con las opciones Variables, Reglas y Analizador inhabilitadas. En este momento puedo Crear un nuevo proyecto o abrir un proyecto ya existente.

Figura 59 Entorno de trabajo Cobor 1.0.



## 2.1 Menú Archivo

Permite crear un nuevo proyecto, abrir un proyecto existente, Guardar los cambios realizados al proyecto abierto actualmente, Editar la información general del proyecto y salir de la herramienta.

### 2.1.1 Nuevo Proyecto

Figura 60 Ventana Nuevo Proyecto

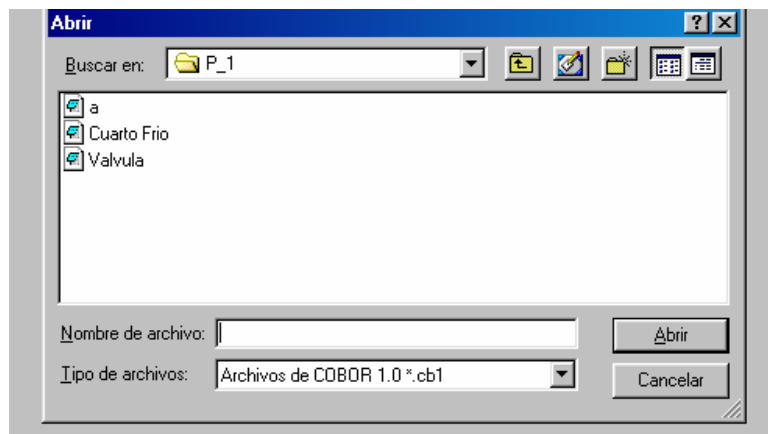
The image shows a software dialog box titled "Nuevo Proyecto". It has a yellow background and a blue title bar. At the top, there is a text input field labeled "Nombre del Proyecto". Below this, there is a section titled "Especificaciones del Proyecto" containing several dropdown menus arranged in two columns. The first column includes "Variables de Entrada" (set to 3), "Variables de Salida" (set to 1), "Metodo Para AND" (set to Min), and "Metodo Para OR" (set to Max). The second column includes "No. de Particiones" (set to 3), "Implicación" (set to Min), and "Agregación" (set to Max). At the bottom of the dialog, there are three buttons: "Aceptar", "Cancelar", and "Ayuda".

El botón aceptar esta inhabilitado hasta que el usuario ingresa el nombre que le dará al nuevo proyecto, a continuación el usuario debe especificar información necesaria para la creación del nuevo proyecto. Por defecto estas casillas tienen los valores mas usados en sistemas de este tipo.

Si la información ingresada para la creación del nuevo proyecto esta conforme con las necesidades del problema, se da clic en el botón *Aceptar*, de lo contrario se puede cancelar la intención de crear un nuevo proyecto dando clic sobre el botón *Cancelar*.

### 2.1.2 Abrir

Figura 61. Cuadro de dialogo abrir proyecto existente

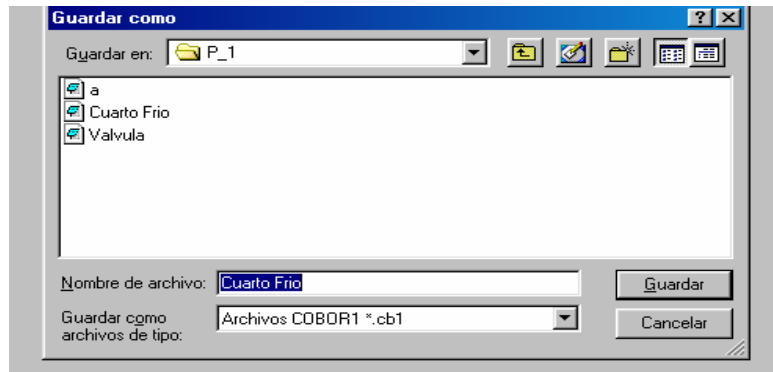


Permite abrir un proyecto ya existente, la extensión de los archivos de texto en que son guardados los proyectos de Cobor 1.0 es .cb1, el cuadro de dialogo filtra estos archivos.

### 2.1.3 Guardar

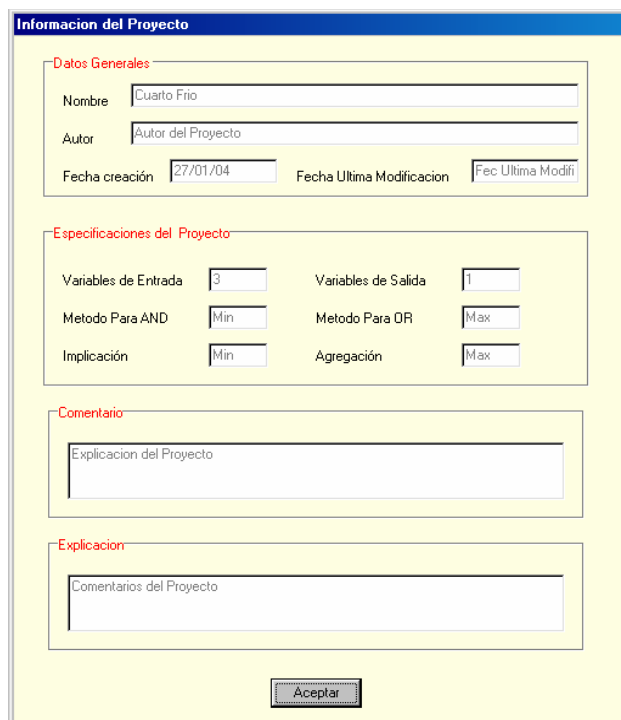
Guarda los cambios realizados al proyecto abierto actualmente, la extensión .cb1 la pone por defecto.

Figura 62. Cuadro de dialogo Guardar Proyecto como



#### 2.1.4 Información del Proyecto

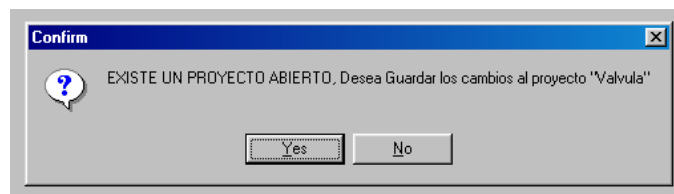
Figura 63. Ventana Información del proyecto



Muestra la información general del proyecto, y permite hacer cambios en las casillas habilitadas para tal fin.

### 2.1.5 Salir

Figura 64. Mensaje de confirmación Guardar Cambios



Dando clic al botón Salir, se despliega un cuadro de dialogo que pregunta si se desea guardar los cambios hechos al proyecto abierto, dependiendo de la opción elegida, se realiza la acción y luego se cierra la aplicación.

## 2.2 Menú Variable

Para agregar, editar y eliminar tanto variables de entrada como de salida, se pueden agregar particiones a las mismas variables, cambiar el tipo de función de membresía.

### 2.2.1 Agregar

Lo primero que se hace es seleccionar el tipo de variable que se desea agregar: Entrada ó Salida, luego se entra la información requerida según el tipo de variable que se esta agregando. Al dar clic sobre el botón aceptar se guarda en el proyecto la nueva variable.

Figura 65. Ventana nueva Variable

The image shows a software dialog box titled "Nueva Variable". It is organized into several sections:

- Tipo de Variable:** Two radio buttons are present: "Entrada" (unselected) and "Salida" (selected).
- No Particiones:** A dropdown menu currently displays the number "3".
- Función de Membresía:** A dropdown menu currently displays the word "Triangular".
- Desemborronado:** A dropdown menu currently displays "Centro de Área".
- Nombre:** A text input field.
- Rango:** Two input fields labeled "Mín" and "Máx".
- Unidades:** A dropdown menu currently displaying "Unidades".
- Explicación:** A text input field.
- Comentario:** A text input field.

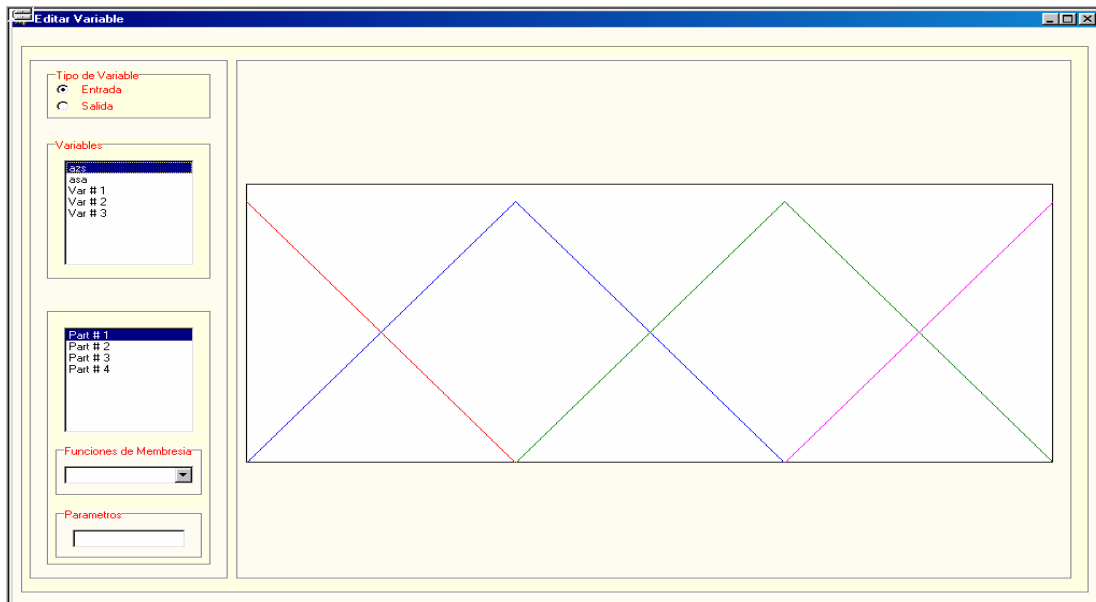
At the bottom of the dialog box, there are three buttons: "Aceptar", "Cancelar", and "Ayuda".

### 2.2.2 Editar

La ventana editar permite editar las variables que hacen parte del proyecto abierto, al seleccionar el tipo de variable se despliega un cuadro de selección que según el tipo escogido muestra las variables existentes en el proyecto, al seleccionar alguna de estas variables se despliega un cuadro de selección que muestra las particiones que tiene la variable seleccionada. Al seleccionar alguna partición es posible cambiar la función de membresía y los parámetros.

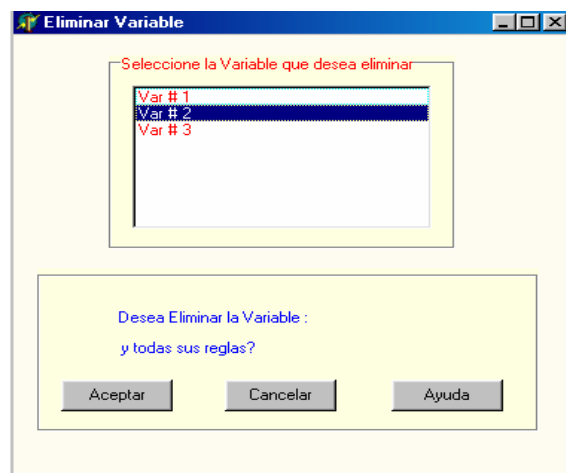
En esta ventana también se puede agregar y eliminar particiones de una variable, lo mismo que agregar y eliminar variables al proyecto.

Figura 66. Ventana Editar Variable



### 2.2.3 Eliminar

Figura 67. Ventana Eliminar Variable



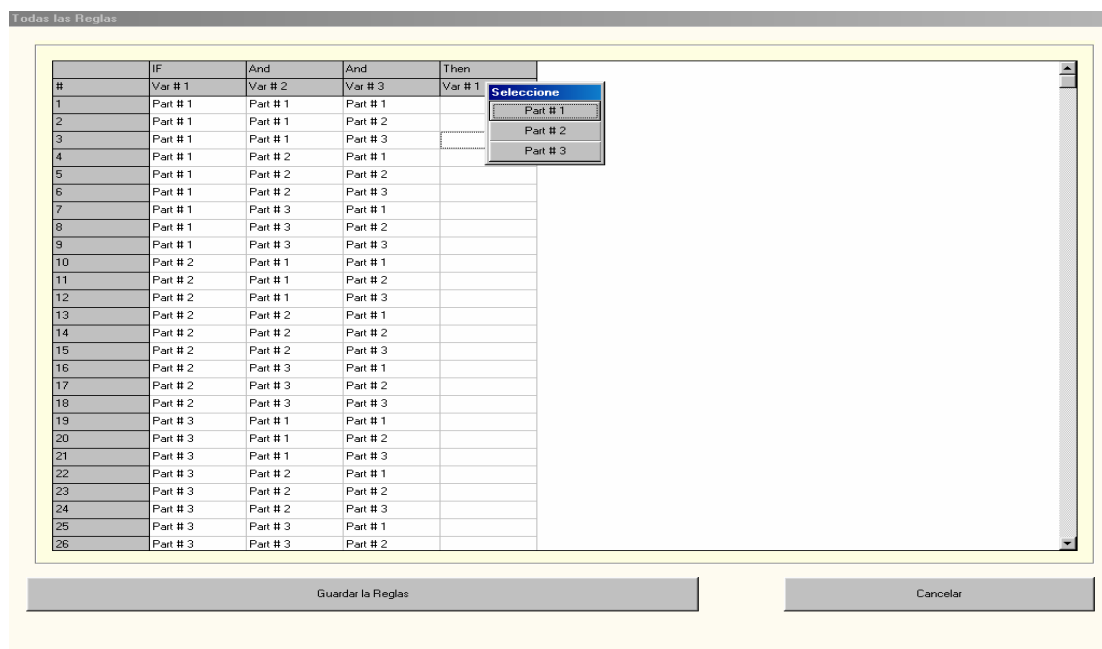
Despliega un cuadro de selección que elimina la variable escogida, luego despliega un mensaje de confirmación y procede según sea la elección.

## 2.3 Reglas

Para crear la base de reglas, agregando, editando y eliminando, presenta opciones para crear todas las reglas al tiempo y una por una.

### 2.3.1 Crear Todas

Figura 68. Ventana Crear Todas las Reglas

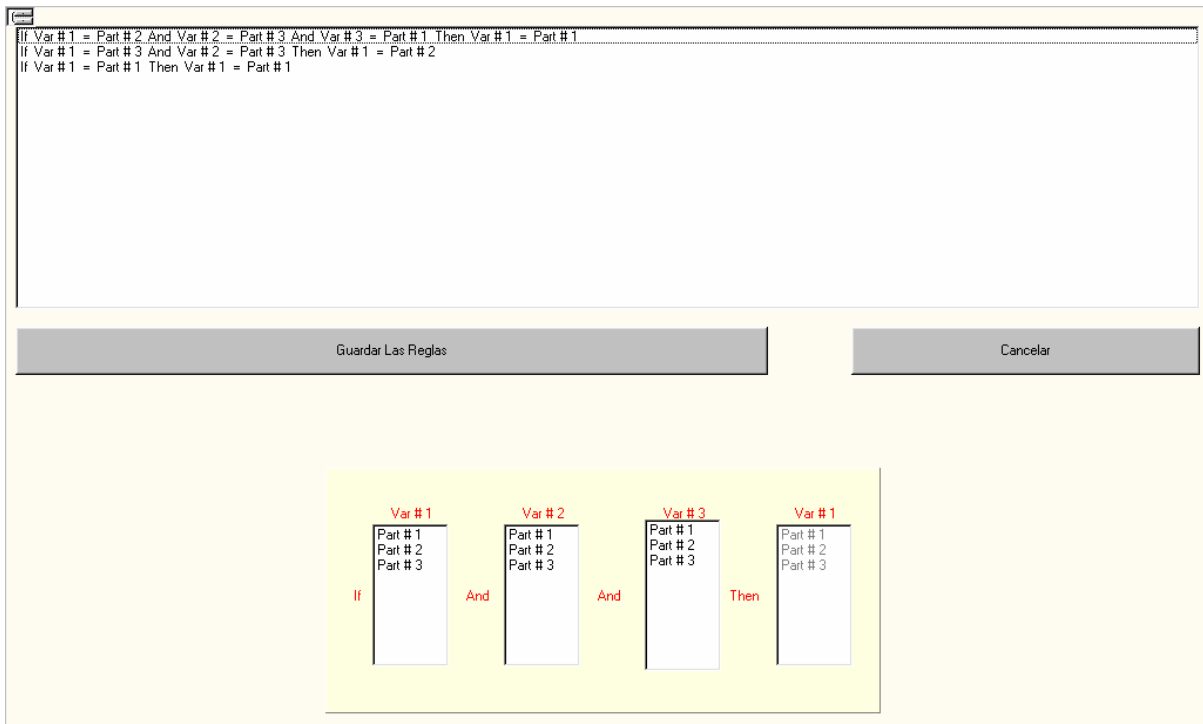


Crea una matriz donde guarda todas las posibles combinaciones de las particiones de las variables de entrada, dejando como elección del usuario la acción que se debe tomar en las variables de salida.

Al dar clic al botón Guardar reglas se guardan en la base de reglas únicamente aquellas que tienen acción a tomar en la variable de salida.

### 2.3.2 Crear Una x Una

Figura 69. Ventana Crear Reglas Una por Una



Se muestran las tanto las variables de entrada como las de salida y cada una de sus particiones, para armar la regla de da doble clic sobre la partición seleccionada, de esta forma se van agregando una por una las reglas a la base de reglas.

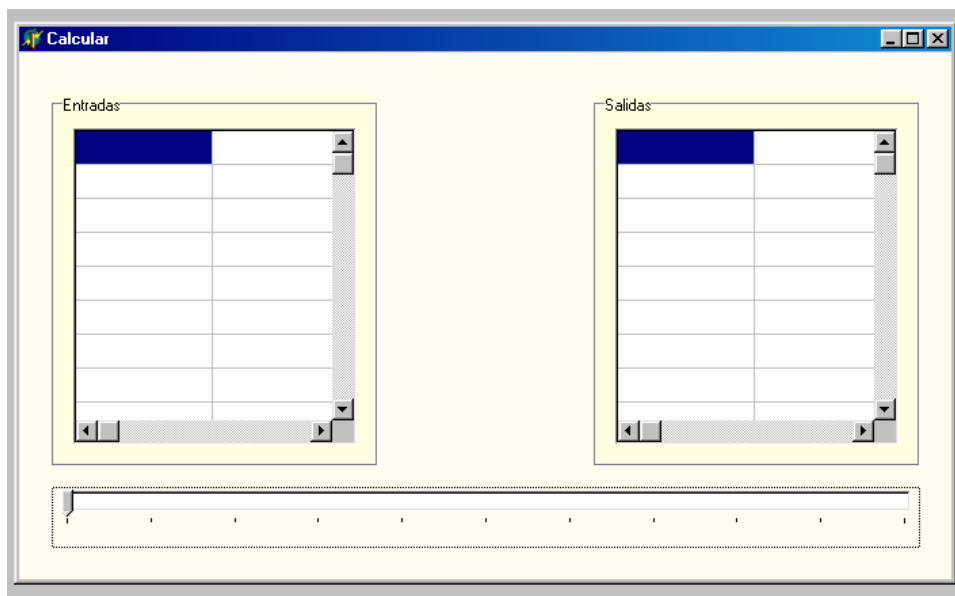
Al dar clic sobre el botón Guardar las reglas, estas son agregadas a la base de reglas.

## 2.4 Analizador

Este menú permite ver los diferentes tipos de resultados que arroja la herramienta.

### 2.4.1 Calcular

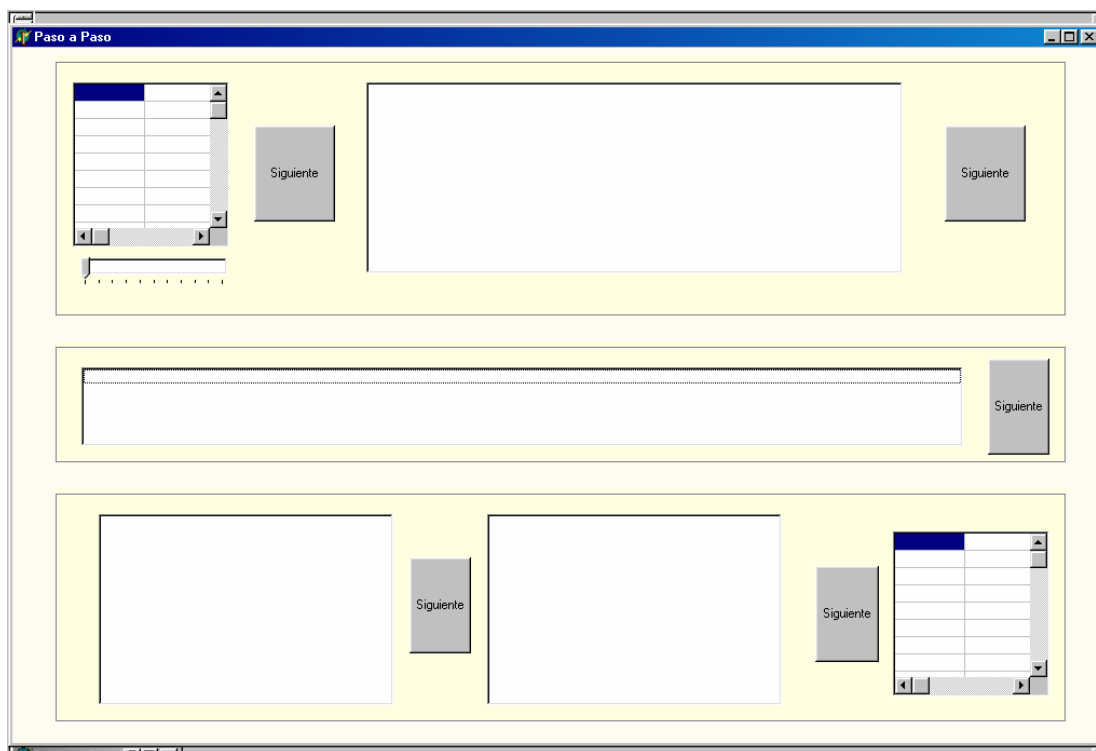
Figura 70. Ventana Calcular



El objetivo es que el usuario ingrese valores para un caso particular y el sistema calcule el valor de las variables de salida. Muestra las variables de entrada existentes y permite modificar los valores que tienen, al hacerlo la respuesta en las variables de salida cambia.

## 2.4.2 Paso a Paso

Figura 71. Ventana Paso a Paso

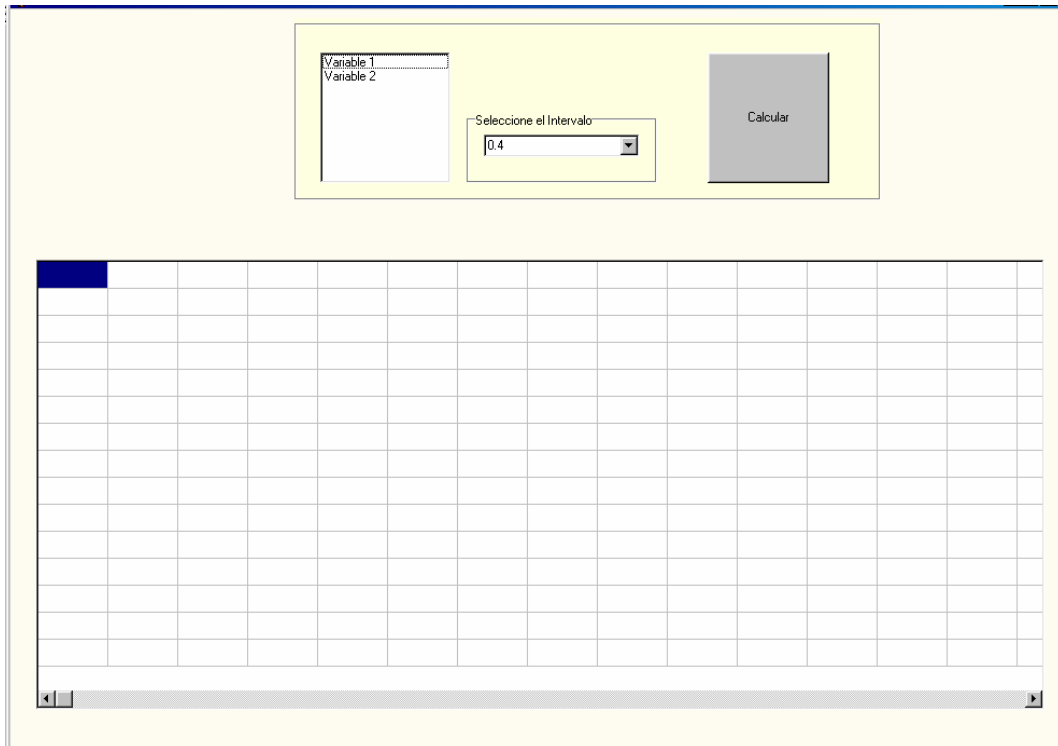


El objetivo es mostrar el procedimiento que se sigue desde el ingreso de los valores iniciales para las variables de entrada hasta la respuesta en las variables de salida. Esto se hace para un caso particular.

Muestra las variables de entrada existentes y permite modificar los valores que tienen, damos clic al botón siguiente, esto muestra la gráfica de las particiones se da clic al botón siguiente y se muestra las reglas que se activan para este caso, de nuevo se da clic al botón siguiente y se muestra la implicación, damos otro clic para ver la agregación y ya para terminar el proceso paso a paso damos clic al botón siguiente, lo que nos muestra el valor de las variables de salida después de hacer todo el proceso.

### 2.4.3 Matriz

Figura 72. Ventana Matriz de Acciones



Genera la matriz de control o matriz de acciones, Inicialmente se debe seleccionar para cada variable de entrada cada cuanto se debe calcular un nuevo valor de salida, luego se da clic al botón calcular.

### 2.4.4 Grafico

Muestra el comportamiento gráficamente para dos variables de entrada y una de salida, se debe seleccionar que variables de entrada van a ser usadas al igual que la variable de salida.

Figura 73. Ventana Resultado Grafico

